PETROLEUM RESERVOIR SIMULATION

Khalid Aziz
and
Antonín Settari

PETROLEUM RESERVOIR SIMULATION

Khalid Aziz
and
Antonín Settari

PETROLEUM RESERVOIR SIMULATION

PETROLEUM RESERVOIR SIMULATION

KHALID AZIZ
Otto N. Miller Professor Emeritus
Energy Resources Engineering
Stanford University
Stanford, California, U.S.A.

and

ANTONÍN SETTARI
Professor Emeritus
Department of Chemical and Petroleum Engineering
University of Calgary
TRS Energy Consultants Ltd.
Calgary, Alberta, Canada

© 1979 Khalid Aziz and Antonín Settari

First printed in 1979
Paperback edition 2002
E-book edition 2020

Published by Khalid Aziz and Antonín Settari

National Library of Canada Cataloguing in Publication Data

Aziz, Khalid, 1936-
Petroleum reservoir simulation

Includes bibliographical references and index.
With 143 illustrations.
ISBN 0-9730614-0-5

1. Oil reservoir engineering—Mathematical models. 2. Oil reservoir engineering—Computer simulation. I. Settari, Antonín. II. Title.
TN871.A93 2002 622'.3382'011 C2002-910339-8

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers.

Printed and bound in Canada by Blitzprint Ltd., Calgary, Alberta.

DEDICATED TO
MUSSARRAT AND BARBARA

ACKNOWLEDGEMENTS

The authors are indebted to many individuals and institutions who have contributed to this work, particularly:

B. Agbi, A. Spivak and J. W. Watts for reviewing the manuscript.

S. C. M. Ko (who also provided results of some of his unpublished work), J. Abou-Kassem, J. W. Grabowski, R. Mehra, B. Rubin and many other students and colleagues for comments on various parts and versions of the manuscript over the years.

Pat Hitchner, Brenda Oberhammer, and Betty Lewis for typing various versions of the manuscript with considerable interest and great patience, and for assisting in many other ways.

The National Research Council (Canada), Energy, Mines and Resources (Canada), The Department of Chemical Engineering of the University of Calgary, and the Computer Modelling Group for directly or indirectly supporting this project.

The University of Calgary for the award to K. Aziz of a Killam Resident Fellowship so that this work could be completed.

Intercomp Resource Development and Engineering Inc. for giving A. Settari permission to work on this project and for creating conditions conducive to such work.

K. H. Coats and other researchers in this field, including many Intercomp staff from whose experience we have benefited.

The Society of Petroleum Engineers, Pergamon Press and the Society of Industrial and Applied Mathematics for permission to reproduce material from their publications.

Marilyn Croot of the University of Calgary for drafting figures. Natasha Aziz and Imraan Aziz for assistance with the literature files and photocopying.

Editors of Applied Science Publishers for their keen interest in this manuscript.

PREFACE

This book is intended for theoretically minded engineers, and practically oriented mathematicians and scientists who want to understand how to develop and use computer models of petroleum reservoirs.

This is not a numerical analysis book, although most of the book deals with the use of numerical techniques for solving partial differential equations. There are several books on the numerical solution of partial differential equations, but they deal with equations that do not have all of the important characteristics of the equations describing multiphase flow in petroleum reservoirs. The equations to be solved for the simulation of petroleum reservoirs have some rather special features that must be considered by the simulation engineer or scientist. The engineering, physics and mathematics of the problem are so intertwined that a good understanding of all three aspects is essential before one can hope to develop good models.

The book should be suitable for short courses designed for practising engineers and for self study. It is also hoped that it will serve as a reference for scientists and engineers engaged in the development and applications of simulation technology. Many of the ideas developed here apply directly to the simulation of ground water movement.

In our own experience, we have found no substitute for gaining the kind of understanding of the theory that is obtained by writing and testing computer programs. It is therefore recommended that in any course dealing with reservoir simulation the readers be asked to develop some simulation programs, such as a simple one-dimensional single-phase model (Chapter 3), a one-dimensional two-phase model (Chapter 5), and a two-dimensional single-phase model (Chapter 7). Some of the basic sub-routines required for these models are contained in Appendix B.

In the presentation of the material, we have tried to introduce every concept in the simplest possible setting and maintain a level of treatment which is as rigorous as possible without being unnecessarily abstract. A brief discussion of some of the basic concepts of numerical analysis has been provided in the text as needed and the reader is referred to appropriate references for more detail. In the presentation of material concerning reservoir simulation, we have attempted to develop a consistent notation and terminology along with a thorough discussion of various theoretical and practical aspects of the subject. It has not been our intention to establish historical precedence, since ideas have been developed simultaneously by several people and some results have not been published for competitive reasons.

This book contains a relatively complete treatment of finite-difference models of black-oil type reservoirs, but does not include such topics as simulation of thermal recovery processes, chemical flooding, miscible displacement (except for a brief treatment in Chapter 12), and the use of variational methods in simulation. This has been done to keep the size of the book reasonable and also because these areas are undergoing rapid development at this time.

KHALID AZIZ
ANTONÍN SETTARI

PREFACE TO THE PAPERBACK EDITION

Although this book was first published in 1979, most of the work was completed twenty-five years ago. After going through five printings and several publishers (Applied Science, Elsevier Applied Science and Chapman & Hall), the book went out of print in 1996. Since then, we have informally made copies available to our students and colleagues.

Much has changed since the original publication, but the fundamental concepts upon which all simulators are built remain the same. Since our original focus was on the underlying fundamentals, the book is just as valuable today as it was at the time of its first publication. While a new book that deals with recent advances is desirable, this text continues to be in demand by petroleum engineering students and practicing engineers. We are pleased to be able to make it available once again.

KHALID AZIZ
ANTONÍN SETTARI
March 2002

PREFACE TO THE e-BOOK EDITION

This book was first published in 1979. Although most of the work was completed by 1977, the material still remains relevant today. After going through five printings and several publishers (Applied Science, Elsevier Applied Science and Chapman & Hall), the book went out of print in 1996. In 2002 we acquired the copyright and published the book ourselves as a paperback edition, without any changes from the last hardback printing. This e-book edition is intended to make the book widely and inexpensively

available. The current version has been reviewed and corrected for errors as well as for consistency in the use of variables and fonts, both in text and equations.

Much progress has been made since the original publication, but the fundamental concepts remain the same. Since our original focus was on the underlying fundamentals, the book continues to be valuable for students and practicing engineers. In addition, it provides a historical perspective of the developments in the field of reservoir simulation from 1970's. The discipline has broadened in the scope of physics represented, numerical techniques and progress in underlying computing science. In order to introduce more recent developments, we intend to invite leaders in the field to make contributions through appendices, thus making it a living book.

KHALID AZIZ
ANTONÍN SETTARI
September 2020

CONTENTS

Preface

Nomenclature

1. *INTRODUCTION*

1.1 What is a Computer Model?
1.2 Other Models
1.3 What Questions Can a Computer Model Answer?
1.4 Concluding Remarks

2. *FLUID FLOW EQUATIONS*

2.1 Introduction
2.2 The Law of Mass Conservation
  2.2.1 Single-Phase Flow
  2.2.2 Multiphase Flow
2.3 Darcy's Law
  2.3.1 Single-Phase Flow
  2.3.2 Multiphase Flow
2.4 The Basic Flow Equations
  2.4.1 Single-Phase Flow
  2.4.2 Multiphase Flow
  2.4.3 Use of Pseudopotential
  2.4.4 Boundary Conditions
2.5 Alternative Forms of Multiphase Flow Equations
  2.5.1 Formulation in 'Parabolic' Form
  2.5.2 Formulation in 'Hyperbolic' Form
2.6 Flow Equations Which Include Non-Darcy Effects
  2.6.1 High Flow Rates (Inertial and Turbulent Effects
  2.6.2 Threshold and Slip Phenomena
  2.6.3 Non-Newtonian Flow
  2.6.4 Other Effects
2.7 Fluid and Rock Properties
  2.7.1 Fluid Properties
  2.7.2 Rock Properties
2.8 Concluding Remarks

Exercises

3. *FLOW OF A SINGLE FLUID IN ONE DIMENSION*

3.1 Introduction
3.2 Finite-Difference Approximations
3.2.1 Discretisation in Space
3.2.2 Discretisation in Time
3.2.3 Discretisation Errors
3.3 Other Selected Methods
3.3.1 Other Explicit Methods
3.3.2 Other Implicit Methods
3.3.3 ODE Methods
3.3.4 Comparison of Methods
3.4 Grid Systems and Boundary Conditions
3.4.1 Two Methods of Grid Construction
3.4.2 Boundary Conditions
3.5 Discretisation of One-Dimensional Flow Equations in Cartesian Co-ordinates
3.5.1 Difference Equations for Irregular Grid
3.5.2 Difference Equations in Matrix Form
3.5.3 Treatment of Variable Coefficients
3.6 Discretisation of One-Dimensional Flow Equations in Radial Cylindrical Co-ordinate
3.6.1 Difference Equations for Irregular Grid
3.6.2 Difference Equations in Matrix Form
3.6.3 Treatment of Variable Coefficients
3.7 Some Properties of Finite-Difference Equations
3.7.1 Existence of Solution and Material Balance
3.7.2 Treatment of Nonlinearities
3.8 Concluding Remarks
Exercises

4. *SOLUTION OF TRIDIAGONAL MATRIX EQUATIONS*
4.1 Introduction
4.2 Methods of Solution
4.2.1 Thomas' Algorithm
4.2.2 Tang's Algorithm
4.2.3 Solution of Symmetric Tridiagonal Matrix Equations
4.2.4 Special Cases of Non-Unique Solution
4.2.5 Other Special Cases
Exercises

5. *MULTIPHASE FLOW IN ONE DIMENSION*
5.1 Introduction
5.2 The Simultaneous Solution (SS) Method
5.2.1 The SS Method for Two-Phase Flow
5.2.2 Extension of the SS Method to Three-Phase Flow
5.2.3 Other Formulations of the SS Method
5.3 The Implicit Pressure-Explicit Saturation (IMPES) Method

5.3.1 IMPES Method for Three-Phase Flow
5.3.2 Other Derivations of the IMPES Method
5.4 Analysis of SS and IMPES Methods
5.4.1 Stability
5.4.2 Existence and Uniqueness of Solution
5.4.3 Convergence
5.5 Treatment of Nonlinearities
5.5.1 Weighting of Transmissibilities
5.5.2 Approximation to Transmissibilities in Time
5.5.3 Nonlinearity due to P_c Function
5.5.4 Gas Percolation
5.6 The Sequential Solution Method (SEQ)
5.6.1 SEQ Method for Two-Phase Flow
5.6.2 Other Forms and Derivations
5.6.3 Numerical Results
5.6.4 SEQ Method for Three-Phase Flow
5.6.5 Discussion
5.7 Treatment of Production Terms
5.7.1 Differential Form of Boundary Conditions
5.7.2 Discretisation of Boundary Conditions
Exercises

6. *SOLUTION OF BLOCK TRIDIAGONAL EQUATIONS*
6.1 Introduction
6.2 Methods of Solution
6.2.1 Extension of Thomas' Algorithm
6.2.2 Use of Methods for Band Matrices

7. *FLOW OF A SINGLE FLUID IN TWO DIMENSIONS*
7.1 Introduction
7.2 Classification of 2-D Problems
7.2.1 Areal Problems (x–y)
7.2.2 Cross-Sectional Problems (x–z)
7.2.3 Single-Well Problems (r–z)
7.2.4 Comments on Two-Dimensional Models
7.3 Discretisation of Flow Equations
7.3.1 Difference Approximations
7.3.2 Stability of Difference Schemes
7.4 Boundary Conditions
7.4.1 No-Flow or Closed Boundaries
7.4.2 Flow Boundaries
7.4.3 Discretisation of Boundary Conditions
7.5 Initial Conditions
7.6 Treatment of Nonlinearities

7.7 Treatment of Individual Wells
7.8 Equations in Matrix Form
7.9 Special Methods for 2-D Problems
7.9.1 Alternating Direction Explicit (ADE) Methods
7.9.2 Alternating Direction Implicit (ADI) and Related Methods
7.9.3 Comparison of Methods
7.10 Methods of Grid Construction
7.10.1 Irregular Grid in 2-D
7.10.2 Use of a Curvilinear Grid
7.11 Concluding Remarks
Exercises

8. *SOLUTION OF PENTADIAGONAL MATRIX EQUATIONS*
8.1 Introduction
8.2 Direct Methods of Solution
8.2.1 LU Factorisation
8.2.2 Ordering of Equations
8.2.3 Sparse Matrix Techniques
8.3 Iterative Methods
8.3.1 Point Jacobi Method
8.3.2 Point Gauss-Seidel Method
8.3.3 Point Successive Over Relaxation (SOR) Method
8.3.4 Line and Block SOR Methods
8.3.5 Additive Correction Methods
8.3.6 Iterative Alternating Direction Implicit (ADI) Methods
8.3.7 Strongly Implicit Method
8.3.8 Other Methods
8.3.9 Comparison of Iterative Methods
8.3.10 Practical Considerations in the Use of Iterative Methods
8.4 Comparison of Iterative and Direct Methods
8.5 Concluding Remarks
Exercises

9. *MULTIPHASE FLOW IN TWO DIMENSIONS*
9.1 Introduction
9.2 Classification of 2-D Problems
9.2.1 Areal Problems (x–y)
9.2.2 Cross-Sectional Problems (x–z)
9.2.3 Single-Well Problems (r–z)
9.2.4 General Comments
9.3 Methods of Solution and Their Comparison
9.3.1 Discretisation in 2-D
9.3.2 Stability of SS and IMPES in 2-D
9.3.3 Comparison of Various Solution Methods and Computer Requirements

9.4 Boundary Conditions
9.4.1 Differential Formulation
9.4.2 Compatibility Conditions and Constraints
9.4.3 Finite-Difference Formulation
9.5 Initial Conditions
9.6 Simulation of Aquifers
9.7 Simulation of Areal and Cross-Sectional Problems
9.7.1 Use of Curvilinear Grid
9.7.2 Treatment of Individual Wells
9.7.3 Grid Orientation Phenomenon
9.8 Simulation of Single-Well Problems
9.8.1 Treatment of the Production Terms (the Well Model)
9.8.2 Comparison of Stability and Efficiency of Various Treatments of Transmissibilities
9.8.3 Practical Considerations
9.9 Concluding Remarks
Exercise

10. *SOLUTION OF BLOCK PENTADIAGONAL EQUATIONS*
10.1 Introduction
10.2 Direct Methods
10.3 Iterative Methods
10.3.1 BSOR Method
10.3.2 Iterative ADI Method
10.3.3 The SIP Method
10.3.4 Comparison of Iterative Methods
10.4 Comparison of Iterative and Direct Methods
10.5 Concluding Remarks
Exercise

11. *THREE-DIMENSIONAL PROBLEMS AND SOLUTION TECHNIQUES*
11.1 Introduction
11.2 Single-Phase Flow
11.2.1 Basic Equation and Discretisation
11.2.2 Special Methods for 3-D Problems
11.2.3 Direct Methods of Solution
11.2.4 Iterative Methods
11.2.5 Comparison of Methods
11.3 Multiphase Flow
11.3.1 Basic Solution Methods and Their Work Requirements
11.3.2 Methods for Solving the Matrix Equations
11.4 Concluding Remarks
Exercises

12. *SPECIAL TOPICS*

12.1 Introduction
12.2 Pseudo-Functions
12.2.1 Coats *et al*. (1971a) Vertical Equilibrium Model
12.2.2 Other Pseudo-Functions
12.3 Stream Tube and Related Models
12.4 Simulation of Variable Bubble-Point Problems
12.5 Simulation of Systems not Described by Black-Oil Models
12.5.1 Simulation of Miscible Displacement
12.5.2 Simulation of Compositional Effects
12.6 History-Dependent Saturation Functions
12.6.1 Physical Model of Hysteresis
12.6.2 Numerical Treatment of Hysteresis
12.7 Simulation of Naturally Fractured Reservoirs
12.8 Automatic Time-Step Control
12.9 Concluding Remarks
Exercise

13. *PRACTICAL CONSIDERATIONS*
13.1 Program Development
13.1.1 Development of the Mathematical Model
13.1.2 Development of the Numerical Model
13.1.3 Development of the Computer Model (program)
13.2 Program Usage
13.2.1 Steps Involved in a Simulation Study
13.2.2 Selection and Design of the Model
13.2.3 History Matching
13.3 Concluding Remarks

APPENDIX A

APPENDIX B

BIBLIOGRAPHY

INDEX

NOMENCLATURE

Symbols and abbreviations used repeatedly are defined below

A	cross-sectional area of a block
B_l	formation volume factors defined by eqns. (2.8–2.10)
$b_l = 1/B_l$	reciprocal of the formation volume factor
C	an arbitrary constant
C	concentration, Chapter 12
c	accumulation coefficient
c_f	fluid compressibility, eqn. (2.37)
c_R	rock compressibility, eqn. (2.41)
$E_\mathrm{n} = \max_i \lvert e^n{}_i \rvert$	norm of error
$e_i = U_i - u_i$	the error in the approximate solution at point i
F_w	inverse function of $P_c(S_\mathrm{w})$
f	an arbitrary function
$f_\mathrm{n} = \lambda_\mathrm{n}/(\lambda_\mathrm{w} + \lambda_\mathrm{n})$	fractional flow coefficient of the non-wetting phase
$f_\mathrm{w} = \lambda_\mathrm{w}/(\lambda_\mathrm{w} + \lambda_\mathrm{n})$	fractional flow coefficient of the wetting phase
g	gravitational acceleration
$\boldsymbol{g}$	gravitational vector
g_c	conversion constant, = 32·2 $\mathrm{lb_m/lb_f}$. $\mathrm{ft/sec^2}$
h	grid spacing, Chapter 3
h	reservoir thickness, Chapter 12
h	elevation (positive downwards)
K_i	vapour–liquid equilibrium ratio (K value) for component i
k, $k_{x,y,z}$	permeability, or the components of the permeability tensor
$k_{\mathrm{r}l}$	relative permeability of phase l
k_rog	oil relative permeability in the oil–gas system
k_row	oil relative permeability in the oil–water system
L	length
M	molecular weight, Chapter 2
M	number of points in a grid system, Chapter 3
$M = \lambda_\mathrm{w}/\lambda_\mathrm{n}$	mobility ratio

$M = \mu_o/\mu_s$	mobility ratio for miscible flow, Chapter 12
$m = \rho\phi$	mass per unit volume
$\dot{m}$	mass flux, mass flow per unit area per unit time
N	number of unknowns in a finite-difference scheme after unknowns due to boundary conditions have been eliminated
P_c	capillary pressure
P_{co}	value of capillary pressure outside the porous medium
P_{cog}	oil–gas capillary pressure
P_{cow}	oil–water capillary pressure
PI	influence function, eqn. (9.52)
p	pressure (U, u also represent pressure)
p_b	bubble-point pressure
p_l	pressure of the phase l
p_s	saturation pressure
p^w	pressure at wellbore
p_{wf}	flowing well pressure
QI	influence function, eqn. (9.51)
$Q'_{lp} = \frac{\partial Q_l}{\partial p}$	rate derivative with respect to pressure
$Q'_{lm} = \frac{\partial Q_l}{\partial S_m}$	rate derivative with respect to saturation
Q_{TL}	total liquid flow rate
Q_{To}	total oil flow rate
Q_{TT}	total fluid flow rate
q	sink (production per unit time); q is negative for injection
$\tilde{q}$	mass depletion per unit volume per unit time, positive for production, negative for injection
q_i	approximate mean value of q in a block i
$q_l = \bar{q}_l/\rho_{lSTC}$	volume of stock tank component l produced per unit of reservoir volume per unit time
R	universal gas constant
$R(\mathbf{A}^v)$	average rate of convergence for v iterations
R_i	local discretisation error at point i
R_s	solution gas–oil ratio
r	space co-ordinate (distance in the radial direction)
r_e	external radius

r_w	radius of the well
S_l	saturation of phase l
S_{gc}	critical or residual gas saturation depending on the direction of displacement
S_{gc}	residual gas saturation in liquid displacement, Chapter 12
S_{gcr}	critical gas saturation, Chapter 12
S_{gmax}	maximum saturation for the gas phase
S_{nc}	critical saturation of the non-wetting phase in a drainage cycle or residual saturation in an imbibition cycle
S_{wc}	critical saturation of the wetting phase in an imbibition cycle or residual saturation in a drainage cycle
S_{wmax}	maximum saturation for the water phase
S_{wo}	value of S_w corresponding to P_{co}
T	temperature, Chapter 2
$T = (\lambda A/\Delta x)$	finite-difference transmissibility
$T_l = (\lambda_l A/\Delta x)$	finite-difference transmissibility for phase l
t	time
Δt	time increment
U	dependent variable (exact solution of a partial differential equation)
$\boldsymbol{u}$	superficial or Darcy velocity
u_i	approximation of U at grid point i (exact solution of algebraic equations obtained by the application of some approximation technique to a partial differential equation)
$\boldsymbol{u}_{\mathbf{T}}$	total velocity, $\boldsymbol{u}_{\mathbf{w}} + \boldsymbol{u}_{\mathbf{n}}$ in two-phase flow
V	volume
WI	productivity coefficient (proportional to productivity index)
x	distance
x_i	value of x at grid point i
y	distance
Z	compressibility factor
z	distance
$\alpha = \Delta t/h^2$	coefficient
β	turbulence factor, eqn. (2.96)
$\beta = \phi \frac{c_f}{B^o} + \phi^o \frac{c_R}{B}$	coefficient of time derivative
Γ	reservoir boundary

$\gamma = \rho g/g_c$	density in terms of pressure/distance
$\Delta\gamma = \gamma_w - \gamma_n$	density difference
$\lambda = k/(\mu B)$	transmissibility
$\lambda = \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n}$	mean mobility
λ_i	eigenvalues
$\lambda_l = kk_{rl}/(\mu_l B_l)$	transmissibility of phase l
$\lambda_l = kk_{rl}/\mu_l$	mobility of phase l
λ_{max}	maximum modulus of eigenvalues
$\lambda R = k_r/(\mu B)$	radial transmissibility
$\lambda_T = k\left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} + \frac{k_{rg}}{\mu_g}\right)$	total mobility
$\lambda X = k_x/(\mu B)$	x-direction transmissibility
$\lambda Y = k_y/(\mu B)$	y-direction transmissibility
$\lambda Z = k_z/(\mu B)$	z-direction transmissibility
μ	viscosity
ν	level of iteration
ξ_m	amplification factor, eqn. (3.51)
ρ	fluid density
$\rho = \ln r$	transformed radial co-ordinate, Chapter 3
$\rho(\mathbf{B})$	spectral radius of matrix **B**
ρ_l	density of phase l
O	order of approximation
θ	angle
ϕ	porosity
$\phi = \int_{p^o}^{p} \frac{dp}{\gamma} - z$	pseudo-potential
ψ	pseudo-pressure, eqn. (2.52)
Ω	reservoir boundary
ω	mixing parameter, Chapter 12
ω	relaxation factor in the SOR method
ω_b	optimum value of ω in the SOR method
ω_{li}	mass fraction of component i in phase l
ω_i	mass fraction of component i in the mixture

Operators

A	coefficient matrix of a system of algebraic equations
A	differential operator for Cartesian co-ordinates
B	coefficient matrix of $\mathbf{u}^n$, eqn. (3.54)
C	coefficient matrix for boundary value problems of the fourth kind
C	differential operator for cylindrical co-ordinates
D	accumulation matrix
E	symmetric tridiagonal matrix with 2's in the main diagonal and – l's for the sub- and super-diagonals
G	vector of gravity terms
I	identity matrix
J	Jacobian
L	lower triangular matrix for *LU* factorisation
L	finite-difference operator for Cartesian Co-ordinates
M	finite-difference operator in cylindrical co-ordinates
Q	source vector
S	symmetric tridiagonal matrix, Chapter 4
T	transmissibility matrix
U	upper triangular matrix for *LU* factorisation, Chapter 8
Δ	difference operator
Δ^2	difference operator for second space derivative
Δ_s	grid spacing of co-ordinate s (s = x, y, z, r, etc.)
Δ_t	difference operator for time derivative

Subscripts

dg	dissolved gas
f	fluid
fg	free gas
i	initial, Chapter 12
$i \pm ½$	boundaries of a block containing point i
i	grid point i
J	Jacobi matrix
l	component or phase, l = o, g, w (oil, gas, water)
N	index of space grid point corresponding to the last unknown
n	non-wetting phase
RC	reservoir conditions
R	rock

r, θ, z	directions in the cylindrical co-ordinate system
s	solvent, Chapter 12
sf	sandface
STC	stock tank or standard conditions
T	total
w	wetting or water phase
x, y, z	directions in the Cartesian co-ordinate system

Superscripts

b	backward difference
f	forward difference
L	logarithmic
n	time level, $n = 0, 1, 2, 3, \ldots$
o	initial conditions ($t = 0$) or reference conditions
p	order of a finite difference approximation
r	reference
T	matrix or vector transpose
2	centred
*	intermediate or perturbed solution
‘	$\frac{\mathrm{d}}{\mathrm{dx}}$ or $\frac{\partial}{\partial x}$
“	$\frac{\mathrm{d}^2}{\mathrm{dx}^2}$ or $\frac{\partial^2}{\partial x^2}$
.	$\frac{\partial}{\partial t}$
–	depth-averaged pseudo value for VE calculations, Chapter 12

Abbreviations

GOR	gas–oil ratio
LSOR	line SOR
SOR	successive over-relaxation
WOR	water–oil ratio
SIP	strongly implicit procedure
1DC	one-dimensional correction
2DC	two-dimensional correction
PI	productivity index
1-D	one dimensional
2-D	two dimensional
3-D	three dimensional
ODE	ordinary differential equation(s)

PDE	partial differential equation(s)
IMPES	implicit pressure-explicit saturation
SS	simultaneous solution
SEQ	sequential solution
VE	vertical equilibrium
w–n	wetting–non-wetting
C–N	Crank–Nicolson
D2	ordering scheme
D4	ordering scheme
WI	productivity coefficient

CHAPTER 1

INTRODUCTION

1.1 WHAT IS A COMPUTER MODEL?

The primary objective of a reservoir study is to predict future performance of a reservoir and find ways and means of increasing ultimate recovery. Classical *reservoir engineering* deals with the reservoir on a gross average basis (tank model) and cannot account adequately for the variations in reservoir and fluid parameters in space and time. *Reservoir simulation* by computers allows a more detailed study of the reservoir by dividing the reservoir into a number of blocks (sometimes several thousand) and applying fundamental equations for flow in porous media to each block. Digital computer programs that perform the necessary calculations to do such model studies are called *computer models*. Because of the advances made since the early 1950s in computer hardware and software technology, it is now possible to write rather sophisticated models to simulate some of the very complex processes that take place in reservoirs during the implementation of recovery schemes. Reservoir simulation technology is being constantly improved and enhanced. New models to simulate more and more complex recovery schemes are being proposed all the time. In this book, we deal with the most basic of all reservoir models, known as the *black-oil model* or *beta-model*. A thorough understanding of the techniques used for black-oil models is essential in order to develop some appreciation for more complex models. In the description of computer models terms like *mathematical models, numerical models, numerical simulators, grid models, finite-difference models* and *reservoir simulators* are used almost interchangeably. In reality, there are three kinds of models involved in developing a program to simulate a reservoir:

1.1.1 Mathematical Model

The physical system to be modelled must be expressed in terms of appropriate mathematical equations. This process almost always involves assumptions. The assumptions are necessary from a practical standpoint in order to make the problem tractable. For example, every reservoir engineer knows that the concept of relative permeability has limitations, but in the absence of anything else, we have no choice but

to use it. The formulation of mathematical models is considered in Chapter 2; it results in a set of nonlinear partial differential equations with appropriate initial and boundary conditions.

1.1.2 Numerical Model

The equations constituting a mathematical model of the reservoir are almost always too complex to be solved by analytical methods. Approximations must be made to put the equations in a form that is amenable to solution by digital computers. Such a set of equations forms a numerical model. This is discussed in Chapters 3 to 12.

1.1.3 Computer Model

A computer program or a set of programs written to solve the equations of the numerical model constitutes a computer model of the reservoir. Some practical aspects of computer models are discussed in Chapter 13. The use of a computer model to solve practical problems will be referred to as 'reservoir simulation' in this book.

1.2 OTHER MODELS

Many other kinds of models have been used by petroleum engineers. They may be divided into two categories, (a) *analog models*, and (b) *physical models*. The most common analog models are the electrical models, where electrical potential and current serve as the analog variables. Discrete electrical models (R–C and R–R networks) which are analogs of finite-difference equations, have been applied to reservoir problems by Bruce (1943) and Karplus (1956). Continuous models of electrolytic type are discussed by Botset (1946). Comprehensive discussion of these and other analog computer methods can be found in a text by Karplus (1958). However, analog methods have now been completely replaced by computer models.

The literature on physical models is extensive (Rapoport, 1955; Geertsma *et al*., 1956; Perkins and Collins, 1960; Redford *et al*., 1976), and they play a key rôle in understanding reservoir behaviour. Physical models can be classified into two categories (cf Redford *et al*., 1976), (a) *scaled models*, and (b) *elemental models*. In a scaled model, reservoir dimensions, fluid properties and rock properties are scaled for the laboratory model so that the ratio of various forces in the reservoir and the physical model are the same. A scaled model would provide results that could be directly applied to the field. Unfortunately, fully scaled models are difficult or impossible to construct (Geertsma *et al*., 1956; Pozzi and Blackwell, 1963).

In an elemental model, experiments are conducted with actual (or simulated) reservoir rock and fluids. Obviously results of such a model are not directly applicable

to the field, but they can help answer some basic questions about reservoir mechanics.

The basic fluid flow equations that describe flow in the reservoir (mathematical model) are also valid for scaled and elemental models. This means that a computer model can be verified and even adjusted by use of results from physical models and then used to predict field performance. Thus, maximum understanding of a complex reservoir phenomenon may require the judicious use of both physical and computer models. It should be clear that computer models cannot eliminate the need for physical models, since they cannot be used to determine the physics of the problem. On the other hand, optimum use of data from physical models is in many cases possible only through computer models. In conclusion, it would be fair to say that computer models of petroleum reservoirs cannot replace all physical models. Computer models can, however, enhance the understanding of data obtained by physical modelling, and aid in the design of experiments done on physical models.

1.3 WHAT QUESTIONS CAN A COMPUTER MODEL ANSWER?

Computer models can be valuable tools for the petroleum engineer attempting to answer questions of the following type:

1. How should a field be developed and produced in order to maximize the economic recovery of hydrocarbons?
2. What is the best enhanced recovery scheme for the reservoir? How and when should it be implemented?
3. Why is the reservoir not behaving according to predictions made by previous reservoir engineering or simulation studies?
4. What is the ultimate economic recovery for the field?
5. What type of laboratory data is required? What is the sensitivity of model predictions to various data?
6. Is it necessary to do physical model studies of the reservoir? How can the results be scaled up for field applications?
7. What are the critical parameters that should be measured in the field application of a recovery scheme?
8. What is the best completion scheme for wells in a reservoir?
9. From what portion of the reservoir is the production coming?

These are some general questions; many more specific questions may be asked when one is considering a particular simulation study. Defining the objectives of the study to be conducted and carefully stating the questions to be answered is an extremely important step in conducting any simulation study (see Chapter 13).

1.4 CONCLUDING REMARKS

Reservoir simulation is a tool that allows the petroleum engineer to gain greater insight into the mechanism of petroleum recovery than is otherwise possible. It can, if properly used, be a most valuable tool. It does not, however, replace good engineering judgement that is essential for conducting all reservoir studies (cf. Coats, 1969; Staggs and Herbeck, 1971). Furthermore, not all reservoirs require a sophisticated model study and in many cases conventional reservoir studies or extremely simple computer model studies may answer the questions being raised. It is easy to generate numbers by a computer model; in most cases the correct interpretation of the numbers requires a careful analysis by someone who understands the mathematical, numerical and the computer model. It is the objective of this book to present background material for such an understanding.

CHAPTER 2

FLUID FLOW EQUATIONS

2.1 INTRODUCTION

Before simulating a petroleum reservoir on a computer, a mathematical model of the system is required. Development of such a model is the objective of this chapter. Fluid motions in porous media are governed by the same fundamental laws that govern their motion in, for example, the atmosphere, pipelines and rivers. These laws are based on the conservation of mass, momentum and energy and are discussed in detail in numerous books including Bird *et al*. (I960), Schlichting (1968), and Monin and Yaglom (1971). From a practical standpoint it is hopeless at this time to try to apply these basic laws directly to the problems of flow in porous media. Instead, a semiempirical approach is used where Darcy's Law is employed instead of the momentum equation. The theoretical bases of the empirical law of Darcy are reviewed by Whitaker (1966, 1969); such studies provide an understanding of the limitations of empirical relations. In addition to the relations discussed above, the physical properties of the fluids involved in the system must also be known as a function of the dependent variables. This book deals only with some of the mathematical models which are known to be of practical significance. Numerical methods for the solution of equations resulting from these models will be discussed in Chapter 3. A brief development of the equations to be solved later will be presented here (Section 2.2). The discussion will be restricted to isothermal flow of a single fluid, or multiphase flow of up to three immiscible fluids. In this context, the following single and multiphase systems are of practical importance: gas; oil; water; gas–oil; gas–water; oil–water; oil–water–gas.

The first two books dealing with the mechanics of fluid flow in porous media were published by Muskat (1937, 1949). These books are of great historical importance and contain many of Muskat's own contributions. A book on the theory of groundwater movement was published in the USSR by Polubarinova-Kochina (1962). This book deals with those single-phase flow problems where analytic solutions are possible. A survey book on the physics of flow in porous media was published by Scheidegger (1974). This book deals briefly with a selection of topics related to the recovery of petroleum from underground reservoirs; it is designed as a reference for research

workers. The book by Collins (1961) deals with the theoretical and practical aspects of petroleum reservoir engineering.

The Society of Petroleum Engineers of AIME has published three monographs: two deal with the application of fluid flow principles to pressure build-up and flow tests (Matthews and Russell, 1967; Earlougher, 1977); the third monograph by Craig (1971) provides practical treatment of the problem of waterflooding petroleum reservoirs.

Bear (1972) provides a complete treatment of the dynamics and statics of fluids in porous media. However, most of the problems considered in the book by Bear are oriented towards groundwater hydrology. The application of fluid flow theory to the testing of gas wells is provided in a publication of the Energy Resources Conservation Board of Alberta (ERCB, 1975).

2.2 THE LAW OF MASS CONSERVATION

2.2.1 Single-Phase Flow

Consider the flow of a single fluid (single component or a homogeneous mixture) in the axial direction in a cylindrical core as shown in Fig. 2.1. The control volume must be representative of the porous medium (see Bear, 1972; p. 19), i.e., it should be large compared to the size of the pores but small compared to the size of the core. The basic physical properties of the porous medium, like the porosity, may be associated with the control volume. If porosity is defined as a fraction of the control volume not occupied by the solid matrix, then we can see that if the control volume is of the size of a pore, the porosity would be either one or zero. As we increase the size of the control volume, the porosity values will fluctuate before reaching a representative value. *The value of porosity associated with a point P is the representative value for a control volume containing point P.* Other physical properties are defined at a point in the porous medium in the same manner. This is the *continuum* approach, where the actual porous medium is replaced by a fictitious continuum to any point of which we can assign variables and parameters which are continuous functions of the space and time co-ordinates.

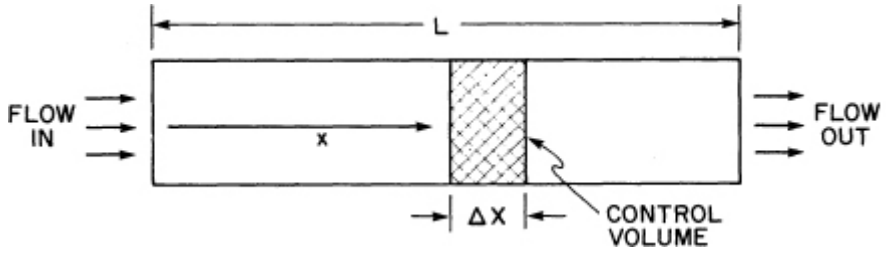


FIG. 2.1. Linear flow in a cylindrical porous rock of length Δx.

Let $\dot{m}_x$ be the x-component of the mass flux vector (mass flow per unit area per unit time) of a fluid of density ρ (single phase, single component). Referring to Fig. 2.1 we see that the mass inflow across the control volume surface at x over a time interval Δt is

$$\dot{m}_x|_x A\,\Delta t$$

and the mass outflow across control volume surface at $x + \Delta x$ over a time interval Δt is

$$\dot{m}_x|_{x+\Delta x} A\,\Delta t$$

The difference between inflow and outflow must be equal to the sum of accumulation of mass within the control volume. Mass accumulation due to compressibility over a time interval Δt is

$$\left[\frac{\partial}{\partial t}(\rho\phi\,\Delta V)\right]\Delta t$$

and the removal of mass from the control volume, i.e. mass depletion (accumulation) due to a sink of strength $\tilde{q}$ (mass per unit volume per unit time) over a time interval Δt is

$$\tilde{q}\,\Delta V\,\Delta t$$

We now have

$$(\dot{m}_x|_x - \dot{m}_x|_{x+\Delta x})A\,\Delta t = \left[\frac{\partial}{\partial t}(\rho\phi\,\Delta V)\right]\Delta t + \tilde{q}\,\Delta V\,\Delta t \tag{2.1}$$

Dividing through by $\Delta V\,\Delta t$ and noting that $\Delta V = A\,\Delta x$, we obtain:

$$\frac{\dot{m}_x|_x - \dot{m}_x|_{x+\Delta x}}{\Delta x} = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q}$$

Taking the limit as $\Delta x \to 0$ we have the equation of mass conservation for this system

$$-\frac{\partial \dot{m}_x}{\partial x} = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q} \tag{2.2}$$

Note that $\tilde{q}$ is negative for a source since we have assumed it to be positive for a sink.

It is possible to express mass flux in terms of a superficial (or Darcy) velocity:

$$\dot{m}_x = \rho u_x \tag{2.3}$$

where u_x is a velocity in the x-direction defined by eqn. (2.3). Substituting eqn. (2.3) into eqn. (2.2) we obtain

$$-\frac{\partial \rho u_x}{\partial x} = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q} \tag{2.4}$$

The corresponding equation for three-dimensional flow in a porous medium of arbitrary shape may be derived in a similar fashion by considering a control volume $\Delta x\, \Delta y\, \Delta z$. This leads to

$$-\left(\frac{\partial}{\partial x}\rho u_x + \frac{\partial}{\partial y}\rho u_y + \frac{\partial}{\partial z}\rho u_z\right) = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q}$$

for the Cartesian system of co-ordinates. More generally, the equation may be written as

$$-\nabla . \rho \boldsymbol{u} = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q} \tag{2.5}$$

The divergence operator on the left-hand side of eqn. (2.5) may be expanded in any co-ordinate system. For example, in cylindrical co-ordinates (r, θ, z) the conservation equation is:

$$-\left(\frac{1}{r}\frac{\partial \rho r u_r}{\partial r} + \frac{1}{r}\frac{\partial \rho u_\theta}{\partial \theta} + \frac{\partial \rho u_z}{\partial z}\right) = \frac{\partial}{\partial t}(\rho\phi) + \tilde{q} \tag{2.6}$$

2.2.2 Multiphase Flow

The conservation for single-phase flow (eqn. 2.5) may be generalized as follows:

$$-\nabla . \dot{\boldsymbol{m}}_l = \frac{\partial (m_l)}{\partial t} + \tilde{q}_l \tag{2.7}$$

where m_l is mass of component l in a unit volume of the medium, $\dot{\boldsymbol{m}}_l$ is the mass flux of component l and $\Delta . \dot{\boldsymbol{m}}_t$ or div $\dot{\boldsymbol{m}}_l$, is the rate of mass efflux per unit volume.

There are two important mathematical models in petroleum reservoir engineering: (1) multiphase or single-phase flow where more than two hydrocarbon components are considered and (2) multiphase flow where the hydrocarbon system can be approximated by two components, a non-volatile component (black oil) and a volatile component

(gas) soluble in the oil phase. We will consider the second case almost exclusively—this is known as the *β-model* or the *black-oil model*. The variable composition system will be considered only briefly in Chapter 12.

2.2.2.1 The *β-Model*

In this model of the fluid flow problem it is assumed that there are at most three distinct phases: oil, water and gas. Usually water is the wetting phase, oil has an intermediate wettability and gas is the nonwetting phase. Water and oil are assumed to be immiscible and they do not exchange mass or change phase. Gas is assumed to be soluble in oil but usually not in water. If we assume that the solubility of gas is zero at stock tank conditions then reservoir oil may be considered to be a solution of two components: stock tank oil and gas at standard conditions. Furthermore, in this kind of treatment it is assumed that the fluids are at constant temperature and in thermodynamic equilibrium throughout the reservoir. Under these conditions the pressure–volume–temperature (PVT) behaviour of the system may be expressed by the formation volume factors as defined below:

$$B_o = \frac{[V_o + V_{dg}]_{RC}}{[V_o]_{STC}} = f(p_o) \tag{2.8}$$

$$B_w = \frac{[V_w]_{RC}}{[V_w]_{STC}} = f(p_w) \tag{2.9}$$

$$B_g = \frac{[V_g]_{RC}}{[V_g]_{STC}} = f(p_g) \tag{2.10}$$

In the above equations $[V_l]_{RC}$ stands for the volume occupied by a fixed mass of component l (o, w or g) at reservoir conditions and $[V_l]_{STC}$ is the volume occupied by the same component at stock tank (or standard) conditions. We note that some authors prefer to work with inverted factors, i.e. $b_l = 1/B_l$. The mass transfer between oil and gas phases is described by the solution gas:oil ratio

$$R_s = \left[\frac{V_{dg}}{V_o}\right]_{STC} = f(p_o) \tag{2.11}$$

which gives the amount of gas dissolved in oil as a function of oil-phase pressure. The densities of the three phases at reservoir conditions are related to densities at stock tank conditions:

$$\rho_o = \frac{1}{B_o}(\rho_{oSTC} + R_s\rho_{gSTC}) \tag{2.12}$$

$$\rho_w = \frac{1}{B_w}(\rho_{wSTC}) \tag{2.13}$$

$$\rho_g = \frac{1}{B_g}(\rho_{gSTC}) \tag{2.14}$$

The density of the oil phase may also be expressed as

$$\rho_o = \bar{\rho}_o + \bar{\rho}_{dg} \tag{2.15}$$

where ρ_o and ρ_{dg} are the densities of the two components

$$\bar{\rho}_o = \frac{1}{B_o}\rho_{oSTC} \tag{2.16}$$

$$\bar{\rho}_{dg} = \frac{R_s}{B_o}\rho_{gSTC} \tag{2.17}$$

Before considering multiphase flow the concept of saturation must be introduced. The saturation, S_l, of phase l is the fraction of the pore volume occupied by phase l. Obviously, $\Sigma_l S_l = 1$. The equation of mass conservation for each component can be written by considering eqn. (2.7).

For the oil component in the oil phase

$$\dot{\boldsymbol{m}}_o = \bar{\rho}_o \boldsymbol{u}_o \tag{2.18}$$

$$m_o = \bar{\rho}_o \phi S_o \tag{2.19}$$

Substituting eqns. (2.18) and (2.19) into (2.7) and dividing through by ρ_{oSTC} yields

$$-\nabla . \left[\frac{1}{B_o}\boldsymbol{u}_o\right] = \frac{\partial}{\partial t}\left[\frac{1}{B_o}\phi S_o\right] + q_o \tag{2.20}$$

where

$$q_o = \frac{\tilde{q}_o}{\rho_{oSTC}}$$

All terms in eqn. (2.20) have the dimension

$$\left(\frac{\text{STC volume}}{\text{RC volume}}\right) \cdot \frac{1}{(\text{time})}$$

The equation for the water phase is obtained in a similar fashion:

$$-\nabla \cdot \left[\frac{1}{B_w}\boldsymbol{u}_w\right] = \frac{\partial}{\partial t}\left[\frac{1}{B_w}\phi S_w\right] + q_w \tag{2.21}$$

The gas component exists both in the gas phase and in solution in the oil phase

$$\dot{\boldsymbol{m}}_g = \rho_g \boldsymbol{u}_g + \bar{\rho}_{dg}\boldsymbol{u}_o \tag{2.22}$$

$$m_g = \phi[S_g\rho_g + \bar{\rho}_{dg}S_o] \tag{2.23}$$

$$\tilde{q}_g = \tilde{q}_{fg} + \tilde{q}_o R_s \left(\frac{\rho_g}{\rho_o}\right)_{STC} = \tilde{q}_{fg} + q_o R_s \rho_{gSTC} \tag{2.24}$$

This yields the final gas equation

$$-\nabla \cdot \left[\frac{R_s}{B_o}\boldsymbol{u}_o + \frac{1}{B_g}\boldsymbol{u}_g\right] = \frac{\partial}{\partial t}\left[\phi\left(\frac{R_s}{B_o}S_o + \frac{1}{B_g}S_g\right)\right] + q_{fg} + R_s q_o \tag{2.25}$$

The production terms q_o, q_w, q_g represent volume produced at stock tank (or standard) conditions, per unit time per unit reservoir volume.

2.3 DARCY'S LAW

2.3.1 Single-Phase Flow

In addition to the equation of continuity or mass conservation developed in the last section, we require a relationship between the flow rate and pressure gradient in each phase. Such a relationship was discovered by Darcy (1856) for single-phase flow. The differential form of this relationship is

$$\boldsymbol{u} = -\frac{k}{\mu}\left(\nabla p + \rho\frac{\boldsymbol{g}}{g_c}\right) \tag{2.26}$$

where k is the absolute permeability tensor of the porous media, μ is the fluid viscosity, $\mathbf{g}$ is the gravitational acceleration vector and g_c is a conversion constant with the units of lb_m/lb_f ft/sec^2 in the Engineering System of Units. If the co-ordinate in the vertical downward direction is z then we can write

$$\rho\frac{\boldsymbol{g}}{g_c} = -\rho\frac{g}{g_c}\nabla z = -\gamma\nabla z \tag{2.27}$$

With the above definition of γ we can write Darcy's Law as

$$\boldsymbol{u} = -\frac{k}{\mu}(\nabla p - \gamma\nabla z) \tag{2.28}$$

When $\boldsymbol{u} = 0$, the above equation yields the static head relationships. In Cartesian co-ordinates with z-axis vertical and oriented downwards they are:

$$\frac{\partial p}{\partial z} = \gamma \tag{2.29}$$

$$\frac{\partial p}{\partial x} = \frac{\partial p}{\partial y} = 0 \tag{2.30}$$

The permeability tensor used in eqn. (2.26) is defined by that equation and it must be determined experimentally. In most practical problems it is possible (or necessary) to assume that k is a diagonal tensor given by

$$k = \begin{bmatrix} k_x & & \\ & k_y & \\ & & k_z \end{bmatrix}$$

If $k_x = k_y = k_z$, the medium is called isotropic, otherwise it is anisotropic. The limitations of Darcy's Law are fully discussed in the literature (e.g. Hubbert, 1956; Scheidegger, 1974; Collins, 1961; Whitaker, 1966, 1969) and will not be considered here.

2.3.2 Multiphase Flow

The law may be extended to describe the simultaneous flow of more than one phase:

$$\boldsymbol{u}_l = -\frac{kk_{rl}}{\mu_l}\left(\nabla p_l + \rho_l \frac{\boldsymbol{g}}{g_c}\right) \tag{2.31}$$

where l = o,w,g (oil, water and gas phases, respectively) and k_{rl} is the relative permeability of phase l. Equation (2.31) may also be written in terms of γ_l

$$\boldsymbol{u}_l = -\frac{kk_{rl}}{\mu_l}(\nabla p_l - \gamma_l \nabla z) \tag{2.32}$$

where

$$\gamma_l = \rho_l \frac{g}{g_c} \tag{2.33}$$

and z is positive in the vertical downward direction. If the velocity is in cm/sec, viscosity in centipoise and pressure gradient in atm/cm then the unit of k is darcy. It can be shown that

$$\begin{aligned} 1 \text{ darcy} &= 9{\cdot}869 \times 10^{-9} \text{ cm}^2 \\ &= 1{\cdot}062 \times 10^{-11} \text{ ft}^2 \end{aligned}$$

Often the unit millidarcy or md (1 darcy = 1000 millidarcy) is used.

2.4 THE BASIC FLOW EQUATIONS

The flow equations for single and multiphase flow are obtained by combining appropriate forms of Darcy's Law and the equation of mass conservation. The fluid density is expressed explicitly or implicitly as a function of pressure through an equation of state. Several different practical situations will be considered here.

2.4.1 Single-Phase Flow

2.4.1.1 *General Equation for Compressible Fluids*

When the entire pore space is occupied by a single phase, eqn. (2.28) can be substituted into eqn. (2.5) to obtain

$$\nabla . \frac{\rho k}{\mu}(\nabla p - \gamma \nabla z) = \frac{\partial}{\partial t}(\rho \phi) + \tilde{q} \tag{2.34}$$

Dividing through by ρ_{STC} and using the definition of $B = [V]_{RC}/[V]_{STC}$ we have

$$\nabla . [\lambda(\nabla p - \gamma \nabla z)] = \frac{\partial}{\partial t}\left[\frac{\phi}{B}\right] + q \tag{2.35}$$

where

$$\lambda = \frac{1}{\mu B} k \tag{2.36}$$

2.4.1.2 *Equation for Slightly Compressible Fluid*

For liquid flow it is often possible to assume that the fluid compressibility defined by

$$c_f = -\frac{1}{V}\frac{\partial V}{\partial p}\bigg|_T = \frac{1}{\rho}\frac{\partial \rho}{\partial p}\bigg|_T \tag{2.37}$$

is constant over the range of pressure of interest. This equation can be then integrated to yield

$$\rho = \rho^\circ \exp\left[c_f(p - p^\circ)\right] \tag{2.38}$$

where ρ° is the density at the reference pressure ρ°. From the definition of the formation volume factor we see that:

$$\frac{\rho}{\rho^\circ} = \frac{B^\circ}{B} = \exp\left[c_f(p - p^\circ)\right] \tag{2.39}$$

$$= 1 + c_f(p - p^\circ) + \frac{1}{2!}c_f^2(p - p^\circ)^2 + \cdots$$

where B° is the formation volume factor at p°.

Considering only the first two terms of the expansion we have

$$B = B^\circ/[1 + c_f(p - p^\circ)] \tag{2.40}$$

This is justified because c_f is small (~10^{-5} to 10^{-6} 1/psia).

If the variation of pore volume with pressure is significant, it may be accounted for by (see eqn. 2.121)

$$\phi = \phi^\circ[1 + c_R(p - p^\circ)] \tag{2.41}$$

where c_R is rock compressibility.

The time derivative term in eqn. (2.35) may be expressed in terms of $\partial p/\partial t$ using the expression for $1/B$ given by eqn. (2.40) and for ϕ by eqn. (2.41). When this is done, eqn. (2.35) becomes

$$\nabla . [\lambda(\nabla p - \gamma \nabla z)] = \left(\phi \frac{c_f}{B^\circ} + \phi^\circ \frac{c_R}{B}\right)\frac{\partial p}{\partial t} + q \tag{2.42}$$

Another useful form of the flow equation is obtained by substituting eqn. (2.38) into (2.34) and neglecting terms involving the square of the pressure gradient multiplied by c_f in comparison to other terms in the equation. The resulting equation is (cf. ERCB, 1975):

$$\nabla^2 p = \frac{\phi \mu c_f}{k}\frac{\partial p}{\partial t} + \frac{\mu}{\rho k}\tilde{q} \tag{2.43}$$

In writing the above equation, which is known as the 'diffusivity equation' (Carslaw and Jaeger, 1959), we have also assumed that fluid properties are constant, $c_R = 0$, and that gravitational terms are negligible.

2.4.1.3 *Equations for Gas Flow*

For gas flow it is usually not appropriate to assume a constant compressibility. For such situations the flow equation may be written as

$$\nabla . [\lambda(\nabla p - \gamma \nabla z)] = c(p)\frac{\partial p}{\partial t} + q \tag{2.44}$$

where

$$c(p) = \phi \frac{\mathrm{d}}{\mathrm{d}p}\left(\frac{1}{B}\right) + \phi^\circ \frac{c_R}{B} \tag{2.45}$$

Another form of the equation is obtained if we use the Gas Law

$$\rho = \frac{p\mathrm{M}}{ZRT} \tag{2.46}$$

Substituting eqn. (2.46) into eqn. (2.34) and neglecting gravitational terms, which are normally small for gas flow, we have

$$\nabla \cdot \left[\frac{p}{\mu Z}\nabla p\right] = \frac{\phi}{k}\frac{\partial}{\partial t}\left[\frac{p}{Z}\right] + \frac{\mathrm{R}T}{\mathrm{M}k}\tilde{q} \tag{2.47}$$

In the above equation we have assumed ϕ and k to be constant. Noting that $2p.\ \Delta p = \Delta p^2$, we can write eqn. (2.47) as (Al-Hussainy *et al*., 1966):

$$\nabla^2 p^2 - \frac{\mathrm{d}}{\mathrm{d}p^2}[\ln(\mu Z)](\nabla p^2)^2 = \frac{2\phi\mu Z}{k}\frac{\partial}{\partial t}\left[\frac{p}{Z}\right] + 2\frac{ZRT\mu}{\mathrm{M}k}\tilde{q} \tag{2.48}$$

The derivative on the right side may be written as

$$\frac{\partial}{\partial t}\left(\frac{p}{Z}\right) = \frac{pc}{Z}\frac{\partial p}{\partial t} \tag{2.49}$$

where

$$\begin{aligned} c &= \frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}p}\bigg|_{\mathrm{T}} \\ &= \frac{1}{p} - \frac{1}{Z}\frac{\mathrm{d}Z}{\mathrm{d}p} \end{aligned} \tag{2.50}$$

Substituting eqn. (2.49) into eqn. (2.48) and neglecting the second term on the left side of eqn. (2.48), we obtain

$$\nabla^2 p^2 = \frac{\phi\mu c}{k}\frac{\partial p^2}{\partial t} + \frac{2ZRT\mu}{\mathrm{M}k}\tilde{q} \tag{2.51}$$

It is also possible to obtain the above equation by other procedures.

A more rigorous equation for compressible flow is obtained by defining a pseudo-pressure ψ (Al-Hussainy *et al*., 1966), as

$$\psi = 2\int_{p_b}^{p}\frac{p}{\mu Z}\mathrm{d}p \tag{2.52}$$

Since

$$\nabla\psi = \frac{\mathrm{d}\psi}{\mathrm{d}p}\nabla p = \frac{2p}{\mu Z}\nabla p$$

and

$$\frac{\partial\psi}{\partial t} = \frac{\mathrm{d}\psi}{\mathrm{d}p}\frac{\partial p}{\partial t} = \frac{2p}{\mu Z}\frac{\partial p}{\partial t}$$

the original eqn. (2.47) is transformed into

$$\nabla^2\psi = \frac{\phi\mu c}{k}\frac{\partial\psi}{\partial t} + \frac{2\mathrm{R}T}{\mathrm{M}k}\tilde{q} \tag{2.53}$$

Note that this equation is of the same form as eqn. (2.51), and it does not involve additional simplifying assumptions. A more complete discussion of single-phase gas-flow equations is available in a manual published by the Energy Resources Conservation Board of Alberta (ERCB, 1975).

2.4.2 Multiphase Flow

Darcy's Law (eqn. 2.32) may be substituted into the mass conservation equation for each phase (eqns. 2.20, 2.21 and 2.25) to obtain fluid flow equations:

$$\nabla\cdot[\lambda_{\mathrm{o}}(\nabla p_{\mathrm{o}} - \gamma_{\mathrm{o}}\nabla z)] = \frac{\partial}{\partial t}\left[\frac{\phi S_{\mathrm{o}}}{B_{\mathrm{o}}}\right] + q_{\mathrm{o}} \tag{2.54}$$

$$\nabla\cdot[\lambda_{\mathrm{w}}(\nabla p_{\mathrm{w}} - \gamma_{\mathrm{w}}\nabla z)] = \frac{\partial}{\partial t}\left[\frac{\phi S_{\mathrm{w}}}{B_{\mathrm{w}}}\right] + q_{\mathrm{w}} \tag{2.55}$$

$$\begin{aligned}\nabla\cdot[R_{\mathrm{s}}\lambda_{\mathrm{o}}(\nabla p_{\mathrm{o}} - \gamma_{\mathrm{o}}\nabla z) + \lambda_{\mathrm{g}}(\nabla p_{\mathrm{g}} - \gamma_{\mathrm{g}}\nabla z)] \\ = \frac{\partial}{\partial t}\left[\phi\left(\frac{R_{\mathrm{s}}}{B_{\mathrm{o}}}S_{\mathrm{o}} + \frac{S_{\mathrm{g}}}{B_{\mathrm{g}}}\right)\right] + R_{\mathrm{s}}q_{\mathrm{o}} + q_{\mathrm{fg}}\end{aligned} \tag{2.56}$$

where transmissibilities λ_l are defined by

$$\lambda_l = \frac{k_{rl}}{\mu_l B_l}k$$

While the conservation equation is sufficient to describe single-phase flow (the only dependent variable being p), this is not the case for multiphase flow. Equations (2.54)

to (2.56) contain six dependent variables. Three additional relations are needed to complete the description:

$$S_o + S_w + S_g = 1 \tag{2.57}$$

$$P_{cow} = p_o - p_w = f(S_w, S_g) \tag{2.58}$$

$$P_{cog} = p_g - p_o = f(S_w, S_g) \tag{2.59}$$

The relationships between capillary pressures and saturations are usually empirical.

2.4.3 Use of Pseudopotential

It is often convenient to express the conservation equations in a form that does not explicitly involve the gravity terms. This is accomplished by the definition of a 'potential', introduced by Hubbert (1940, 1956).

Define

$$\Phi = \int_{p^\circ}^{p} \frac{dp}{\gamma} - z \tag{2.60}$$

then Darcy's Law can be written as

$$\boldsymbol{u} = -\frac{k}{\mu}(\nabla p - \gamma \nabla z) = -\frac{k}{\mu}\gamma \nabla \Phi \tag{2.61}$$

and the flow equations are formally simplified. For example, the single-phase equation (eqn. (2.44)) becomes

$$\nabla . [\lambda \gamma \nabla \Phi] = c(p)\gamma \frac{\partial \Phi}{\partial t} + q$$

and eqn. (2.54) becomes

$$\nabla[\lambda_o \gamma_o \nabla \Phi_o] = \frac{\partial}{\partial t}\left[\phi \frac{S_o}{B_o}\right] + q_o$$

Only for incompressible flow can we use the true potential (piezometric head)

$$\Phi' = p - \gamma z$$

which is then related to Φ by

$$\Phi' = \gamma\Phi$$

2.4.4 Boundary Conditions

The mathematical model discussed so far is not complete without the necessary boundary and initial conditions. It is, however, instructive to present a discussion of them in later chapters along with the finite difference representation (numerical model).

Boundary conditions for single-phase flow are introduced in Section 3.4 of Chapter 3 and further discussed in Sections 7.4 and 7.7 of Chapter 7. Boundary conditions for multiphase flow are introduced in Chapter 5, Section 5.7 and further discussed in Chapter 9, Sections 9.4 and 9.8. Initial conditions are discussed in Chapter 9, Section 9.5.

2.5 ALTERNATIVE FORMS OF MULTIPHASE FLOW EQUATIONS

Several alternative formulations of the flow equations given in the previous section will be derived here. For greater clarity, the development is restricted to a two-phase system, with subscripts 'w' and 'n' denoting the wetting and non-wetting phases, respectively. The two-phase formulation in four variables is in this notation

$$\nabla . [\lambda_w(\nabla p_w - \gamma_w \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{S_w}{B_w}\right] + q_w \tag{2.62}$$

$$\nabla . [\lambda_n(\nabla p_n - \gamma_n \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{S_n}{B_n}\right] + q_n \tag{2.63}$$

$$P_c = p_n - p_w = f(S_w) \tag{2.64}$$

$$S_w + S_n = 1 \tag{2.65}$$

2.5.1 Formulation in 'Parabolic' Form

Formulation in p_w, p_n

Suppose that a unique inverse function to $P_c(S_w)$ exists, i.e.,

$$S_w = F_w(P_c) = F_w(p_n - p_w) \tag{2.66}$$

Function F_w exists if P_c is monotonically increasing or monotonically decreasing. Then eqns. (2.62) and (2.63) can be expressed as

$$\nabla . [\lambda_w(\nabla p_w - \gamma_w \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{F_w}{B_w}\right] + q_w \tag{2.67}$$

$$\nabla . [\lambda_n(\nabla p_n - \gamma_n \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{(1 - F_w)}{B_n}\right] + q_n \tag{2.68}$$

Equations (2.67) and (2.68) are the basis of the method called 'simultaneous solution method' in petroleum literature (Douglas *et al*. 1959; Coats, 1968a; Sheffield, 1969). The equations remain coupled regardless of the treatment of nonlinearities and a 'dummy' P_c function must be employed for the simulation of zero capillarity.

Formulation in p_n, P_c

This formulation is similar to the previous case and can be written as

$$\nabla . [\lambda_w(\nabla p_n - \nabla P_c - \gamma_w \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{F_w}{B_w}\right] + q_w \tag{2.69}$$

$$\nabla . [\lambda_n(\nabla p_n - \gamma_n \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{(1 - F_w)}{B_n}\right] + q_n \tag{2.70}$$

An equivalent formulation can also be written in terms of p_w and P_c.

Formulation in p_n, S_w

When p_w is expressed as $p_n - P_c$ and eqn. (2.64) is used, one obtains

$$\nabla . [\lambda_w(\nabla p_n - P_c' \nabla S_w - \gamma_w \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{S_w}{B_w}\right] + q_w \tag{2.71}$$

$$\nabla . [\lambda_n(\nabla p_n - \gamma_n \nabla z)] = \frac{\partial}{\partial t}\left[\phi \frac{(1 - S_w)}{B_n}\right] + q_n \tag{2.72}$$

The finite difference form of these equations can be decoupled under suitable assumptions. This is better seen if (2.71) and (2.72) are expressed in a different form. When eqn. (2.71) is multiplied by B_w, eqn. (2.72) by B_n and the equations are added, one obtains

$$B_n\nabla\cdot[\lambda_n(\nabla p_n-\gamma_n\nabla z)]+B_w\nabla\cdot[\lambda_w(\nabla p_n-\gamma_n\nabla z)] \qquad (2.73)$$
$$-B_w\nabla\cdot[\lambda_w(P'_c\nabla S_w+\Delta\gamma\nabla z)]$$
$$=B_n\left[(1-S_w)\frac{\partial}{\partial t}\left(\frac{\phi}{B_n}\right)+q_n\right]+B_w\left[S_w\frac{\partial}{\partial t}\left(\frac{\phi}{B_w}\right)+q_w\right]$$

where

$$\Delta\gamma=\gamma_w-\gamma_n \qquad (2.74)$$

Equation (2.73) is an alternate form of eqn. (2.71). Note that in the *p, S* formulation capillary-pressure function can be arbitrary, provided P'_c exists. In a finite difference form, if the saturations in eqn. (2.73) are taken explicitly, then λ_n, λ_w, P'_c and ΔS_w are known and $\partial/\partial t\ (\phi/B_w)$ may be taken as a function of p_n. Then the equations are decoupled as eqn. (2.73) may be solved for p_n. Equation (2.72) is then used to solve for S_w. This is known as the 'implicit pressure–explicit saturation' or IMPES method (Stone and Garder, 1961; Breitenbach *et al*., 1969) and is widely used in reservoir simulation. When the explicit treatment of saturations is not justifiable, as in the case of coning simulation, the equations remain coupled.

When $P_c=0$ ($p_n=p_w=p$), eqns. (2.73) and (2.72) simplify to

$$B_n\nabla\cdot[\lambda_n(\nabla p-\gamma_n\nabla z)]+B_w\nabla\cdot[\lambda_w(\nabla p-\gamma_w\nabla z)] \qquad (2.75)$$
$$=B_n[(1-S_w)\partial/\partial t(\phi/B_n)+q_n]+B_w[S_w\partial/\partial t(\phi/B_w)+q_w]$$

$$\nabla\cdot[\lambda_n(\nabla p-\gamma_n\nabla z)]=\frac{\partial}{\partial t}\left(\phi\frac{(1-S_w)}{B_n}\right)+q_n \qquad (2.76)$$

A different simplification results for incompressible flow in an incompressible medium, i.e., when B_w, B_n and ϕ are constants (B_n, B_w are not necessarily equal to one). Then the following equations result:

$$\nabla\cdot[(B_n\lambda_n+B_w\lambda_w)(\nabla p_n-\gamma_n\nabla z)]-\nabla\cdot[B_w\lambda_w(P'_c\nabla S_w+\Delta\gamma\nabla z)] \qquad (2.77)$$
$$=B_nq_n+B_wq_w$$

$$\nabla\cdot[B_n\lambda_n(\nabla p_n-\gamma_n\nabla z)]=-\phi\frac{\partial S_w}{\partial t}+B_nq_n \qquad (2.78)$$

Finally, for incompressible flow of fluids of equal density with $B_w = B_n = 1$ and without capillary forces, the classical equations (Muskat, 1937; Collins, 1961) are obtained:

$$\nabla . [(\lambda_n + \lambda_w)(\nabla p - \gamma \nabla z)] = q_n + q_w \tag{2.79}$$

$$\nabla . [\lambda_n(\nabla p - \gamma \nabla z)] = -\phi \frac{\partial S_w}{\partial t} + q_n \tag{2.80}$$

Obviously, equivalent formulations can be written in p_n, S_n; p_w, S_n or p_w, S_w.

2.5.2 Formulation in 'Hyperbolic' Form

This formulation is possible in a simple form only for incompressible flow. It was first used in linear waterflood calculations (Fayers and Sheldon, 1959) and re-derived more recently by Hiatt (1968). The general formulation given here can also be found in Bear (1972) and Spivak (1974).

The conservation equations for a two-phase system with negligible rock compressibility are

$$-\nabla . (\rho_w \boldsymbol{u}_w) = \phi \frac{\partial}{\partial t}(\rho_w S_w) + \tilde{q}_w \tag{2.81}$$

$$-\nabla . (\rho_n \boldsymbol{u}_n) = \phi \frac{\partial}{\partial t}[\rho_n(1 - S_w)] + \tilde{q}_n \tag{2.82}$$

After expansion of terms, division of equations by densities and adding them together, the equation for the total velocity $\boldsymbol{u}_T = u_w + u_n$ is obtained:

$$\nabla . \boldsymbol{u}_T = \nabla . (\boldsymbol{u}_w + \boldsymbol{u}_n) = -q_n - q_w + \phi \left[\frac{(S_w - 1)}{\rho_n} \frac{\partial \rho_n}{\partial t} - \frac{S_w}{\rho_w} \frac{\partial \rho_w}{\partial t} \right] - \frac{1}{\rho_n} \boldsymbol{u}_n . \nabla \rho_n - \frac{1}{\rho_w} \boldsymbol{u}_w . \nabla \rho_w \tag{2.83}$$

where $q_l = \tilde{q}_l / \rho_l$. In the incompressible case, all terms due to compressibility are zero and eqn. (2.83) simplifies to

$$\nabla . \boldsymbol{u}_T = -(q_n + q_w) = -q_T \tag{2.84}$$

Darcy's Law may be written as

$$\boldsymbol{u}_{\mathrm{w}} = -\lambda_{\mathrm{w}}(\nabla p_{\mathrm{w}} - \gamma_{\mathrm{w}}\nabla z), \qquad \boldsymbol{u}_{\mathrm{n}} = -\lambda_{\mathrm{n}}(\nabla p_{\mathrm{n}} - \gamma_{\mathrm{n}}\nabla z)$$

where

$$\lambda_l = \frac{kk_{rl}}{\mu_l} \qquad l = \mathrm{w}, \mathrm{n} \tag{2.85}$$

is the mobility † of phase l. The above equations can be combined to obtain the fractional flow equation

$$\boldsymbol{u}_{\mathrm{w}} = M\boldsymbol{u}_{\mathrm{n}} + \lambda_{\mathrm{w}}(\nabla P_c + \Delta\gamma\nabla z) \tag{2.86}$$

where $M = \lambda_{\mathrm{w}}/\lambda_{\mathrm{n}}$ is the mobility ratio and $\Delta\gamma = \gamma_{\mathrm{w}} - \gamma_{\mathrm{n}}$. The velocity $\boldsymbol{u}_{\mathrm{w}}$ can be replaced by $\boldsymbol{u}_{\boldsymbol{T}} - \boldsymbol{u}_{\mathrm{n}}$ in eqn. (2.86) to obtain

$$\boldsymbol{u}_{\mathrm{n}} = \frac{1}{1+M}[\boldsymbol{u}_{\mathrm{T}} - \lambda_{\mathrm{w}}(\nabla P_c + \Delta\gamma\nabla z)] \tag{2.87}$$

Finally, eqn. (2.87) is substituted in the conservation eqn. (2.82) with the assumption of incompressibility. With the definition of fractional flow coefficients and the mean mobility as

$$f_{\mathrm{n}} = \frac{\lambda_{\mathrm{n}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{n}}} \qquad f_{\mathrm{w}} = \frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{n}}} \tag{2.88}$$

and

$$\bar{\lambda} = \frac{\lambda_{\mathrm{w}}\lambda_{\mathrm{n}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{n}}} \tag{2.89}$$

the resulting equation is

$$\nabla \cdot [f_{\mathrm{n}}\boldsymbol{u}_{\mathrm{T}} - \bar{\lambda}(\nabla P_c + \Delta\gamma\nabla z)] = \phi\frac{\partial S_{\mathrm{w}}}{\partial t} - q_{\mathrm{n}} \tag{2.90}$$

Various terms in this equation can be expressed in terms of saturation:

$$\nabla P_c = \frac{\mathrm{d}P_c}{\mathrm{d}S_\mathrm{w}} \nabla S_\mathrm{w}$$

$$\nabla . (f_\mathrm{n} \boldsymbol{u}_\mathrm{T}) = \boldsymbol{u}_\mathrm{T} . \nabla f_\mathrm{n} + f_\mathrm{n} \nabla . \boldsymbol{u}_\mathrm{T} = \boldsymbol{u}_\mathrm{T} . \frac{\mathrm{d}f_\mathrm{n}}{\mathrm{d}S_\mathrm{w}} \nabla S_\mathrm{w} - f_\mathrm{n} q_\mathrm{T}$$

$$\nabla . (\bar{\lambda} \Delta\gamma \nabla z) = \Delta\gamma \nabla z . \nabla \bar{\lambda} = \Delta\gamma \nabla z . \frac{\mathrm{d}\bar{\lambda}}{\mathrm{d}S_\mathrm{w}} \nabla S_\mathrm{w}$$

In the last equation, it was assumed that ∇z is not a function of position. This will be satisfied for most co-ordinate systems. From eqn. (2.88) follows

$$\frac{\mathrm{d}f_\mathrm{n}}{\mathrm{d}S_\mathrm{w}} = -\frac{\mathrm{d}f_\mathrm{w}}{\mathrm{d}S_\mathrm{w}}$$

and since $f_\mathrm{w} + f_\mathrm{n} = 1$, we can write

$$-q_\mathrm{n} + f_\mathrm{n} q_\mathrm{T} = q_\mathrm{w} - f_\mathrm{w} q_\mathrm{T}$$

After substitution of the above expressions into eqn. (2.90), the final equation takes on the form

$$-\nabla . \left[\bar{\lambda} \frac{\mathrm{d}P_c}{\mathrm{d}S_\mathrm{w}} \nabla S_\mathrm{w} \right] - \left[\boldsymbol{u}_\mathrm{T} \frac{\mathrm{d}f_\mathrm{w}}{\mathrm{d}S_\mathrm{w}} + \Delta\gamma \nabla z \frac{\mathrm{d}\bar{\lambda}}{\mathrm{d}S_\mathrm{w}} \right] . \nabla S_\mathrm{w} = \phi \frac{\partial S_\mathrm{w}}{\partial t} + q_\mathrm{w} - f_\mathrm{w} q_\mathrm{T} \tag{2.91}$$

Equation (2.91) is the general formulation that includes the equations derived by Fayers and Sheldon (1959) and Hiatt (1968) as special cases. In order to solve this equation, it is necessary to first solve eqn. (2.84) for $\boldsymbol{u}_\mathrm{T}$, which is trivial only for the one-dimensional case (Exercise 2.2 at the end of this chapter). Equation (2.91) is generally of the parabolic type because $\mathrm{d}P_c/\mathrm{d}S_\mathrm{w} < 0$ and changes to the hyperbolic type if $P_c = 0$. In the latter case it reduces to

$$-\left[\boldsymbol{u}_\mathrm{T} \frac{\mathrm{d}f_\mathrm{w}}{\mathrm{d}S_\mathrm{w}} + \Delta\gamma \nabla z \frac{\mathrm{d}\bar{\lambda}}{\mathrm{d}S_\mathrm{w}} \right] . \nabla S_\mathrm{w} = \phi \frac{\partial S_\mathrm{w}}{\partial t} + q_\mathrm{w} - f_\mathrm{w} q_\mathrm{T} \tag{2.92}$$

which is a hyperbolic equation, because $\mathrm{d}f_\mathrm{w}/\mathrm{d}S_\mathrm{w} > 0$. Finally, if the fluids have equal density (or $\nabla z = 0$), the fractional flow equation simplifies to

$$\boldsymbol{u}_\mathrm{n} = f_\mathrm{n} \boldsymbol{u}_\mathrm{T} \qquad \boldsymbol{u}_\mathrm{w} = f_\mathrm{w} \boldsymbol{u}_\mathrm{T}$$

and eqn. (2.92) for this case may be written in the familiar form of the waterflood equation:

$$-\boldsymbol{u}_{\mathrm{T}}\frac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}}\cdot\nabla S_{\mathrm{w}}=\phi\frac{\partial S_{\mathrm{w}}}{\partial t}+q_{\mathrm{w}}-f_{\mathrm{w}}q_{\mathrm{T}} \tag{2.93}$$

Equations (2.84) and (2.93) are equivalent to the system of eqns. (2.79) and (2.80). The source term $q_{\mathrm{w}}-f_{\mathrm{w}}q_{\mathrm{T}}$ will be zero for production, since for this case $q_{\mathrm{w}}=f_{\mathrm{w}}q_{\mathrm{T}}$ by Darcy's law. However, for injection, the source term may be non-zero. For example, when the wetting phase is injected, $q_{\mathrm{T}}=q_{\mathrm{w}}$, and

$$q_{\mathrm{w}}-f_{\mathrm{w}}q_{\mathrm{T}}=(1-f_{\mathrm{w}})q_{\mathrm{w}}\neq 0$$

Further discussion of source terms is contained in Chapters 5 and 7.

The derivation for compressible flow follows the same lines, but the resulting equations are considerably more complex. Written in terms of p_{n} and S_{w}, they are

$$\left(\frac{b_{\mathrm{n}}}{b_{\mathrm{w}}}\right)\nabla\cdot[\lambda_{\mathrm{w}}(\nabla p_{\mathrm{n}}-\nabla P_c-\gamma_{\mathrm{w}}\nabla z)]+\nabla\cdot[\lambda_{\mathrm{n}}(\nabla p_{\mathrm{n}}-\gamma_{\mathrm{n}}\nabla z)] \tag{2.94}$$

$$=\phi\left[\frac{b_{\mathrm{n}}}{b_{\mathrm{w}}}S_{\mathrm{w}}\frac{\partial b_{\mathrm{w}}}{\partial t}+(1-S_{\mathrm{w}})\frac{\partial b_{\mathrm{n}}}{\partial t}\right]+\frac{b_{\mathrm{n}}}{b_{\mathrm{w}}}q_{\mathrm{w}}+q_{\mathrm{n}}$$

$$b_{\mathrm{w}}\left(\boldsymbol{u}_{\mathrm{T}}\frac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}}+k\Delta\gamma\frac{\mathrm{d}\psi}{\mathrm{d}S_{\mathrm{w}}}\nabla z\right)\cdot\nabla S_{\mathrm{w}}+\nabla\cdot\left[b_{\mathrm{w}}k\psi\frac{\mathrm{d}P_c}{\mathrm{d}S_{\mathrm{w}}}\nabla S_{\mathrm{w}}\right] \tag{2.95}$$

$$=-\phi\frac{\partial}{\partial t}(b_{\mathrm{w}}S_{\mathrm{w}})-q_{\mathrm{w}}-f_{\mathrm{w}}\nabla\cdot(b_{\mathrm{w}}\boldsymbol{u}_{\mathrm{T}})-\psi\nabla\cdot[b_{\mathrm{w}}k\,\Delta\gamma\nabla z]$$

where

$$b_l=1/B_l$$

$$f_{\mathrm{w}}=\frac{k_{\mathrm{rw}}/\mu_{\mathrm{w}}}{k_{\mathrm{rn}}/\mu_{\mathrm{n}}+k_{\mathrm{rw}}/\mu_{\mathrm{w}}},\qquad \psi=\frac{k_{\mathrm{rn}}}{\mu_{\mathrm{n}}}f_{\mathrm{w}}$$

and λ_l is now defined by

$$\lambda_l=\frac{kk_{\mathrm{r}l}}{\mu_l B_l}$$

In Exercises 2.2 and 2.3 we outline the derivation of two-phase flow equations given above and the corresponding equations for three-phase flow.

2.6 FLOW EQUATIONS WHICH INCLUDE NON-DARCY EFFECTS

Strictly speaking, Darcy's Law is only valid for Newtonian fluids over a limited range of flow rates where turbulence, inertial and other high-velocity effects are negligible. Furthermore, at very low pressures this law does not hold due to the slip phenomenon. In this section we provide some of the relations used in practice when the classical form of Darcy's Law does not work.

2.6.1 High Flow Rates (Inertial and Turbulent Effects)

As the flow velocity is increased, deviations from Darcy's Law are observed. Investigators have variously attributed this to turbulent flow (Fancher and Lewis, 1933; Elenbaas and Katz, 1947; Cornell and Katz, 1953) or inertial effects (Hubbert, 1956; Houpeurt, 1959). The generally accepted explanation (Wright, 1968) is that, as the velocity is increased, deviation is due to inertial effects initially, followed later by turbulent effects. Hubbert (1956) noted deviation from Darcy's Law at a Reynolds' number of flow of about 1 (based on the grain diameter of unconsolidated media), whereas turbulence was not observed until the Reynolds' number approached 600. The transition from purely laminar flow to fully turbulent flow is a long one. This range of flow rates is adequately represented by a quadratic equation (Forschheimer, 1901) given for one-dimensional steady-state flow without significant gravitational effects by

$$-\frac{\mathrm{d}p}{\mathrm{d}x} = \frac{\mu}{k}u + \beta\rho|u|u \tag{2.96}$$

where β is the 'turbulence' factor (Katz *et al*., 1959).

For multidimensional flow the equation can be written as (Geertsma, 1974):

$$-\nabla p = \frac{\mu}{k}\boldsymbol{u} + \beta\rho|u|\boldsymbol{u}$$

Equation (2.96) which incorporates laminar, inertial and turbulent (LIT) effects is a general momentum balance equation. It may be rearranged to the form

$$u = -\delta\frac{k}{\mu}\frac{\mathrm{d}p}{\mathrm{d}x} \tag{2.97}$$

where

$$\delta = 1 \Big/ \left(1 + \frac{\beta \rho k}{\mu} |u|\right)$$

is a 'turbulence' correction factor (Wattenbarger and Ramey, 1968; Govier, 1961). When δ = 1•0, the above equation is equivalent to Darcy's Law. In an anisotropic medium, δ is different in different directions. Flow through such a medium is then given, in generalised form, by

$$\boldsymbol{u} = -\frac{1}{\mu} k \delta \nabla p \tag{2.98}$$

where in general k and δ are tensors.

It is seen that eqn. (2.98) will represent both laminar flow and flow where inertial–turbulent (IT) effects are present. It has been referred to as the generalised laminar–inertial–turbulent (LIT) equation in the ERCB (1975) manual. IT effects are usually only important with gas flow near the well bore.

The gas-flow equation obtained by combining eqn. (2.98) with the conservation of mass is

$$\nabla . \left[\frac{p}{Z} \frac{k}{\mu} \delta \nabla p\right] = \frac{\partial}{\partial t}\left(\frac{\phi p}{Z}\right) + \tilde{q} \frac{\mathrm{R}T}{\mathrm{M}}$$

Equations of this type must be solved iteratively.

2.6.2 Threshold and Slip Phenomena

It has been experimentally observed that a certain non-zero pressure gradient is necessary to initiate flow. The relationship between q and $\partial p/\partial x$ for low rates is shown in Fig. 2.2. The slip (or Klinkenberg) phenomenon occurs in gas flow at low pressures and results in an increase of effective permeability compared to that measured for liquids. Although both phenomena are relatively unimportant, Darcy's Law can be easily modified to account for them. For a detailed discussion of these effects, see works by Collins (1961) and Bear (1972).

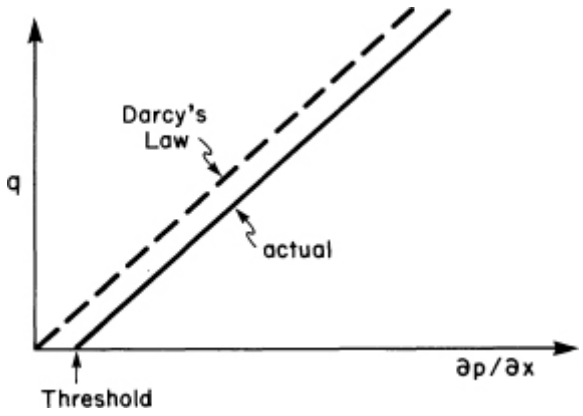


FIG. 2.2. Threshold phenomenon.

2.6.3 Non-Newtonian Flow

Some fluids (e.g. polymer solutions) exhibit non-Newtonian behaviour, characterised by nonlinear dependence of shear stress on shear rate. The theory of such behaviour, which is beyond the scope of this book, is discussed in the literature on rheology. For practical purposes, the resistance to flow in porous media can be described by Darcy's Law which includes apparent viscosity μ_{app} dependent on flow velocity. An example of function μ for polymer solution is in Fig. 2.3. The Darcy's velocity $\boldsymbol{u}$ can thus be written as

$$\boldsymbol{u} = -\frac{k}{\mu(\boldsymbol{u})}(\nabla p - \gamma \nabla z) \tag{2.99}$$

The pseudoplastic region of flow can be approximated over a wide range of velocities with the power-law model (Blake–Kozeny equation, see Bird *et al.*, 1960):

$$\mu_{app} = \mathrm{H}u^{\mathrm{n}-1} \tag{2.100}$$

The constants H and n must be determined empirically.

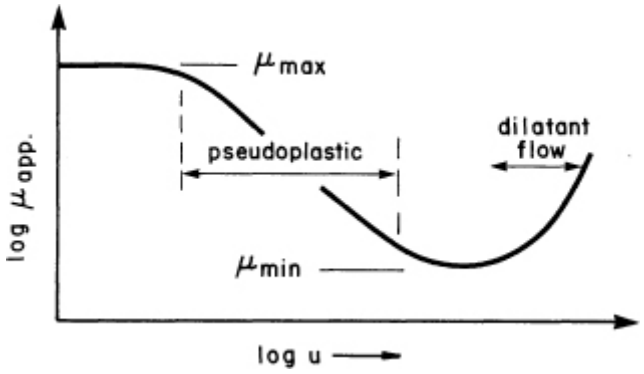


FIG. 2.3. Apparent viscosity for non-Newtonian fluids (after Bondor *et al*., 1972).

2.6.4 Other Effects

There are several other effects that cause additional nonlinearities in the basic flow equations. They are usually associated with special secondary and tertiary recovery techniques. For example, polymer in the solution is adsorbed on reservoir rock and solution changes into water. Also, contact with polymer reduces relative permeability to subsequent water flow.

Properties dependent on concentration must be considered when immiscible equations are applied to miscible systems, $C0_2$ and micellar floods, etc. In thermal recovery techniques, all coefficients in Darcy's Law become functions of temperature. As a last example, Finol and Farouq Ali (1975) also considered compaction of reservoir rock under changing pressure (ground subsidence).

2.7 FLUID AND ROCK PROPERTIES

The character of the equations and the kind of methods that must be employed to simulate them depend to a large degree on fluid and rock properties. A brief discussion of these properties is presented in this section, so that their rôle in reservoir simulation can be fully appreciated. Extensive treatment of physical properties and collections of correlations are found in Frick and Taylor (1962) and Katz *et al*. (1959).

2.7.1 Fluid Properties

For fluids that can be approximated by the isothermal β-model, formation volume factors and viscosities are functions of pressure only and should be determined at reservoir temperature. Note that B_g is related to gas compressibility Z. Because compressibility of water c_w is small, it can be expressed by the eqn. (2.40).

$$B_w = \frac{B_{wb}}{[1 + c_w(p_w - p_{wb})]} \tag{2.101}$$

where B_{wb} and p_{wb} are the conditions at some reference point (usually bubble point).

Viscosities of oil and gas are generally strong functions of temperature and this must be taken into account if temperature changes cannot be ignored as in the case of the flow in the wellbore or in thermal-recovery processes. Temperature dependence at a given pressure can usually be assumed to be linear in logarithmic coordinates, i.e.,

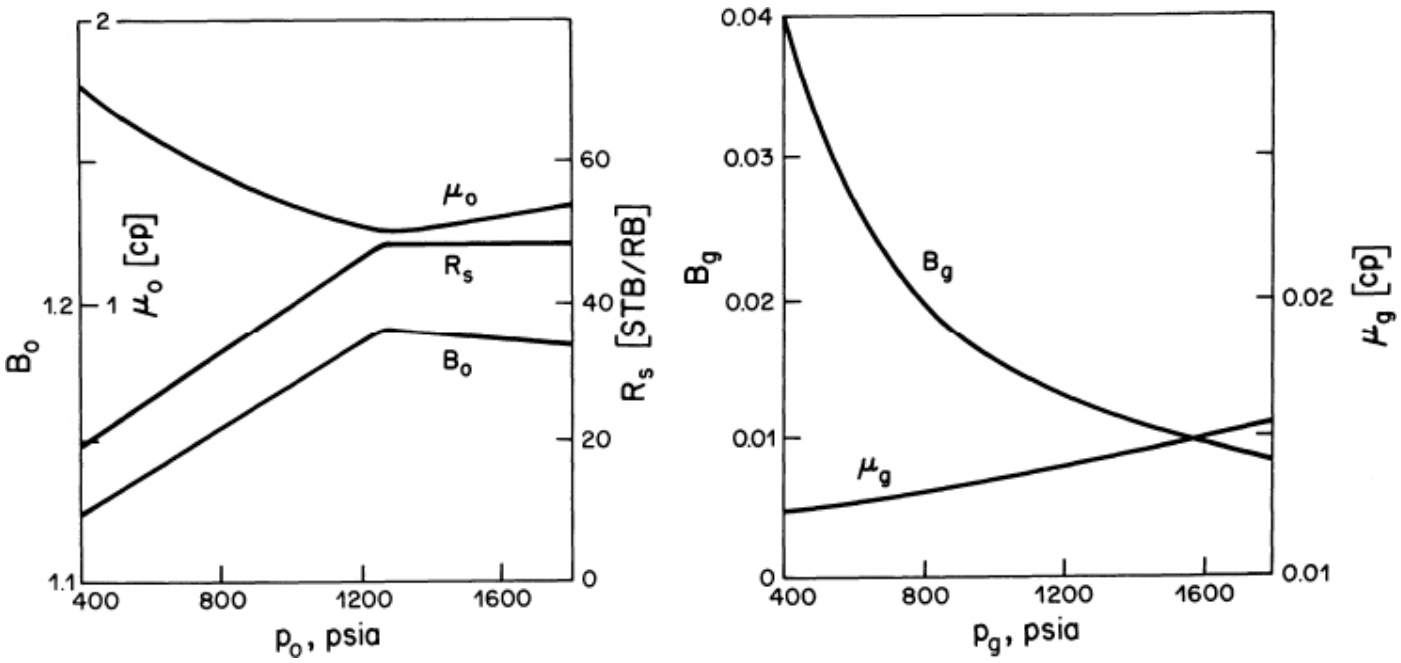


FIG. 2.4. Typical pressure-dependent functions (from Settari and Aziz, 1975).

$$\mu = \mu^\circ \left(\frac{T}{T^\circ} \right)^C \tag{2.102}$$

where T° and μ° are the values at a reference point and the constant C may be determined from the value of μ at one $T \neq T^\circ$. Obviously, this approximation will not be accurate for oil, if the bubble point is crossed within the temperature range considered. When the β-model is not adequate, more involved data characterising compositional properties of fluids must be specified (Chapter 12). An example of pressure-dependent properties for oil and gas is shown in Fig. 2.4.

2.7.2 Rock Properties

2.7.2.1 *Capillary Pressure*

In the first approximation, capillary pressure and relative permeabilities may be considered as functions of the reservoir rock only. In a two-phase situation, a typical capillary pressure curve is represented by Fig. 2.5. Capillarity depends on the saturation of wetting fluid and direction of saturation change (drainage or imbibition curve). The value P_{cb} which is necessary to start displacement is called 'threshold pressure' (Bear, 1972) and is important for low permeability rocks (Thomas *et al.*, 1968). The value of saturation at which the wetting phase no longer can be displaced by applying a pressure gradient is called the 'irreducible saturation'. Theoretically, the P_c curve should have an asymptote at that value in order that pressure gradient remains

continuous in both phases. This can be seen by considering vertical gravity equilibrium. A similar situation occurs at the other end of the curve during the imbibition cycle, when the value of the irreducible saturation of non-wetting phase is approached. Welge (1949) was first to measure negative capillary pressure which initiated discussion as well as further experimental work (for example, Calhoun *et al*., 1949; Morrow and Harris, 1965; Morrow, 1970). Later workers concluded that wettability and capillarity are functions of several additional variables. Among them are properties of reservoir fluids, contamination, and temperature (Poston *et al*., 1970).

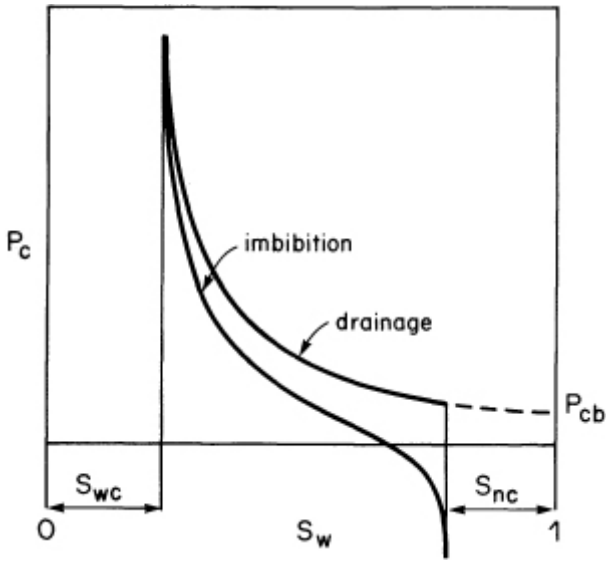


FIG. 2.5. Typical P_c curve.

For numerical simulation, the slopes must be always finite. Some of the early workers attempted to incorporate large P_c curve slopes into computer models to approximate the asymptotic character. This created computational problems and it was later recognised that such a procedure was not necessary. Use of reasonable values for slopes in the region of S_{wc} makes the problem easier to solve. On the other hand, it creates the problem of 'overshoot' which must be handled numerically (Chapter 12).

For three-phase flow, Leverett and Lewis (1941) first investigated the functions (2.58) and (2.59) and found some justification for the following assumptions:

$$P_{cow} = \mathrm{f}(S_w) \tag{2.103a}$$

$$P_{cog} = \mathrm{f}(S_g) \tag{2.103b}$$

Assumptions (2.103a and b) are still generally used, although some refinements have been proposed, as by Shutler (1969).

Although it is possible to formulate a model that accounts for hysteresis resulting from the change of direction of flow, in most situations the direction of flow can be predicted and only one set of capillary pressure curves is required.

2.7.2.2 *Relative Permeability*

Most of the experimental work on relative permeabilities has also been done on two-phase systems. Figure 2.6 shows typical results that might be obtained for an oil–water system with water displacing oil. The value of S_w at which water starts to flow is called the critical saturation, S_{wc}. The saturation S_{nc} at which the displaced phase ceases to flow is called the residual saturation. Similarly, in a drainage cycle S_{nc} will be the critical saturation and S_{wc} the residual saturation. Because the slopes of capillary pressure curves at irreducible saturations must be finite in numerical models, the capillary pressure curve itself cannot be used to define the saturation at which the displaced phase becomes immobile. This value is therefore determined by the residual saturation, at which the relative permeability becomes zero. In terms of Darcy's flow equation it means that the phase ceases to flow *because the mobility is zero and not because the external force is zero*. Consequently, there is no need to distinguish between critical and irreducible saturations.

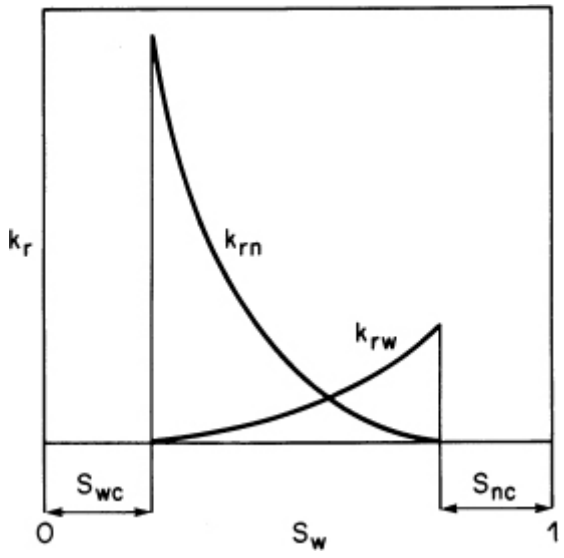


FIG. 2.6. Typical k_r curves.

Along with numerous experimental results, there are also theoretical methods which show that porosity, capillary pressure, and relative permeability are related. An approximate relationship between them is represented by the Leverett *J*-function [$J = (P_c/\sigma)\ \sqrt{(k/\phi)}$] which is the basis for some theoretical methods of calculating

permeability (Ashford, 1969). Therefore, in a reservoir with strongly varying properties, different relative permeability curves and residual saturations should be used in different parts of the reservoir.

Wettability of the rock has also strong influence on relative permeabilities (Owens and Archer, 1971). Mungan (1972) shows that reservoir fluids rather than refined fluids should be used for relative permeability measurements.

Starting with the first measurements by Leverett and Lewis (1941), almost all three-phase work has been done experimentally (Corey *et al*., 1956; Snell, 1962). These investigations show that functional dependence can be approximated by

$$k_{rw} = f(S_w) \qquad k_{rg} = f(S_g) \tag{2.104}$$

$$k_{ro} = f(S_w, S_g) \tag{2.105}$$

The function (2.105) is rarely known and even when it is, the form is not convenient for use in numerical models.

Practical approaches are based on the estimation of three-phase relative permeability from two sets of two-phase data: relative permeability in an oil-water system

$$k_{row} = f(S_w) \tag{2.106}$$

and in an oil−gas system

$$k_{rog} = f(S_g) \tag{2.107}$$

The concept can be better understood if it is realised that as far as the k_{row} is concerned, the non-wetting phase can be thought of as the sum of the oil and gas phases, and similarly for k_{rog} the wetting phase is all of the liquid (oil and water) present. Therefore, in Fig. 2.7a the point with $k_{row} = 0$ denotes the maximum water saturation rather than the critical oil saturation, because oil saturation could be further decreased by increasing gas saturation. However, it has been found experimentally that a non-zero residual oil saturation S_{om} exists when oil is displaced simultaneously by water and gas. The remark on P_c hysteresis made earlier also applies to k_r data and the correct data according to displacement (drainage or inhibition) must be selected. Numerical treatment of hysteresis is discussed in Chapter 12.

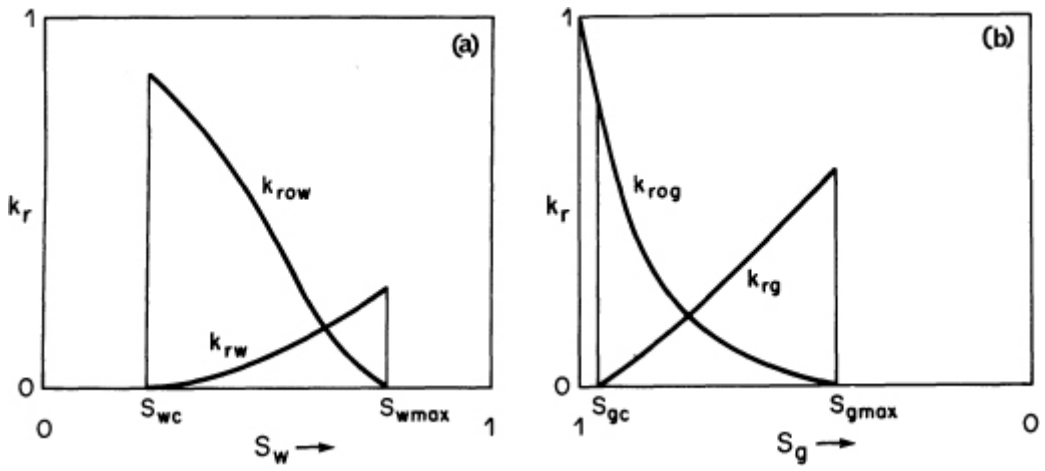


FIG. 2.7.(a) Oil–water system; (b) gas–oil system.

The simplest approach to predicting k_{ro} is to define

$$k_{ro} = k_{row} k_{rog} \tag{2.108}$$

Two more accurate models were proposed by Stone. In the first method, Stone (1970) defines normalised saturations as

$$S_0^* = \frac{S_o - S_{om}}{(1 - S_{wc} - S_{om})} \qquad S_o \geq S_{om} \tag{2.109}$$

$$S_w^* = \frac{S_w - S_{wc}}{(1 - S_{wc} - S_{om})} \qquad S_w \geq S_{wc} \tag{2.110}$$

$$S_g^* = \frac{S_g}{(1 - S_{wc} - S_{om})} \tag{2.111}$$

Now the oil relative permeability in a three-phase system is assumed to be

$$k_{ro} = S_o^* \beta_w \beta_g \tag{2.112}$$

The multipliers β_w and β_g are determined from the condition that eqn. (2.112) reduces to two-phase data for the two extreme cases of $S_g = S_g^* = 0$ and $S_w = S_{wc}$. This gives

$$\beta_w = \frac{k_{row}(S_w)}{1 - S_w^*} \qquad \beta_g = \frac{k_{rog}(S_g)}{1 - S_g^*} \tag{2.113}$$

The region of mobile oil phase for the model (2.113) is shown on the ternary diagram in Fig. 2.8, assuming increasing S_w and S_g.

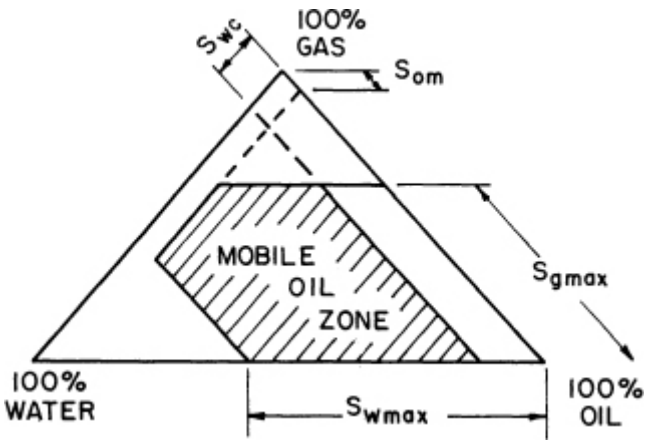


FIG. 2.8. Zone of mobile oil for three-phase flow.

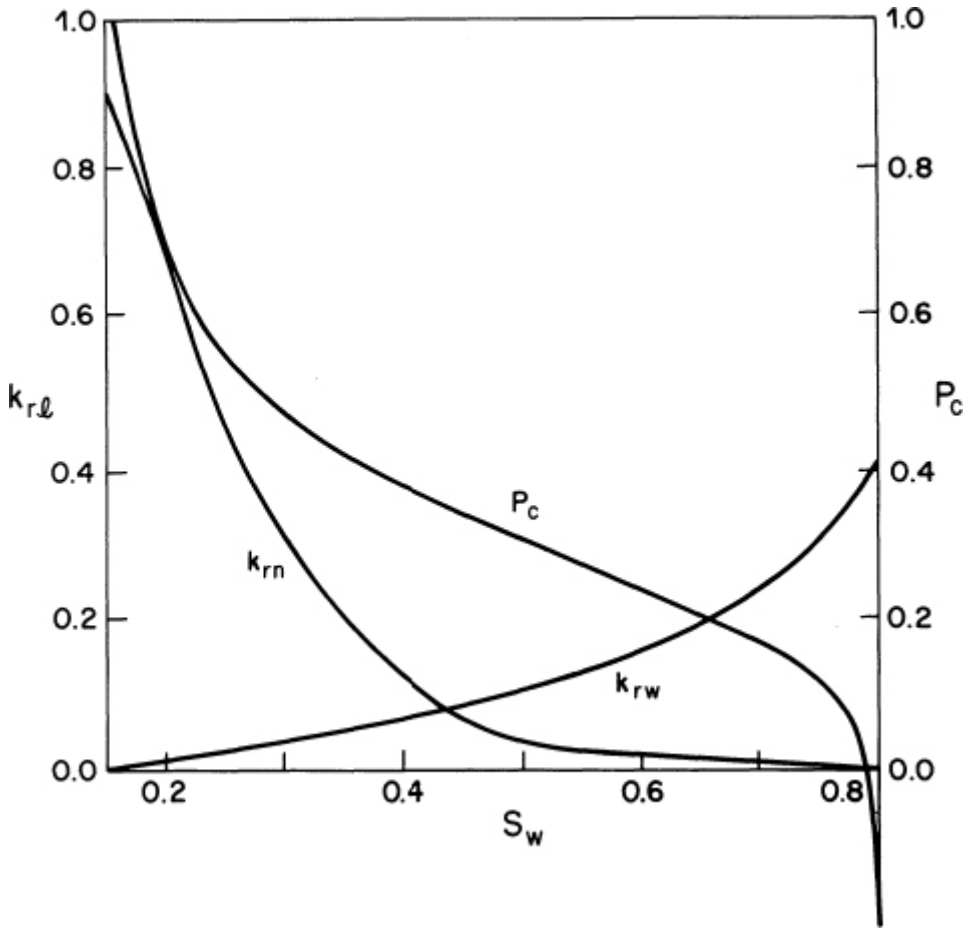


FIG. 2.9. Typical saturation-dependent functions for water-oil systems with water as the wetting phase (from Settari and Aziz, 1975).

The second model (Stone, 1973) does not require specification of S_{om} and it is, in fact, capable of predicting it. The prediction equation is derived from channel-flow considerations and has the form

$$k_{ro} = (k_{row} + k_{rw})(k_{rog} + k_{rg}) - (k_{rw} + k_{rg}) \tag{2.114}$$

with the restriction that $k_{ro} \geq 0$, i.e., negative values of k_{ro} imply immobile oil.

It is worth noting that both models will reduce exactly to two-phase data *only* if the relative permeability at the end points is equal to one:

$$k_{row}(S_{wc}) = k_{rog}(S_g = 0) = 1 \tag{2.115}$$

Otherwise, the function $k_{ro}(S_w, S_g)$ will only approximate the two-phase data. A model which does not have this limitation can be developed if we assume that the gas-oil data are measured *in the presence of irreducible water*. In this case, an oil-water system at S_{wc} and an oil-gas system at $S_g = 0$ are physically identical (i.e., both have $S_w = S_{wc}$, $S_o = 1 - S_{wc}$) and therefore eqn. (2.115) would be equivalent to defining the absolute permeability as the effective permeability of the oil phase in presence of S_{wc}. In order to preserve the customary definition of permeability, the models presented by Stone must be altered.

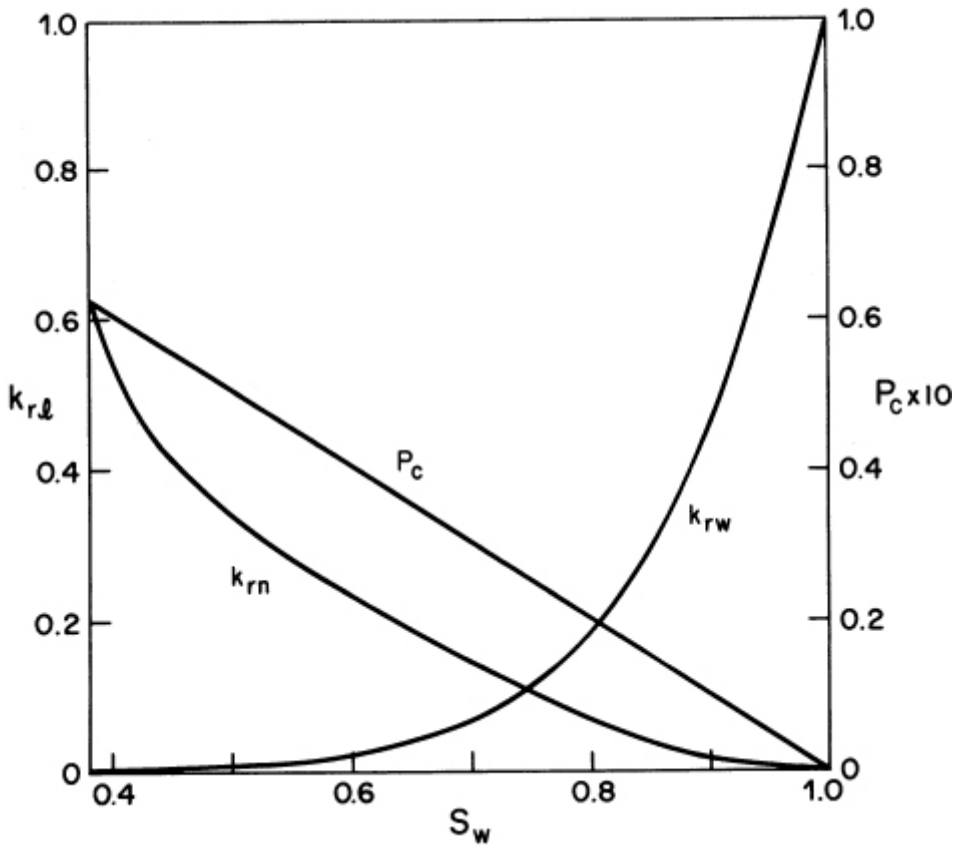


FIG. 2.10. Typical saturation-dependent functions for gas-oil systems with oil as the wetting phase (from Settari and Aziz, 1975).

Let us denote

$$k_{row}(S_{wc}) = k_{rog}(S_L = 1) = k_{rocw} \tag{2.116}$$

where $S_L = 1 - S_g = S_o + S_{wc}$ for the oil–gas system. The two models of Stone can now be modified as follows:

Model I

$$k_{ro} = k_{rocw} S_o^* \beta_w \beta_g \tag{2.117}$$

where

$$\beta_w = \frac{k_{row}(S_w)/k_{rocw}}{1 - S_w^*} \qquad \beta_g = \frac{k_{rog}(S_L)/k_{rocw}}{1 - S_g^*} \tag{2.118}$$

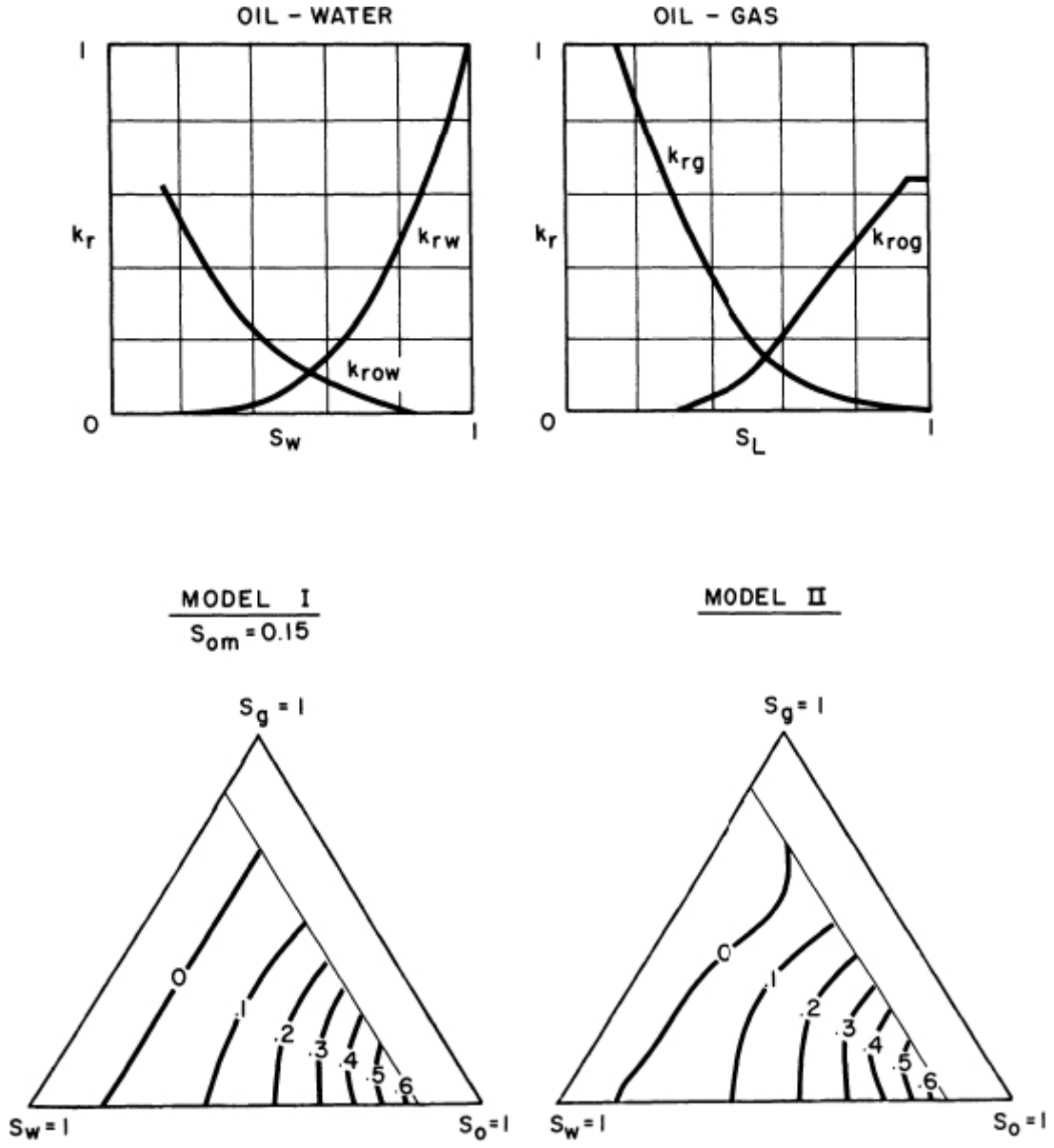


FIG. 2.11. Three-phase k_{ro} calculated by Stone's Model I and II.

Model II

$$k_{ro} = k_{rocw}[(k_{row}/k_{rocw} + k_{rw})(k_{rog}/k_{rocw} + k_{rg}) - (k_{rw} + k_{rg})] \tag{2.119}$$

Dietrich and Bondor (1976) have suggested another normalised form of Model II:

$$k_{ro} = \frac{1}{k_{rocw}}(k_{row} + k_{rw})(k_{rog} + k_{rg}) - (k_{rw} + k_{rg}) \tag{2.120}$$

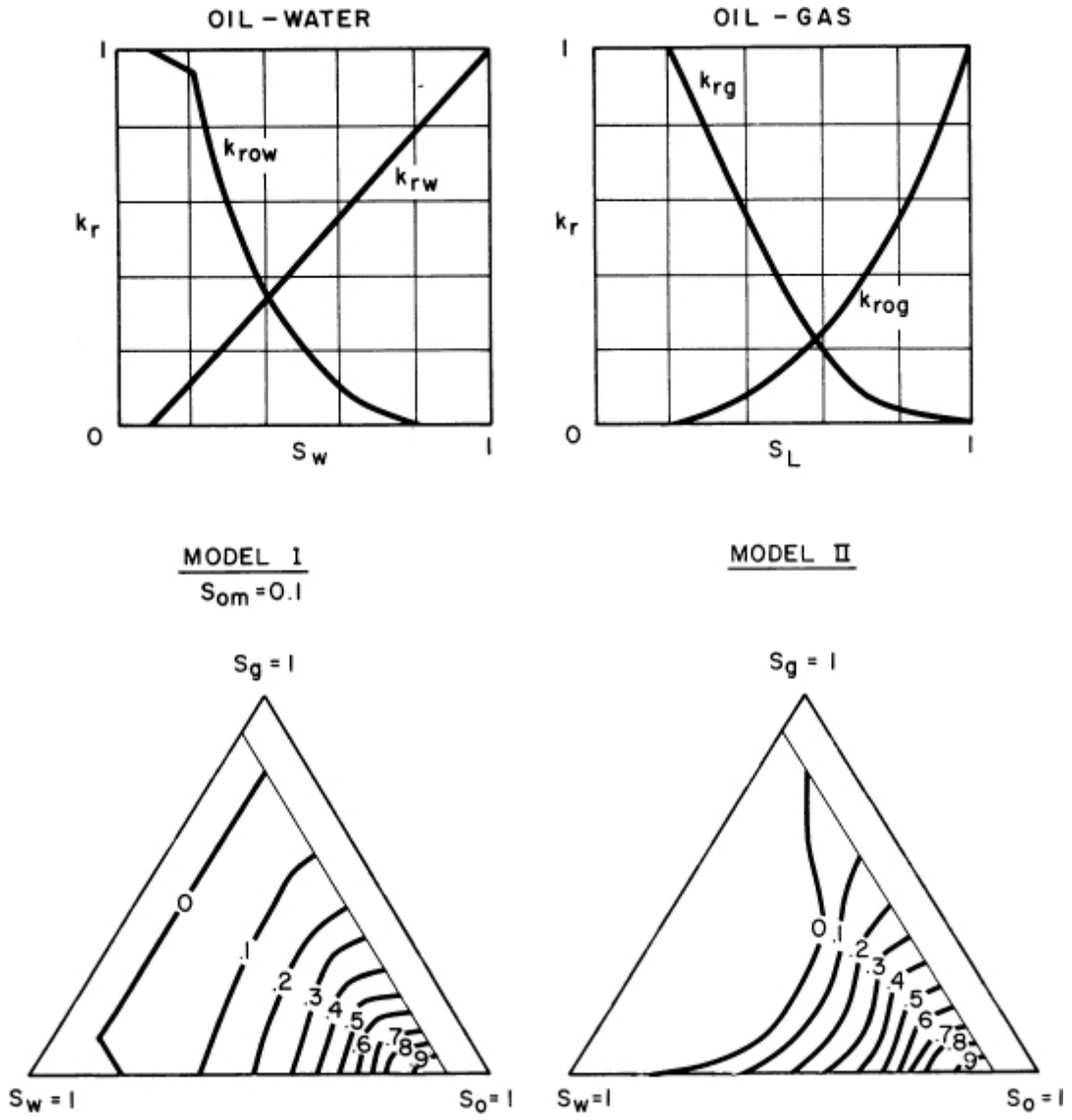


FIG. 2.12. Three-phase k_{ro} calculated by Stone's Model I and II.

This model tends to predict higher S_{om} compared to eqn. (2.119). Also, it gives incorrect values for small k_{rocw} since

$$\lim k_{ro} = \infty \qquad \text{as} \qquad k_{rocw} \to 0.$$

Therefore eqn. (2.119) is a more reasonable model.

Examples

Figures 2.9 and 2.10 give examples of two-phase rock properties. Two examples of three-phase k_{ro} calculated by eqns. (2.118) and (2.119) are in Figs. 2.11 and 2.12. The plots of two-phase data are linear interpolations between tabulated data, as it would be

typically entered in a simulator. The second set of data shows that the two models can predict quite different permeabilities in the region of low k_{ro}.

2.7.2.3 *Other Properties*

The tensorial character of absolute permeability has already been discussed in Section 2.2.3. Because of natural stratification of reservoir rocks, the main axes of the tensor are usually horizontal and vertical and can be identified with horizontal (k_H) and vertical *(k_V)* permeability. Therefore for a Cartesian co-ordinate system *xyz* with *z* vertical, we will have $k_x = k_y = k_H$, $k_z = k_V$.

Porosity of the rock is dependent on pressure due to rock compressibility c_R. Usually, c_R is comparable to water compressibility and likewise can be assumed constant, which gives us

$$\phi = \phi^\circ[1 + c_R(p - p^\circ)] \tag{2.121}$$

where ϕ° is porosity at the reference pressure p°.

2.8 CONCLUDING REMARKS

In this chapter the basic fluid flow equations for black-oil models have been developed. Limitations of these equations have not been fully discussed since no alternative is available anyway, and the discussion of this topic must be restricted in a book that deals mainly with numerical methods.

The greatest weakness of the mathematical model is in the area of three-phase relative permeability and capillary pressure. Simple models based on two-phase data require verification.

We have also ignored problems of fluid flow with temperature change and mass diffusion. These topics are beyond the scope of this text.

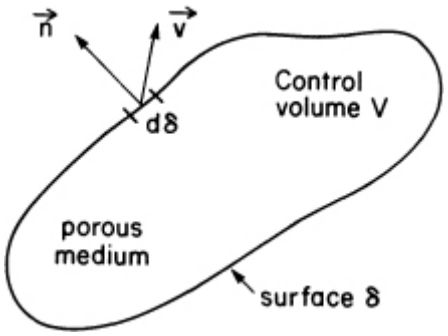
EXERCISES

Exercise 2.1

Derive the equation of mass conservation for an arbitrary substance by considering a porous medium of arbitrary shape saturated with one or more phases.

Solution Outline

Consider flow of a certain substance Q which is distributed with concentration C in the system shown below.



Show that:

$$\text{the net flow across the surface} = \int_0 \boldsymbol{v} \,.\, \boldsymbol{n}\, \mathrm{d}\sigma \tag{A}$$

$$\text{the generation of } Q \text{ within } V = \int_V \tilde{q}\, \mathrm{d}V \tag{B}$$

$$\text{the rate of changes of } Q \text{ in } V = \int_V C\, \mathrm{d}V \tag{C}$$

where

$\boldsymbol{v}$ = actual velocity of the substance (interstitial velocity) in the porous medium

$\tilde{q}$ = generation of Q per unit volume per unit time.

Now using Gauss' theorem show that the conservation equation for Q

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V C\, \mathrm{d}V = -\int_\sigma c\boldsymbol{v} \,.\, \boldsymbol{n}\, \mathrm{d}\sigma + \int_V \tilde{q}\, \mathrm{d}V$$

can be written as

$$\int_V \left(\frac{\partial C}{\partial t} + \nabla \,.\, C\boldsymbol{v} - \tilde{q} \right) \mathrm{d}V = 0 \tag{D}$$

Next show that for a porous medium saturated with a single-component fluid eqn. (D) can be written as

$$-\nabla . (\rho \boldsymbol{u}) = \frac{\partial}{\partial t}(\rho \phi) - \tilde{q} \tag{E}$$

where

$$\boldsymbol{u} = \boldsymbol{v}\phi = \text{Darcy velocity}$$

Hint: Mass flux = $\int C\boldsymbol{v}.\mathrm{n} = \int \rho \boldsymbol{u}.\boldsymbol{n}$ since $C = \rho$ for the pore space and $C = 0$ for the rock matrix. Similarly $C = \rho\phi$ for the accumulation term.

Now extend eqn. (E) to multiphase flow and provide definitions of Darcy velocity and relative permeabilities based on the mass flux.

Exercise 2.2

Derive eqns. (2.94) *and* (2.95).

Solution Outline

(a) Starting with eqns. (2.20) and (2.21), eliminate $\partial S_l/\partial t$ terms by multiplying the wetting-phase equation by b_n/b_w to get eqn. (2.94).

(b) Darcy velocity $\mathbf{u}_w$ is given by

$$\boldsymbol{u}_w = f_w \boldsymbol{u}_T + k\psi(\nabla P_c + \Delta\gamma \nabla z) \tag{A}$$

Substitute this in the wetting-phase equation to obtain eqn. (2.95).

Exercise 2.3

Derive the equivalent of eqns. (2.94) *and* (2.95) *for three-phase flow*.

Solution Outline

Start with

$$-\nabla . (b_w \boldsymbol{u}_w) = \frac{\partial}{\partial t}(\phi b_w S_w) + q_w \tag{A}$$

$$-\nabla . (b_o \boldsymbol{u}_o) = \frac{\partial}{\partial t}(\phi b_o S_o) + q_o \tag{B}$$

$$-\nabla . (b_g \boldsymbol{u}_g) - \nabla R_s . (b_o \boldsymbol{u}_o) = \frac{\partial}{\partial t}(\phi b_g S_g) + \phi b_o S_o \frac{\partial R_s}{\partial t} + q_g \tag{C}$$

(See Exercise 5.2 of Chapter 5 for the form of eqn. (C).

(a) Eliminate saturation derivatives by multiplying the water and gas equations by b_o/b_w and b_o/b_g and adding:

$$\begin{aligned}
&\nabla . [\lambda_o(\nabla p_o - \gamma_o \nabla z)] + (b_o/b_w)\nabla . [\lambda_w(\nabla p_o - \nabla P_{cow} - \gamma_w \nabla z)] \\
&+ (b_o/b_g)\nabla . [\lambda_g(\nabla p_o + \nabla P_{cog} - \gamma_g \nabla z)] + (b_o/b_g)\nabla R_s . [\lambda_o(\nabla p_o - \gamma_o \nabla z)] \\
&= S_o\left[\frac{\partial(\phi b_o)}{\partial t} + \left(\frac{b_o^2}{b_g}\right)\phi\frac{\partial R_s}{\partial t}\right] + (b_o/b_w)S_w\frac{\partial(\phi b_w)}{\partial t} \\
&+ \left(\frac{b_o}{b_g}\right)S_g\frac{\partial(\phi b_g)}{\partial t} + q_o + (b_o/b_w)q_w + (b_o/b_g)q_g
\end{aligned} \tag{D}$$

(b) The fractional flow equation for water is:

$$\boldsymbol{u}_w = f_w \boldsymbol{u}_T + k\psi_{og}(\nabla P_{cow} + \Delta\gamma_{ow}\nabla z) + k\psi_g(\nabla P_{cog} + \Delta\gamma_{og}\nabla z) \tag{E}$$

where

$$\Delta\gamma_{ow} = \gamma_w - \gamma_o \qquad \Delta\gamma_{og} = \gamma_o - \gamma_g$$

$$\psi_{og} = \left(\frac{k_{ro}}{\mu_o} + \frac{k_{rg}}{\mu_g}\right)f_w \qquad \psi_g = \frac{k_{rg}}{\mu_g}f_w$$

Substitute (E) in (A) to get

$$\begin{aligned}
&b_w\left[\boldsymbol{u}_T\frac{\partial f_w}{\partial S_w} + k\left(\Delta\gamma_{ow}\frac{\partial\psi_{og}}{\partial S_w} + \Delta\gamma_{og}\frac{\partial\psi_g}{\partial S_w}\right)\nabla z\right] . \nabla S_w \\
&+ b_w\left[\boldsymbol{u}_T\frac{\partial f_w}{\partial S_g} + k\left(\Delta\gamma_{ow}\frac{\partial\psi_{og}}{\partial S_g} + \Delta\gamma_{og}\frac{\partial\psi_g}{\partial S_g}\right)\nabla z\right] . \nabla S_g \\
&+ \nabla . \left[b_w k\psi_{og}\frac{dP_{cow}}{dS_w}\nabla S_w\right] + \nabla . \left[b_w k\psi_g\frac{dP_{cog}}{dS_g}\nabla S_g\right] \\
&= -\frac{\partial}{\partial t}(\phi b_w S_w) - q_w - f_w\nabla . (b_w\boldsymbol{u}_T) - \psi_{og}\nabla . (kb_w\Delta\gamma_{ow}\nabla z) \\
&- \psi_g\nabla . (kb_w\Delta\gamma_{og}\nabla z)
\end{aligned} \tag{F}$$

Similarly, derive the oil fractional flow equation

$$\boldsymbol{u}_o = f_o \boldsymbol{u}_T - k\xi_w(\nabla P_{cow} + \Delta\gamma_{ow}\nabla z) + k\xi_g(\nabla P_{cog} + \Delta\gamma_{og}\nabla z) \tag{G}$$

$$\xi_w = \frac{k_{rw}}{\mu_w} f_o \qquad \xi_g = \frac{k_{rg}}{\mu_g} f_o$$

and substitute (G) in (B) to get

$$\begin{aligned} & b_o\left[\boldsymbol{u}_T \frac{\partial f_o}{\partial S_w} + k\left(-\Delta\gamma_{ow}\frac{\partial \xi_w}{\partial S_w} + \Delta\gamma_{og}\frac{\partial \xi_g}{\partial S_w}\right)\nabla z\right].\nabla S_w \\ & + b_o\left[\boldsymbol{u}_T \frac{\partial f_o}{\partial S_g} + k\left(-\Delta\gamma_{ow}\frac{\partial \xi_w}{\partial S_g} + \Delta\gamma_{og}\frac{\partial \xi_g}{\partial S_g}\right)\nabla z\right].\nabla S_g \\ & - \nabla.\left[b_o k\xi_w \frac{dP_{cow}}{dS_w}\nabla S_w\right] + \nabla.\left[b_o k\xi_g \frac{dP_{cog}}{dS_g}\nabla S_g\right] \\ & = -\frac{\partial}{\partial t}(\phi b_o S_o) - q_o - f_o\nabla.(b_o\boldsymbol{u}_T) + \xi_w\nabla.(kb_o\Delta\gamma_{ow}\nabla z) \\ & - \xi_g\nabla.(kb_o\Delta\gamma_{og}\nabla z) \end{aligned} \tag{H}$$

---

† The terms ‘transmissibility’ (used earlier) and ‘mobility’ defined by eqn. (2.85) are different. ‘Transmissibility’ includes the formation volume factor term while ‘mobility’ does not.

CHAPTER 3

FLOW OF A SINGLE FLUID IN ONE DIMENSION

3.1 INTRODUCTION

This chapter deals with the numerical solutions of single-phase flow equations in one space dimension. Among the problems treated in this book, one-dimensional problems are the simplest and they do not always necessitate the use of numerical techniques. Analytical solutions are known for various special cases (cf. Muskat, 1937; Collins, 1961) and they provide a means of checking the accuracy of approximate methods. Furthermore, most of the concepts related to finite-difference methods and solution techniques can be introduced here in a simple, easy-to-understand form and later extended to several dimensions. (In this book, the term 'dimension' is used for reference only to dimensionality in space—not in time. Thus, a transient problem in x and t co-ordinates is one dimensional.)

The mathematical formulation of problems and their numerical treatment depends on the type of physical systems and assumptions involved in the development of a mathematical model of the system.

The flow equations to be considered in this chapter have been discussed in general three-dimensional form in Chapter 2. Here a summary of various one-dimensional forms of the flow equation is presented. Finite-difference methods for the solution of these equations will be presented in the following sections of this chapter. Most of the one-dimensional flow problems can be put into one of the following forms:

1. *Steady-state linear flow problems—linear ODE case*

$$\frac{\mathrm{d}^2 U}{\mathrm{d}x^2} = q(x) \tag{3.1}$$

2. *Transient linear flow problems—linear PDE case*

$$\frac{\partial^2 U}{\partial x^2} = \frac{\partial U}{\partial t} + q(x,t) \tag{3.2}$$

3. *Transient linear flow problems—variable coefficient and nonlinear PDE case*

$$\frac{\partial}{\partial x}\left[\lambda(x,U)\frac{\partial U}{\partial x}\right] = c(x,U)\frac{\partial U}{\partial t} + q(x,t) \tag{3.3}$$

4. *Transient radial flow problems—variable coefficient and nonlinear PDE case*

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\lambda(r,U)\frac{\partial U}{\partial r}\right] = c(r,U)\frac{\partial U}{\partial t} + q(r,t) \tag{3.4}$$

The use of the symbol U to represent the dependent variable p is convenient for a general discussion of the finite-difference methods. It will, however, be more convenient to use the symbol p for both the approximate and the exact solution in later chapters, and in some cases even in this chapter, when we must take advantage of the physical characteristics of the problem.

The next two sections introduce the terminology and the basic concepts of finite-difference methods. For the sake of clarity, this is carried out for a simple elliptic equation (eqn. (3.1)) and the simplest parabolic equation (eqn. (3.2)) with regular grid. The special topics of irregular grids and the treatment of nonlinearities are dealt with in subsequent sections.

The equations that describe single-phase flow in porous media are not unique to reservoir mechanics. Consequently, literature pertinent to the numerical solution of these equations is vast. The background of numerical analysis related to this problem can be found in numerous books (e.g., Lapidus, 1962; Smith, 1965; Von Rosenberg, 1969; Ames, 1969; Mitchell, 1969). Many other books have also been published and it is hard to single out one for reference. The advances in numerical analysis can be followed in journals such as the SIAM *Journal of Numerical Analysis*, *Numerische Mathematik*, *Mathematics of Computation*, etc. The literature concerning applications is scattered throughout several branches of engineering, in particular, heat transfer.

3.2 FINITE-DIFFERENCE APPROXIMATIONS

The following exposition has two purposes: (1) to define the terminology, and (2) to summarise the basic facts which will be required later for the development of special techniques. Full treatment of both theory and applications can be found in books on numerical solution of differential equations (e.g., Forsythe and Wasow, 1960; Henrici, 1962; Saul'jev, 1964; Richtmyer and Morton, 1967; Ames, 1969; Lapidus and Seinfeld, 1971) as well as in books devoted to numerical analysis (e.g., Hildebrand, 1956; Young and Gregory, 1973; Blum, 1972).

The basic idea of any approximate method is to replace the original problem by another problem that is easier to solve and whose solution is in some sense close to the solution of the original problem. As a simple example, consider the equation

$$\mathrm{A}\,U \equiv \frac{\mathrm{d}^2 U}{\mathrm{d}x^2} - q(x) = 0 \qquad 0 < x < L \tag{3.5}$$

with

$$U(0) = U(L) = 0$$

In the finite-difference approach, instead of trying to find a continuous, sufficiently smooth function $U(x)$ which satisfies (3.5), we seek only approximate values of the solution, denoted by u on a finite set of distinct points $x_1, x_2, \ldots, x_N$ inside the interval $(0, L)$. The points x_i are called grid points (also mesh points or net points). The differential equation is replaced by a set of algebraic equations relating values u_i at x_i for all points. These equations are called 'finite-difference equations' and the differential problem is thus reduced to an algebraic problem. If we can show that the discrete problem is 'close' to the original problem, then the values u_i will approximate the true solution $U_i = U(x_i)$ at grid positions x_i. The process of obtaining finite-difference equations that approximate a given differential equation is called 'discretisation'.

Three types of questions may be asked at this stage:

(a) How can a given differential equation be discretised?
(b) How can we ascertain that the finite-difference solution u_i is close to U_i in some sense, and what is the magnitude of the difference?
(c) What is the best method of solving the resulting system of algebraic equations?

The first two questions are discussed in this chapter. The third question is extremely important from the practical point of view, and involves two steps. First, whenever the finite-difference equations are nonlinear they must be linearised. This question will be considered in Section 3.7. The second step involves the solution of resulting matrix equations, and this important problem will be considered in the next chapter.

3.2.1 Discretisation in Space

Let us consider eqn. (3.5) with boundary conditions $U(0) = U(L) = 0$. Basically, there are three methods available for discretisation of any given operator A: the Taylor

series method, the integral method and the variational method (Forsythe and Wasow, 1960; Varga, 1962). These correspond to differential, integral and variational formulations of the conservation equation (eqn. (3.5)).

The problem to be solved is

$$\mathrm{A}U = 0$$

instead of this we solve

$$\mathrm{L}u = 0$$

where L is a finite-difference operator, approximating the differential operator A. Generally, we write

$$\mathrm{A}U_i = \mathrm{L}U_i + R_i \tag{3.6}$$

where L U_i is obtained by approximating the derivatives in the differential operator A and R_i is the remainder term usually referred to as the *truncation error* or the *local discretisation error*. This will be discussed in more detail later.

3.2.1.1 *Taylor Series Method*

Let us consider a uniform grid, with grid points at

$$x_0, x_1, \ldots, x_{N+1}$$

with $x_0 = 0$, $x_{N+1} = L$ and the grid spacing h defined by

$$h = x_{i+1} - x_i = L/(N+1)$$

Now expand U_{i+1} and U_{i-1} into Taylor series about U_i:

$$U_{i+1} = U_i + U_i'h + U_i''\frac{h^2}{2} + U_i'''\frac{h^3}{6} + U_i^{\mathrm{IV}}\frac{h^4}{24} + U_i^{\mathrm{V}}\frac{h^5}{120} + U_i^{\mathrm{VI}}\frac{h^6}{720} + \cdots \tag{3.7}$$

$$U_{i-1} = U_i - U_i'h + U_i''\frac{h^2}{2} - U_i'''\frac{h^3}{6} + U_i^{\mathrm{IV}}\frac{h^4}{24} - U_i^{\mathrm{V}}\frac{h^5}{120} + U_i^{\mathrm{VI}}\frac{h^6}{720} - \cdots \tag{3.8}$$

Using the above two expansions we can derive several difference approximations for U_i' and one for U_i''. Solving eqn. (3.7) for U_i' we have

$$U_i' = \frac{U_{i+1} - U_i}{h} + R_i^f \tag{3.9}$$

where

$$R_i^f = -U_i'' \frac{h}{2} - U_i'''' \frac{h^2}{6} - \cdots \tag{3.10}$$

In eqn. (3.9) the term $(U_{i+1} - U_i)/h$ is the *forward-difference approximation* for the derivative U_i'. This is obtained by assuming that R_i^f is small.

Similarly, by rearranging eqn. (3.8) we obtain

$$U_i' = \frac{U_i - U_{i-1}}{h} + R_i^b \tag{3.11}$$

where

$$R_i^b = U_i'' \frac{h}{2} - U_i'''' \frac{h^2}{6} + \cdots \tag{3.12}$$

In eqn. (3.11) the term $(U_i - U_{i-1})/h$ is the *backward-difference approximation* for the derivative U_i' and R_i^b is the local discretisation error term for the backward-difference approximation.

The *central-difference approximation* of U_i' is obtained by subtracting eqn. (3.8) from (3.7) and rearranging:

$$U_i' = \frac{U_{i+1} - U_{i-1}}{2h} + R_i^c \tag{3.13}$$

where

$$R_i^c = -U_i'''' \frac{h^2}{6} - U_i^{\mathrm{V}} \frac{h^4}{120} - \cdots \tag{3.14}$$

Here $(U_{i+1} - U_{i-1})/2h$ provides an approximation for U_i'.

So far only the first derivative has been considered. An approximation for the second derivative is accomplished by adding eqns. (3.7) and (3.8) and rearranging:

$$U_i'' = \frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} + R_i^2 \tag{3.15}$$

where

$$R_i^2 = -U_i^{\mathrm{IV}} \frac{h^2}{12} - U_i^{\mathrm{VI}} \frac{h^4}{360} - \cdots \tag{3.16}$$

In eqn. (3.15) the term

$$\frac{1}{h^2} \Delta^2 U_i = \frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} \tag{3.17}$$

is the *central-difference approximation* for U_i'' and R_i^2 is the corresponding remainder term. Equation (3.17) defines the linear operator Δ^2.

As an example, consider the differential operator, A, defined by eqn. (3.5). Using the central-difference approximation of the derivative, we have

$$\mathrm{A}U_i = \frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} - q_i + R_i^2 \tag{3.18}$$

where $q_i = q(x_i)$. Comparing the above expression with eqn. (3.6) we see that

$$\begin{aligned} \mathrm{L}U_i &\equiv \frac{1}{h^2} \Delta^2 U_i - q_i \\ R_i &= R_i^2 \end{aligned} \tag{3.19}$$

Generally, it is not possible for us to obtain the exact solution U_i. Instead we solve the problem

$$\mathrm{L}u_i = \frac{1}{h^2} \Delta^2 u_i - q_i = 0 \tag{3.20}$$

where u_i is a finite-difference approximation for U_i and eqn. (3.20) is a finite-difference approximation for eqn. (3.5).

The other finite-difference approximations derived here will be utilised later in more general problems.

An approach which is equivalent to the above treatment is provided by the theory of interpolation by polynomials (e.g., Carnahan *et al.*, 1969; Kelley, 1968). Since the difference formula for $U^{(p)}$ is the pth derivative of a polynomial which interpolates U at a given set of points, it follows immediately that:

(a) a difference operator for $U^{(p)}$ must involve at least $p + 1$ distinct values U_i, and

(b) the approximation for $U^{(p)}$ is exact if the function U is a polynomial of degree $q \leq p$.

This approach is convenient for the derivation of higher order approximations (see Exercise 3.1 at the end of this chapter).

3.2.1.2 *Integral Method*

The integral method is closer to the physical meaning of the equation. Note that eqn. (3.17) could be derived by first writing $U'' = (U')'$ and approximating this as

$$(U')' \simeq (U'_{i+1/2} - U'_{i-1/2})/h$$

then approximating U′ as

$$U'_{i+1/2} \simeq (U_{i+1} - U_i)/h \qquad U'_{i-1/2} \simeq (U_i - U_{i-1})/h$$

This procedure is, in fact, the derivation of finite-difference equations by the integral method. This—in contrast to Taylor series method—requires the additional concept of a block (or mesh region), which is the region bounded by boundaries between the given point and its neighbours. In one dimension, we replace x-axis by a 'rod' of constant cross-section A and the block for grid point i is defined by the boundaries at $x_{i+½}$ and $x_{i-½}$ (Fig. 3.1).

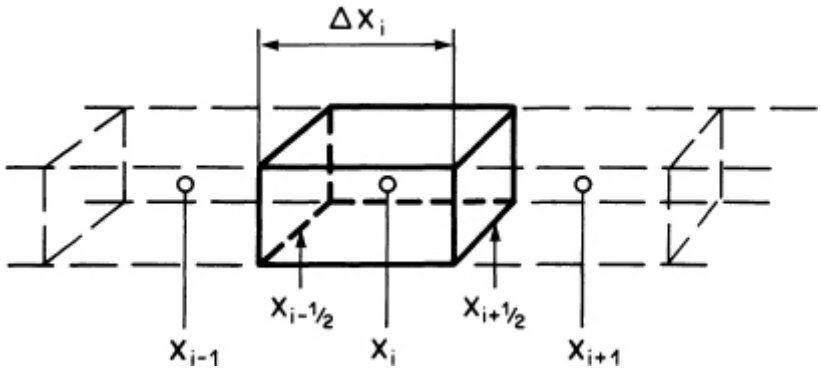


FIG. 3.1. One-dimensional discretisation into blocks.

Since eqn. (3.5) expresses conservation of mass, as discussed in detail in Chapter 2, it can be written in integral form as

$$A\left[\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 U}{\partial x^2}\,\mathrm{d}x - \int_{x_{i-1/2}}^{x_{i+1/2}} q\,\mathrm{d}x\right] \tag{3.21}$$

$$= A\left[\left.\frac{\partial U}{\partial x}\right|_{i+1/2} - \left.\frac{\partial U}{\partial x}\right|_{i-1/2}\right] - A\left[\int_{x_{i-1/2}}^{x_{i+1/2}} q\,\mathrm{d}x\right] = 0$$

where the integration has been carried over the block volume and the integral converted using Green's Theorem. Equation (3.21) expresses conservation of mass for block i, because

$$\left.A\frac{\partial U}{\partial x}\right|_{i\pm 1/2}$$

are interblock flow rates and

$$A\int_{x_{i-1/2}}^{x_{i+1/2}} q\,\mathrm{d}x$$

is the total source rate for block i. Now we can discretise eqn. (3.21) by the use of central differences for $\partial U/\partial x$ terms:

$$A\left[\left.\frac{\partial U}{\partial x}\right|_{i+1/2} - \left.\frac{\partial U}{\partial x}\right|_{i-1/2}\right] - A\int q\,\mathrm{d}x \tag{3.22}$$

$$\simeq A(u_{i+1} - 2u_i + u_{i-1})/h - Aq_i h = 0$$

where q_i approximates the mean value of q in block i. Dividing eqn. (3.22) by the volume of the block $V = Ah$ gives again eqn. (3.20). Note that integral method gives equations that are expressed in units of mass/time, while Taylor series method gives equations in units of mass/(volume × time). In practical applications, the first form, i.e., eqn. (3.22) is preferable because it gives symmetric matrices and is suitable for the calculation of material balances. These aspects are discussed in Section 3.7.

3.2.1.3 *Variational Method*

Finally, we briefly touch on the use of the variational method. The starting point here is the variational formulation of the conservation equation. As discussed at length in references on variational methods (Mikhlin, 1964; Courant and Hilbert, 1953; Hildebrand, 1965; Schechter, 1967) the solution U of eqn. (3.5) minimises the integral

$$I = \int_0^L \left[\frac{1}{2} \left(\frac{\partial \mathrm{V}}{\partial x} \right)^2 - q\mathrm{V} \right] \mathrm{d}x \tag{3.23}$$

among all functions V that are sufficiently smooth and satisfy the boundary conditions. At this stage we may choose to approximate directly the functional I, as shown by Varga (1962). The second possibility is to construct a class of functions v (usually polynomials), which approximate functions V and then solve the minimisation problem only for functions from this class. For example, let us assume that all approximating functions v have the form

$$v(x) = \sum_{i=1}^{i=N} c_i S_i(x) \tag{3.24}$$

where $S_i(x)$ are the so-called 'Chapeau' functions shown on Fig. 3.2. The functions S_i are called the basis functions and each S_i is non-zero only on the interval (x_{i-1}, x_{i+1}), called the support of S_i. It is easily seen that the class of functions defined by eqn. (3.24) consists of all functions v, that are continuous and piecewise linear between the points x_i. The functions v satisfy $v(0) = v(L) = 0$ and the values of v at x_i are equal to c_i. Substitution of eqn. (3.24) into eqn. (3.23) gives a quadratic form in c_i which possesses a minimum for certain values of $v_i = c_i$. The minimisation process (outlined in Exercise 3.2) leads to a set of algebraic equations for v_i, identical with eqn. (3.20).

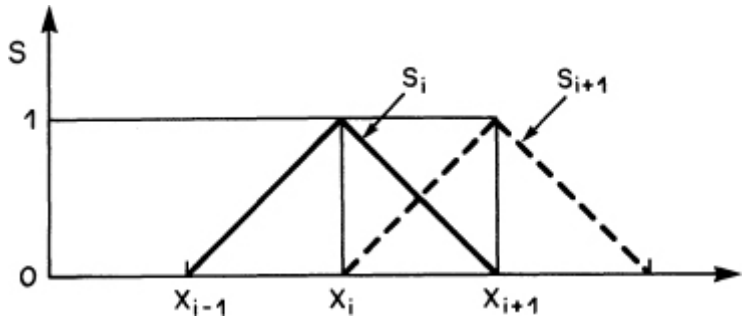


FIG. 3.2. Chapeau functions.

Of course, we can choose different basis functions that will generate different algebraic equations. This approach in its broadest sense encompasses all variational-type methods (Ritz, Galerkin, finite-element, weighted-residual and other methods). For this reason, and also because in this approach we work with continuous functions rather

than with discrete values, the algebraic equations generated by these methods are *not* referred to as finite difference equations.

3.2.1.4 *Discretised Problem*

All of the methods of discretisation in space yield a set of algebraic equations for the boundary value problem (eqn. (3.5)):

$$(u_{i+1} - 2u_i + u_{i-1})/h^2 = q_i, \qquad i = 1, 2, \ldots, N \tag{3.25}$$
$$u_0 = u_{N+1} = 0$$

These equations can be written in matrix form as

$$\frac{1}{h^2}\mathbf{E}\boldsymbol{u} = -\boldsymbol{q} \tag{3.26}$$

where **E** is a symmetric tridiagonal matrix

$$\mathbf{E} = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & -1 & 2 \end{bmatrix} \tag{3.27}$$

and vectors $\boldsymbol{u}$ and $\boldsymbol{q}$ are defined as

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_N \end{bmatrix} \qquad \boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \\ q_N \end{bmatrix}$$

Methods for the solution of eqn. (3.26) are considered in the next chapter.

3.2.2 Discretisation in Time

Consider the parabolic equation (eqn. (3.2))

$$\frac{\partial^2 U}{\partial x^2} = \frac{\partial U}{\partial t} + q \tag{3.28}$$

with the initial condition $U(x, 0) = U^0(x) = f(x)$. We have already discussed the discretisation of the left-hand side in terms of grid values u_i. For a time-dependent problem, these values are functions of time, $u_i = u_i(t)$. Then for each grid point x_i, the right-hand side of eqn. (3.28) may be replaced by $du_i/dt + q_i$ and we obtain a system of ordinary differential equations

$$(u_{i+1} - 2u_i + u_{i-1})/h^2 = \frac{du_i}{dt} + q_i \qquad i = 1, 2, \ldots, N \tag{3.29}$$

with initial conditions $u_i(0) = U^0(x_i)$.

Equation (3.29) is discretised in space, but continuous in time; for this reason this process is called semi-discretisation (Varga, 1962).

The usual way to solve the system of ordinary differential equations (ODEs) (3.29) is to discretise the time derivative as well. One approach is to apply the methods developed for ODEs, as discussed in Section 3.3.3. However, because the number of simultaneous equations is usually large, special numerical methods for parabolic equations have been developed independently of the numerical methods for ODEs.

Let us discretise the time co-ordinate into time steps Δt, and seek numerical solution only on discrete levels $t_0 = 0$, $t_1 = \Delta t, \ldots$, $t_n = n\ \Delta t, \ldots$. Then each function $u_i(t)$ is approximated by values u_i^n, $n = 1,2, \ldots$. The derivative du_i/dt is then expressed in terms of u_i^n.

The simplest method is obtained when du_i/dt is approximated by the forward difference:

$$\left(\frac{du_i}{dt}\right)^n \simeq (u_i^{n+1} - u_i^n)/\Delta t$$

Substitution of this into eqn. (3.29) with the left side written at the level n, results in the classical explicit method:

$$\alpha \Delta^2 u_i^n \equiv \alpha(u_{i-1} - 2u_i + u_{i+1})^n = (u_i^{n+1} - u_i^n) + q_i \Delta t \tag{3.30}$$

$$i = 1, 2, \ldots, N \qquad n = 0, 1, \ldots$$

where $\alpha = \Delta t/h^2$.

Since at the first time step ($n = 0$) all $u_i^n = U^0(x_i i)$ are known, the only unknown in equation i is u_i^{n+1}. The equations can be solved explicitly point by point; hence the name 'explicit method'.

Another choice is to use the backward difference to approximate the time derivative

$$\left(\frac{\mathrm{d}u_i}{\mathrm{d}t}\right)^{n+1} \simeq (u_i^{n+1} - u_i^n)/\Delta t$$

and write the left side at the level $n + 1$:

$$\alpha \Delta^2 u_i^n \equiv \alpha(u_{i+1} - 2u_i + u_{i-1})^{n+1} = (u_i^{n+1} - u_i^n) + q_i \Delta t \qquad (3.31)$$

$$i = 1, 2, \ldots, N \qquad n = 0, 1, 2, \ldots$$

All u_i^{n+1} are unknown, therefore the N equations for a given time level n must be solved simultaneously. Consequently, eqn. (3.31) is called the (classical) 'implicit method'. In matrix notation, it can be written as

$$(\alpha \mathbf{E} + \mathbf{I})\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t \boldsymbol{q} \qquad n = 0, 1, 2, \ldots \qquad (3.32)$$

where **E** is the matrix defined by eqn. (3.27) and **I** is the identity matrix.

A number of other methods will be discussed later in this chapter. It should be mentioned that a unifying theory for time discretisation is provided by the Theory of Padé approximations (Varga, 1962).

3.2.3 Discretisation Errors

In developing finite-difference approximations to various derivatives, the remainder terms were neglected. It is worthwhile to consider at this stage the influence of these approximations and ask:

1. Does the solution u_i^n provide a reasonable approximation for U_i^{n+1}?
2. What is the behaviour of discretisation errors as the mesh sizes are refined?

These questions are considered next.

3.2.3.1 *Local Order of Approximation and Consistency*

Let us take as an example expression (3.9) for the approximation of U'. The remainder R_i^f given by eqn. (3.10) is called the *truncation* or the *local discretisation*

error of the approximation of U' by $(U_{i+1} - U_i)/h$. To evaluate the magnitude of R_i^f, we need higher derivatives of U, which are usually not known. However, some qualitative information about the behaviour of R_i^f for $h \to 0$ can be obtained. To do this, we need to introduce the 'asymptotic notation': A function f (h) is of 'order' p, if there is a constant K such that for all $h < h_0$, $|f(h)| \leq Kh^p$. We write this as

$$\mathrm{f}(h) = O(h^p).$$

In other words, f(h) tends to zero at least as fast as the pth power of h (the most common case is when p is a positive integer). On the other hand, notation

$$\mathrm{f}(h) = o(h^p)$$

means that f(h)/$h^p \to 0$ as $h \to 0$, i.e. f(h) tends to zero faster than h^p. (For a more detailed exposition, see the text by Ames, 1969.)

Applying this notation to

$$R_i^f = -U_i'' \frac{h}{2} - U_i'''' \frac{h^2}{6} - \cdots$$

we see that for $h \to 0$ the first term will eventually dominate and therefore

$$\lim_{h \to 0} R_i^f = \lim_{h \to 0} \left(-U_i'' \frac{h}{2} \right)$$

which implies

$$R^f = O(h) \tag{3.33}$$

Also, since U'' is not a function of h, $R^f \neq O(h^p)$ for any $p > 1$. The highest p for which $R = O(h^p)$ is called the *order* of the error R (and the approximation associated with R). Therefore the forward difference formula (3.9) is a first order approximation to U' and we can write

$$U_i' = (U_{i+1} - U_i)/h + O(h) \tag{3.34}$$

Similarly, we find

$$U_i' = (U_i - U_{i-1})/h + O(h) \tag{3.35}$$

$$U_i' = (U_{i+1} - U_{i-1})/2h + O(h^2) \tag{3.36}$$

$$U_i'' = (U_{i+1} - 2U_i + U_{i-1})/h^2 + O(h^2) \tag{3.37}$$

The error term in each of the above expressions can be made arbitrarily small by choosing a sufficiently small value of h. Such approximations are called *consistent*. More precisely a difference operator L is said to be a consistent approximation for the corresponding differential operator A at the point x_i, if the truncation error, R_i, satisfies

$$R_i \equiv \mathrm{A}U_i - \mathrm{L}U_i \to 0 \qquad \text{as} \qquad h \to 0 \tag{3.38}$$

Since the above definition applies to the point x_i only, a more general definition is more useful from a practical standpoint:

DEFINITION: *A difference operator L is consistent with the differential operator* A, *if* $\|\boldsymbol{R}\| \to 0$ *as* $h \to 0$ *where* $\|\boldsymbol{R}\|$ *is some norm of vector* $\boldsymbol{R}$ *containing elements* $\boldsymbol{R}_i$ (see Appendix A).

The above definition has been used, among others, by Richtmyer and Morton (1967) and Blum (1972). If only integer orders of approximation are considered, the above definition is equivalent to saying that

$$\text{L is consistent if } R_i = O(h^p) \text{ with } p \geq 1$$

and

$$\text{L is inconsistent if } R_i = O(h^0)$$

Note that consistency is a property of the difference operator and not of the solution.

3.2.3.2 *Convergence*

Let e_i be the error in the approximate solution at i,

$$e_i = U_i - u_i \tag{3.39}$$

This is called the error of the solution or the global discretisation error.

DEFINITION: *A difference operator* **L** *is convergent to the differential operator A, if* $\|\boldsymbol{e}\| \to 0$ *as* $h \to 0$.

The errors e_i are more important than R_i from a practical standpoint. To show their connection, let us evaluate

$$\mathrm{L}U_i - \mathrm{L}u_i$$

for the elliptic boundary value problem (eqn. (3.5)). By virtue of the definition (3.6)

$$\mathrm{L}U_i = -R_i \tag{3.40}$$

since $\mathrm{A}U_i = 0$. Now the finite-difference operator for this problem is defined by

$$\mathrm{L}u_i \equiv \frac{1}{h^2}\Delta^2 u_i - q_i = 0 \tag{3.41}$$

Subtracting eqn. (3.41) from (3.40) and using the definition (3.39) we have:

$$\begin{aligned}\mathrm{L}U_i - \mathrm{L}u_i &= -R_i \\ &= \frac{1}{h^2}\Delta^2 U_i - \frac{1}{h^2}\Delta^2 u_i - q(x_i) + q_i \\ &= \frac{1}{h^2}\Delta^2 e_i\end{aligned}$$

or

$$\frac{1}{h^2}\Delta^2 e_i + R_i = 0 \tag{3.42}$$

In the derivation of eqn. (3.42) we have utilised the fact that Δ^2 is a linear operator. Equation (3.42) illustrates the important relationship between local errors and errors of solution:

> *The errors of solution e_i satisfy the difference equation for u_i with the source q_i replaced by the local error, $-R_i$.*

For the example considered above it is easy to see that

$$\|\boldsymbol{e}\| \to 0 \quad \text{as} \quad \|\boldsymbol{R}\| \to 0$$

and consistency implies convergence. Furthermore, if $\|\boldsymbol{R}\| = O(h^p)$, then the solution of eqn. (3.42) will also go to zero as $O(h^p)$. An example of this is outlined in Exercise 3.3. This result is not valid in general, since the order of convergence of $\boldsymbol{e}$ also depends on the approximation of boundary conditions. Additional complications also arise when we consider time-dependent problems.

Convergence of the explicit method. Let us now consider the parabolic problem (eqn. (3.28)):

$$\mathrm{A}U \equiv \frac{\partial^2 U}{\partial x^2} - \frac{\partial U}{\partial t} - q = 0$$

The forward difference approximation for this problem is given by eqn. (3.30):

$$\Delta t \mathrm{L}u_i = \alpha \Delta^2 u_i^n - (u_i^{n+1} - u_i^n) - q_i \Delta t = 0 \tag{3.43}$$
$$i = 1, 2, \ldots, N \qquad n = 0, 1, 2, \ldots$$

Since

$$R_i = \mathrm{A}U_i^n - \mathrm{L}U_i^n = O(h^2) + O(\Delta t)$$

L is a consistent approximation. As we discovered for eqn. (3.41), we find that the errors e_i^n satisfy:

$$\alpha \Delta^2 e_i^n - (e_i^{n+1} - e_i^n) = -\Delta t R_i \tag{3.44}$$

Therefore the error e_i^{n+1} generated by errors e_i^n at the previous level and by R_i^n is

$$e_i^{n+1} = \alpha e_{i+1}^n + (1 - 2\alpha) e_i^n + \alpha e_{i-1}^n + \Delta t R_i \tag{3.45}$$

We can now estimate the growth of errors introduced at $t = 0$. Denote

$$E_n = \max_i |e_i^n|$$

and

$$M = \max_{i,n} |R_i^n|$$

By the application of the triangular inequality to eqn. (3.45) it follows that

$$E_{n+1} \leq 2\alpha E_n + |(1 - 2\alpha)E_n| + \Delta t M \tag{3.46}$$

Let us first consider the case when $\alpha \leq ½$. Since $(1 - 2\alpha)$ is positive, eqn. (3.46) reduces to

$$E_{n+1} \leq E_n + \Delta t M$$

The successive application of the above relation yields

$$E_n \leq E_0 + n\,\Delta t M = E_0 + t_n M \tag{3.47}$$

where E_0 is the maximum error due to the approximation of initial conditions. There are two important conclusions we can draw from eqn. (3.47):

1. If $E_0 = 0$, then $E_n \to 0$ as the time step and grid spacing go to zero. This proves the convergence of the explicit method for $\alpha \leq ½$.
2. If $E_0 \neq 0$, then since M is bounded, the error at any step, n, is bounded. We will come back to this point when we discuss stability of finite-difference schemes.

So far we have only considered the case when $\alpha \leq ½$. Let us now consider the case of $\alpha > ½$. Now eqn. (3.46) gives

$$E_{n+1} \leq (4\alpha - 1)E_n + \Delta t M = \beta E_n + \Delta t M \tag{3.48}$$

where $\beta = 4\alpha - 1 > 1$. By successive application of the above relation, we obtain

$$E_n \leq \beta^n E_0 + \Delta t M \sum_{l=0}^{n-1} \beta^l \tag{3.49}$$

Again, there are two important points to be made about eqn. (3.49):

1. If $E_0 = 0$, the second term of eqn. (3.49) may not go to zero as the time step and grid spacing go to zero. This shows that we may not get convergence for this problem.
2. If $E_0 \neq 0$, then it is clear that initial errors may—and in practice do—amplify exponentially with number of time steps. This point is considered in more detail along with the discussion of stability.

Here we have an example of a scheme that is *consistent* but only *conditionally convergent*.

Convergence of the implicit method. An analysis, similar to the one given for the explicit method, shows that the backward difference approximation of $\partial U/\partial t$ in eqn. (3.31) results in a method for which

$$\|e\| \to 0 \quad \text{as} \quad \Delta t,\, h \to 0$$

regardless of the value of α. This will be shown through the stability analysis in Exercises 3.4 and 3.5.

3.2.3.3 *Stability*

The concept of stability is important for time-dependent problems. The following statement provides a general definition of stability.

DEFINITION: *A numerical algorithm is considered stable if any errors introduced at some stage of computation do not amplify during subsequent computations.*

In a more general sense, stability means that the machine-computed solution depends continuously on the initial and boundary conditions. Several definitions exist in the literature; they are discussed, e.g., by Richtmyer and Morton (1967) and Forsythe and Wasow (1960).

For elliptic equations, the approximation will always be stable if it is consistent (including the approximation of boundary conditions) and if the method used to solve the matrix equation is itself stable against rounding errors.

The situation is different for parabolic equations, which are approximated at a sequence of time steps. Any error e_i^n introduced on some time level n will propagate in time and affect the solutions at all later time levels $m > n$. This is true not only of round-off errors, but also of discretisation errors; therefore, stability is always necessary if meaningful solutions are to be achieved.

Two kinds of errors may be amplified if the numerical scheme is unstable:

1. *Rounding Errors:* When the finite-difference equations are solved on computers of finite word size, the computed values, denoted by u_i^*, will be different from u_i. The difference $u_i^* - u_i$, called 'departure', is caused by rounding errors and depends on the word length of the machine and the organisation of computations.

2. *Discretisation Errors:* When, for example, initial conditions of the problem must be approximated for the purposes of numerical solution, some errors are introduced and $E_0 \neq 0$. During the discussion of convergence it was shown that the effect of a non-zero E_0 magnifies with time if a consistent numerical scheme does not converge. Although

not stated explicitly before, a numerical scheme for which the errors initiated by E_0 magnify with time is called *unstable.*

It is clear from the above discussion that *stability* and *convergence* are related properties. This result is contained in *Lax's Equivalence Theorem:*

> *For a consistent approximation, stability is a necessary and sufficient condition for convergence*. (Richtmyer and Morton, 1967).

This theorem is of great practical value because there are some relatively simple methods for the investigation of stability, while the direct proof of convergence is usually quite difficult for practical problems.

However, the theorem is valid only for 'properly posed' problems (Richtmyer and Morton, 1967, p. 39). Some nonlinear problems may not satisfy this condition, and convergence in such cases is not guaranteed by stability. An example of this situation is given in Chapter 5, Section 5.5.1, where a consistent, stable approximation to the hyperbolic equation does not converge to the true solution.

It should be clear by now that it is quite easy to see if a scheme is consistent. Two methods for the investigation of stability are discussed next.

Fourier series method. The solution of an initial value problem can be formally written in the form of Fourier series (Richtmyer and Morton, 1967). Similarly, the solution of difference equations can be written in terms of discrete Fourier series:

$$u_i^n = \sum_m A_m u_i^{n(m)} \qquad m = 1, 2, \ldots, N \tag{3.50}$$

where

$$u_i^{n(m)} = \xi_m^n e^{\sqrt{-1}\, ihm} \tag{3.51}$$

and the coefficients A_m are determined by initial and boundary conditions. The growth of the mth component $u_i^{n(m)}$ is determined by ξ_m, which is called the *amplification factor.* The difference equation will be stable if

$$|\xi_m| < 1 \tag{3.52}$$

The amplification factor is obtained by substitution of the trial solution (eqn. (3.51)) into the difference equation and solving for ξ_m. For example, for the explicit method (3.43), we obtain after manipulations outlined in Exercise 3.4,

$$\xi_m = 1 - 4\alpha \sin^2\left(\frac{mh}{2}\right) \tag{3.53}$$

The requirement that $|\xi_m| < 1$ for any m leads again to the condition $\alpha < 1/2$.

Matrix method. Since the Fourier series method considered only one particular solution, it does not take into account boundary conditions. In contrast to this, the matrix method deals with the complete solution including the boundary conditions.

Every two-level difference formula can be written in matrix form as

$$\boldsymbol{u}^{n+1} = \mathbf{B}\boldsymbol{u}^n + \boldsymbol{k} \qquad n = 0, 1, 2, \ldots \tag{3.54}$$

(e.g., for the explicit method (3.30), $\mathbf{B} = (\mathbf{I} - \alpha\mathbf{E})$, for the implicit method (3.32), $\mathbf{B} = (\alpha\mathbf{E} + \mathbf{I})^{-1}$). Let us introduce errors in the initial vector $\boldsymbol{u}^0$, so that the perturbed initial vector is $\boldsymbol{u}^{*,0} = \boldsymbol{u}^0 + \boldsymbol{\varepsilon}^0$ and assume that the rest of the calculations are performed without errors. Then the solution at a time level n will be $\boldsymbol{u}^{*,n} = \boldsymbol{u}^n + \boldsymbol{\varepsilon}^n$. Substitution into (3.54) gives $\boldsymbol{\varepsilon}^{n+1} = \mathbf{B}\,\boldsymbol{\varepsilon}^n$ and successive reduction of time level gives

$$\boldsymbol{\varepsilon}^n = (\mathbf{B})^n \boldsymbol{\varepsilon}^0 \tag{3.55}$$

In order that the equation be stable, $\|\boldsymbol{\varepsilon}^n\|$ must be bounded. A sufficient condition for this is that $\|\mathbf{B}\| < 1$ in a matrix norm consistent with the vector norm of $\boldsymbol{\varepsilon}$. Using the properties of consistent norms discussed in Appendix A, we have

$$\|\boldsymbol{\varepsilon}^n\| = \|\mathbf{B}^n\boldsymbol{\varepsilon}^0\| \le \|\mathbf{B}^n\|\|\boldsymbol{\varepsilon}^0\| = \|\mathbf{B}\|^n\|\boldsymbol{\varepsilon}^0\|$$

$$\|\boldsymbol{\varepsilon}^n\| \to 0 \qquad \text{as} \qquad n \to \infty \qquad \text{if} \qquad \|\mathbf{B}\| < 1$$

In particular, if we choose the Euclidean norm of $\boldsymbol{\varepsilon}$ the corresponding norm of $\mathbf{B}$ is the spectral norm (see Appendix A). If $\mathbf{B}$ is symmetric (which is frequently the case), its spectral norm is equal to its spectral radius:

$$\|\mathbf{B}\| = \rho(\mathbf{B}) = \max_i |\lambda_i|$$

where λ_i are the eigenvalues of $\mathbf{B}$. This leads to the following result:

For a symmetric $\mathbf{B}$ *the difference scheme is stable if*

$$|\lambda_{\max}| < 1 \tag{3.56}$$

where λ_{max} is the largest eigenvalue of **B**.

More general results are found in Varga (1962), Faddeev and Faddeeva (1963), and Richtmyer and Morton (1967) (see Appendix A). In general we have the following result:

> *For a difference scheme to be stable it is necessary and sufficient that all eigenvalues of* **B** *be less than unity in modules. Furthermore, a sufficient condition for stability is that any norm of* **B** *be less than unity.*

In simple cases of eqns. (3.30) or (3.32), the eigenvalues can be obtained explicitly and they turn out to be precisely the amplification factors ξ_m of the Fourier method (Exercise 3.5).

3.3 OTHER SELECTED METHODS

Two basic methods for obtaining an approximate solution of eqn. (3.25) have been discussed in the last section. The literature is full of other methods; however, it is not practical to present a full discussion of all available methods in this book. However, a brief discussion of some selected methods is presented.

3.3.1 Other Explicit Methods

An example of an explicit method has already been given by eqn. (3.30). Such methods are attractive because they do not require solution of systems of equations. However, they are only conditionally stable depending on the mesh ratio $\alpha = \Delta t/h^2$. Here we will consider some of the other explicit methods reported in the literature.

One of the early methods is due to Richardson (1910):

$$2\alpha\Delta^2 u_i^n = u_i^{n+1} - u_i^{n-1} \tag{3.57}$$

Although the method is of $O(\Delta t^2)$ accuracy, it is unstable for all values of α. DuFort and Frankel (1953) modified eqn. (3.57) by replacing the term $2u_i^n$ on the left by $u_i^{n+1} + u_i^{n-1}$, which gives

$$2\alpha(u_{i-1}^n - u_i^{n-1} + u_{i+1}^n - u_i^{n+1}) = u_i^{n+1} - u_i^{n-1} \tag{3.58}$$

This makes the method unconditionally stable. However, since the space derivatives are evaluated with values of u at three different time levels, this approximation is not

always consistent (Exercise 3.6). Both eqns. (3.57) and (3.58) are three-level equations (involving values of u at t^{n-1}, t^n and t^{n+1}. In order to start the solution, they require values at the first two levels. While $\boldsymbol{u}^0$ is the initial condition, $\boldsymbol{u}^1$ must be generated by some other method. This complicates programming and may also affect the accuracy.

Unconditionally stable explicit methods that do not require additional starting values have been introduced by Saul'jev (1964) and later with slight modifications by Larkin (1964) and Barakat and Clark (1966).

Saul'jev's method consists of two equations that are used alternatively on even and odd time steps:

$$\alpha(u_{i-1}^{n+1} - u_i^{n+1} + u_{i+1}^n - u_i^n) = u_i^{n+1} - u_i^n \qquad n = 0, 2, 4, \ldots \tag{3.59a}$$

$$\alpha(u_{i-1}^n - u_i^n + u_{i+1}^{n+1} - u_i^{n+1}) = u_i^{n+1} - u_i^n \qquad n = 1, 3, 5, \ldots \tag{3.59b}$$

These equations can be solved explicitly if eqn. (3.59a) is solved in order of increasing i, and eqn. (3.59b) in order of decreasing i. The method isunconditionally stable, but consistent only if $\alpha \to 0$ as $\Delta t \to 0$ (like the method of DuFort and Frankel).

Without the change of direction of sweeps, Saul'jev's method would suffer from accumulation of errors at the end of the line. Therefore, further improvement is expected if two sweeps in opposite direction are performed at every time step and averaged:

$$\alpha(v_{i-1} - v_i + u_{i+1}^n - u_i^n) = v_i - u_i^n \qquad i = 1, 2, 3, \ldots, N \tag{3.60a}$$

$$\alpha(u_{i-1}^n - u_i^n + w_{i+1} - w_i) = w_i - u_i^n \qquad i = N, N-1, \ldots, 1 \tag{3.60b}$$

$$u_i^{n+1} = \tfrac{1}{2}(v_i + w_i) \tag{3.60c}$$

This modification is due to Larkin (1964) and Barakat and Clark (1966).

Another class of explicit methods, called 'hopscotch methods' by Gourlay (1970) and Gourlay and McGuire (1971), is also based on an idea proposed by Saul'jev (1964). In its simplest form, the hopscotch algorithm is used by first calculating values of u_i^{n+1} for all odd i by the explicit method (3.30), and then the solution is filled in for the even values of i by an implicit method

$$\alpha(u_{i-1}^n - 2u_i^n + u_{i+1}^n) = u_i^{n+1} - u_i^n \qquad i = 1, 3, 5, \ldots \tag{3.61a}$$

$$\alpha(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}) = u_i^{n+1} - u_i^n \qquad i = 2, 4, 6, \ldots \tag{3.61b}$$

After (3.61a) is performed, u_{i-1}^{n+1} and u_{i+1}^{n+1} in the second equation are known and eqn. (3.61b) also becomes an explicit calculation. The method is unconditionally stable and

$O(\Delta x^2 + \Delta t^2)$. In a more general setting, hopscotch methods have been shown to be perturbations of the Crank–Nicolson method to be discussed later.

3.3.2 Other Implicit Methods

The classical implicit method, given by eqn. (3.31), is of only $O(\Delta t)$ accuracy. We will discuss here some additional methods which require solution of simultaneous equations.

Crank–Nicolson method. A second-order method in time is the Crank–Nicolson (C–N) formula:

$$\tfrac{1}{2}\alpha(\Delta^2 u_i^n + \Delta^2 u_i^{n+1}) = u_i^{n+1} - u_i^n \tag{3.62}$$

This method may be generalised by assigning different weighting factors to the explicit and implicit parts of the left-hand side. If θ is the weighting factor, $0 < \theta < 1$, the generalised method is

$$\theta\alpha\Delta^2 u_i^n + (1-\theta)\alpha\Delta^2 u_i^{n+1} = u_i^{n+1} - u_i^n \tag{3.63}$$

The usual method of arriving at eqn. (3.62) is to form an approximation to $\partial^2 u/\partial x^2$ at the n + ½ time level for which $u^{n+1} - u^n$ is a central difference of $\partial u/\partial t$. We want to demonstrate here a different approach which shows the equivalence of the C–N method to the explicit–implicit method of Saul'jev (1964). The method given by Saul'jev consists of alternating use of explicit and implicit steps:

$$\alpha\Delta^2 u_i^n = u_i^{n+1} - u_i^n \qquad n = 0, 2, 4, \ldots \tag{3.64a}$$

$$\alpha\Delta^2 u_i^{n+2} = u_i^{n+2} - u_i^{n+1} \qquad n = 0, 2, 4, \ldots \tag{3.64b}$$

By adding the above two equations, we obtain

$$\alpha(\Delta^2 u_i^n + \Delta^2 u_i^{n+2}) = u_i^{n+2} - u_i^n \tag{3.65}$$

The above equation is actually the C–N method with a time step of $2\Delta t$. The $n + 1$ time level corresponds to the n + ½ level which is not *explicitly* present in the C–N formula. Therefore, the C–N formula with time step Δt can be obtained by taking first an explicit step of $\Delta t/2$, followed by an implicit step of $\Delta t/2$. In order to obtain the generalised method (3.63), we only need to use (3.64a) with time step $\theta\Delta t$ and (3.64b) with time step $(1 - \theta)\Delta t$.

The form (3.64) shows that C–N is a predictor–corrector type of method and also provides an alternative way of programming.

The generalised method is unconditionally stable if $0 < \theta < 1$ and generally $O(h^2 + \Delta t)$. It becomes $O(h^2 + \Delta t^2)$ for $\theta = \frac{1}{2}$ and $O(\mathrm{h}^4 + \Delta t^2)$ for $\theta = \frac{1}{2} - h^2/12\Delta t$ (Crandall, 1955).

Some problems arise in the use of the C–N method with derivative boundary conditions (Smith, 1965; Keast and Mitchell, 1966).

An example of a multilevel implicit method is the formula (Lees, 1966; Richtmyer and Morton, 1967, p. 190):

$$\alpha\Delta^2 u_i^{n+1} = \tfrac{3}{2}(u_i^{n+1} - u_i^n) - \tfrac{1}{2}(u_i^n - u_i^{n-1}) \qquad n = 1, 2, \ldots \tag{3.66}$$

The weighting in this case is on the right-hand side. The scheme is unconditionally stable and $O(h^2 + \Delta t^2)$, but it needs auxiliary starting values (Exercise 3.7).

Richtmyer and Morton (1967) present a compilation of several other multilevel difference schemes.

3.3.3 ODE Methods

As we have seen in Section 3.2.2, once the space derivatives are discretised, a system of ordinary differential equations (ODE) is obtained (eqn. (3.29)). We can then apply (at least in principle) any of the numerous methods developed for ODEs to advance the solution in time. In fact, all methods presented thus far can be derived from the ODE methods. In this section we will present some of the old methods in a new setting and some new methods.

Consider, for example, the boundary value problem

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{3.67}$$

By discretisation of the right-hand side we obtain a set of ordinary differential equations:

$$\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = \frac{1}{h^2}\Delta^2 u_i(t) \qquad i = 1, 2, \ldots, N \tag{3.68}$$

which can also be written in matrix form as

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -\frac{1}{h^2}\mathbf{E}\boldsymbol{u}(t) \tag{3.69}$$

The matrix **E** has been defined by (3.27).

In the ODE literature, the system of equations to be solved are usually written as

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{u},t) \tag{3.70}$$

hence for our example

$$\boldsymbol{f} = -\frac{1}{h^2}\mathbf{E}\boldsymbol{u}(t) \tag{3.71}$$

All ODE methods may be viewed as approximations to

$$\boldsymbol{u}^{n-1} = \boldsymbol{u}^n + \int_n^{n+1} \boldsymbol{f}\,\mathrm{d}t$$

3.3.3.1 *Euler's Method*

The Euler's method (Henrici, 1962) for eqn. (3.70) is an explicit method given by

$$u_i^{n+1} = u_i^n + \Delta t f_i(\boldsymbol{u}^n, t^n) = u_i^n + \alpha\,\Delta^2 u_i(t^n) \qquad i = 1, 2, \ldots, N \tag{3.72}$$

or in matrix form as

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n - \alpha\mathbf{E}\boldsymbol{u}^n \\ &= (\mathbf{I} - \alpha\mathbf{E})\boldsymbol{u}^n \end{aligned} \tag{3.73}$$

which is exactly the classical explicit method given earlier by eqn. (3.30).

3.3.3.2 *Modified Euler's Method*

A very popular, explicit, second-order (in Δt) method is provided by the following two-step process:

1. First we predict the solution by Euler's method

$$u_i^{*,n+1} = u_i^n + \Delta t f_i(\boldsymbol{u}^n, t^n) \qquad i = 1, 2, \ldots, N \tag{3.74}$$

2. The predicted value is used to find the solution at $n + 1$ level by

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{2}[f_i(\boldsymbol{u}^n, t^n) + f_i(\boldsymbol{u}^{*,n+1}, t^{n+1})] \qquad (3.75)$$

$$i = 1, 2, \ldots, N$$

In matrix form the method may be written as

$$\boldsymbol{u}^{n+1} = \tfrac{1}{2}[\mathbf{I} + (\mathbf{I} - \alpha\mathbf{E})^2]\boldsymbol{u}^n \qquad (3.76)$$

and the stability of method depends upon the spectral radius of the matrix on the right-hand side. This method is known in the literature by many different names. We can generalise the methods in the form of second-order Runge–Kutta-type methods (Henrici, 1962):

$$u_i^* = u_i^n + \frac{\Delta t}{2b} f(\boldsymbol{u}^n, t^n) \quad | \quad t^* = t^n + \frac{\Delta t}{2b} \qquad (3.77)$$

$$u_i^{n+1} = u_i^n + \Delta t[(1 - b)f(\boldsymbol{u}^n, t^n) + bf(u^*, t^*)] \qquad (3.78)$$

For $b = 0$ the above scheme reduces to the Euler's method. For $b \neq 0$ the process is second order. For example with $b = ½$ we obtain the improved Euler method discussed earlier.

3.3.3.3 *The Classical Runge–Kutta Method*

A fourth-order explicit method for eqn. (3.70) is the classical Runge–Kutta method (Henrici, 1962):

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{\Delta t}{6}(\boldsymbol{a}_1 + 2\boldsymbol{a}_2 + 2\boldsymbol{a}_3 + \boldsymbol{a}_4) \qquad (3.79)$$

where

$$\begin{aligned}
\boldsymbol{a}_1 &= \boldsymbol{f}(\boldsymbol{u}^n, t^n) \\
\boldsymbol{a}_2 &= \boldsymbol{f}(\boldsymbol{u}^n + \tfrac{1}{2}\Delta t\boldsymbol{a}_1, t^n + \tfrac{1}{2}\Delta t) \\
\boldsymbol{a}_3 &= \boldsymbol{f}(\boldsymbol{u}^n + \tfrac{1}{2}\Delta t\,\boldsymbol{a}_2, t^n + \tfrac{1}{2}\Delta t) \\
\boldsymbol{a}_4 &= \boldsymbol{f}(\boldsymbol{u}^n + \Delta t\boldsymbol{a}_3, t^n + \Delta t)
\end{aligned}$$

The matrix form of this method for the linear case is considered in Exercise 3.8. This and the modified Euler's method belong to a class of methods known as Explicit

Runge–Kutta methods. These methods are, as expected, *conditionally stable.*

3.3.3.4 *Implicit Runge–Kutta-type Methods*

The *explicit Runge–Kutta* type methods discussed above allow us to obtain high-order approximations for the time derivative, but they suffer from conditional stability. For nonlinear problems it is often more desirable to use unconditionally stable *implicit Runge–Kutta*-type methods. A general class of these methods is discussed by Gear (1971). Here we will present a few selected methods.

Second-order methods. The linear partial differential equation

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{u}) = \mathbf{A}\boldsymbol{u} \tag{3.80}$$

may be solved by

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \tfrac{1}{2}(\boldsymbol{a}_1 + \boldsymbol{a}_2) \tag{3.81}$$

where

$$\boldsymbol{a}_1 = \Delta t \boldsymbol{f}(\boldsymbol{u}^n) = \Delta t \mathbf{A}\boldsymbol{u}^n \tag{3.82}$$

$$\boldsymbol{a}_2 = \Delta t \boldsymbol{f}(\boldsymbol{u}^{n+1}) = \Delta t \mathbf{A}\boldsymbol{u}^{n+1} \tag{3.83}$$

Substituting eqns. (3.82) and (3.83) into eqn. (3.80) we see that this is precisely the C–N method. The above scheme may also be applied to nonlinear problems.

Another second-order method is given by (Rosenbrock, 1963)

$$\boldsymbol{a}_1 = \Delta t(\boldsymbol{f}(\boldsymbol{u}^n) + b\mathbf{J}^n\boldsymbol{a}_1) \tag{3.84}$$

$$\boldsymbol{a}_2 = \Delta t(\boldsymbol{f}(\boldsymbol{u}^n + \beta\boldsymbol{a}_1) + b\mathbf{J}^n\boldsymbol{a}_2) \tag{3.85}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \boldsymbol{a}_2$$

where $\mathbf{J}$ is the Jacobian of $\boldsymbol{f}$ with elements $(\partial f_i/\partial u_j)$ and

$$b = 1 - \frac{\sqrt{2}}{2} = 0{\cdot}29289 \tag{3.86}$$

$$\beta = (\sqrt{2} - 1)/2 = 0{\cdot}207106 \tag{3.87}$$

For example when this process is applied to the linear problem (3.80)

$$\mathbf{J} = \mathbf{A} \tag{3.88}$$

and the method is given by the following two-stage process:

$$\left(\mathbf{A} - \frac{1}{b\Delta t}\mathbf{I}\right)\boldsymbol{a}_1 = -\frac{1}{b}\mathbf{A}\boldsymbol{u}^n \tag{3.89}$$

$$\left(\mathbf{A} - \frac{1}{b\Delta t}\mathbf{I}\right)\boldsymbol{a}_2 = -\frac{1}{b}\mathbf{A}\boldsymbol{u}^n - \frac{\beta}{b}\mathbf{A}\boldsymbol{a}_1 \tag{3.90}$$

This process requires only slightly more work than the C–N method, if the inverse of $(\mathbf{A} - (1/b\Delta t)\mathbf{I})$ can be stored. The method has some advantages over the C–N method as discussed in Exercise 3.9.

A third-order method. An example of a third-order implicit Runge–Kutta-type method is (Calahan, 1968):

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \tfrac{1}{4}\boldsymbol{a}_1 + \tfrac{3}{4}\boldsymbol{a}_2 \tag{3.91}$$

with

$$\boldsymbol{a}_1 = \Delta t[\boldsymbol{f}(\boldsymbol{u}^n) + \beta\mathbf{J}(\boldsymbol{u}^n)\boldsymbol{a}_1] \tag{3.92}$$

$$\boldsymbol{a}_2 = \Delta t[\boldsymbol{f}(\boldsymbol{u}^n + \gamma\boldsymbol{a}_1) + \beta\mathbf{J}(\boldsymbol{u}^n)\boldsymbol{a}_2] \tag{3.93}$$

$$\beta = \tfrac{1}{2}(1 + \sqrt{3}/3) = 0{\cdot}78867 \tag{3.94}$$

$$\gamma = -\frac{2}{\sqrt{3}} = -1{\cdot}154700 \tag{3.95}$$

Derivation of the matrix form of this method is left as an exercise to the reader.

A thorough discussion of methods for the numerical solution of ODEs including 'stiff equations, is given by Lapidus and Seinfeld (1971), and Gear (1971). Methods of Runge–Kutta type and their applications to nonlinear reservoir simulation problems are discussed by Price *et al.* (1978).

3.3.4 Comparison of Methods

The question which method is the best does not have a unique answer. The best choice is dictated to a large extent by the particular problem, especially for nonlinear equations.

In general, explicit methods require less work per time step than implicit methods, but the time step for explicit methods is limited by stability and accuracy considerations. As we progress from explicit to implicit methods, the work per time step increases and so does stability. This is even more pronounced in multidimensional nonlinear problems. For linear and mildly nonlinear problems second-order C–N-type methods are most popular, but for strongly nonlinear problems the explicit component of the C–N method may limit stability.

From the programming point of view, explicit methods are extremely simple and have modest storage requirements. All implicit methods require some algorithm for the solution of simultaneous linear equations and additional storage to perform these computations. These questions are discussed in detail in Chapters 4, 6 and 8.

The approach via ODE (sometimes also called the 'method of lines') appears attractive for several reasons. The theory of ODEs is at an advanced stage and there is a wide variety of methods to choose from, many of them suitable for nonlinear problems. Some of the higher-order methods like Runge–Kutta require much more computer work per time step than simpler, lower-order methods. For a higher-order method to be competitive with lower-order methods, it must allow the use of correspondingly larger time steps for any specified accuracy. This may be in conflict with time-step limitations imposed by stability considerations. Also, the space truncation error is not affected by the choice of the method for $\partial u/\partial t$. When the space truncation error dominates, improvements in the accuracy of time approximation through the use of higher-order methods cannot be justified. In order to achieve an adequate space approximation the number of simultaneous ODEs is usually large. Some ODE methods that are useful for a single equation, quickly become impractical for large systems (e.g., predictor–corrector methods with iteration on corrector).

Work of Wallis and Aziz (1975) with the problems of pollutant dispersion into the atmosphere shows the superiority of ODE methods under certain conditions. In problems of this type it is necessary to take small time steps (smaller than the stability limit) for reasons of accuracy requirements; this makes ODE methods attractive. Another advantage of ODE methods is that they can be automated for the solution of different problems much more easily than methods developed for partial differential equations (PDEs). Several general-purpose software packages for the solution of systems of partial differential equations are based on ODE methods (Carver, 1973; Cardenas, 1973). These packages usually reduce the human effort required to put a problem on the computer, but this is gained at the expense of increased computer time and storage.

It appears that for cases where the same PDEs have to be solved over and over for different problems, greatest success is achieved by taking full advantage of the peculiarities of the problem in developing simulation programs.

3.4 GRID SYSTEMS AND BOUNDARY CONDITIONS

The two topics of this section are interrelated, because the grid system used determines the form of the boundary conditions. We will first show two methods of grid construction (point-distributed and block-centred grids) and the associated boundary conditions for uniform spacing, and then discuss irregular spacings.

It is necessary at this point to introduce some notation. Regardless of the method of discretisation or the type of boundary conditions, N will be the number of finite-difference equations (i.e. number of unknowns at a given time step) to be solved.

The range of the index i will depend upon the type of boundary conditions and the method of approximation and it can vary from 0 to $N + 1$. These ideas will become clearer after we discuss the boundary conditions.

3.4.1 Two Methods of Grid Construction

Given a reservoir of length L and uniform cross-section A, we can construct a grid system of M points in basically two ways:

(a) Place the first and the last grid points at $x = 0$ and $x = L$, respectively, and distribute the rest of the points uniformly between them (Fig. 3.3 shows this for $M = 5$). In order to determine the volumes of blocks associated with each grid point, we place the block boundaries (dashed lines in Fig. 3.3) midway between the grid points. Thus, $\Delta x = L/(M - 1)$ and the volumes are $V = \Delta xA$, *except* for the points on the boundaries which have volumes of $V/2$. This method of grid construction has been called 'the point-distributed grid' method by Settari and Aziz (1972).

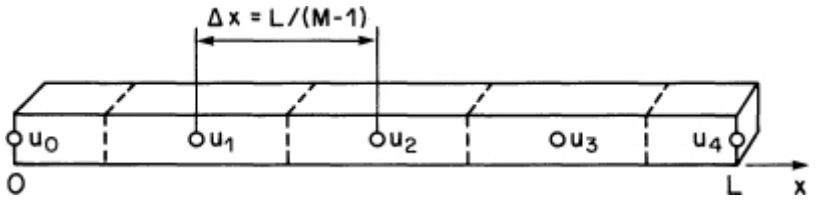


FIG. 3.3. Point-distributed grid.

(b) We can also divide the length L into M equal blocks and then locate grid points in the centres of these blocks, as shown on Fig. 3.4. The blocks are smaller for this case than in the case of the point-distributed grid, because now $\Delta x = L/M$. Furthermore, there are no points at the boundaries. This method has been used extensively by petroleum engineers and it is usually referred to as the 'block-centred grid' method.

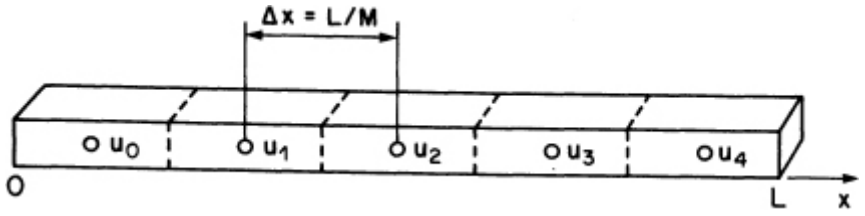


FIG. 3.4. Block-centred grid.

At this stage (i.e., for uniform grid spacing) the only difference between the two methods is in the treatment of boundary conditions. However, we will see later that when arbitrary grid spacing is considered, the analysis shows that the point-distributed grid is the correct approach as opposed to the block-centred grid. The underlying principle is that the block boundaries must be placed between the grid points and not vice versa, since the differential equation is approximated at grid points and not at the boundaries. Detailed discussion of these points follows in Section 3.5.1.

3.4.2 Boundary Conditions

The reservoir being simulated interacts with its surroundings through the conditions specified at its boundaries. It is essential that the boundary conditions be formulated and approximated so that the desired interaction of the reservoir with its surroundings takes place. In this section we will present most of the commonly encountered boundary conditions along with methods of approximating them.

3.4.2.1 *Boundary Conditions of the* 1*st Kind*

The conditions of the 1st kind (Dirichlet) specify the value of U at the boundary. In reservoir simulation, Dirichlet boundary conditions arise when we specify pressure at the reservoir boundary or at the well.

Suppose that at $x = 0$, the boundary condition for the time-dependent eqn. (3.28) is

$$U(0, t) = \mathrm{f}_1(t) \tag{3.96}$$

For the point-distributed grid, the finite-difference boundary condition is simply

$$u_0^n = f_1(t^n) \qquad n = 0, 1, \ldots \tag{3.97}$$

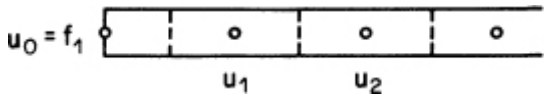


FIG. 3.5. Dirichlet boundary condition for point-distributed grid.

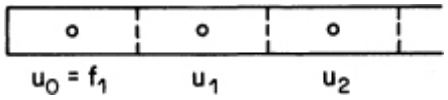


FIG. 3.6. Dirichlet boundary condition for block-centred grid.

Equation (3.97) is used whenever u_0^n is needed in the difference equation, and the difference equation is not needed for the first grid point. Therefore the first unknown, u_1, is associated with the second grid point (Fig. 3.5).

For the block-centred grid, the closest point from the boundary is at $\Delta x/2$ and the value of u_0 must be extrapolated to this point. The simplest possibility is to write (Fig. 3.6)

$$u_0^n = f_1(t^n) + O(\Delta x) \tag{3.98}$$

which is only a first-order approximation. A second-order approximation is (Fig. 3.7):

$$\tfrac{1}{2}(3u_1^n - u_2^n) = f_1(t^n) + O(\Delta x^2) \tag{3.99}$$

A practical disadvantage of this formula is that eqn. (3.99) must be included in the set of difference equations to be solved. For this reason, the block-centred grid is sometimes modified by use of half-blocks at the boundary (this, in fact, converts it to the point-distributed grid at the boundaries).

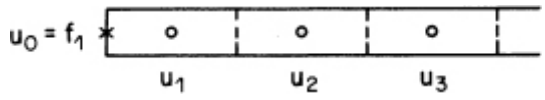


FIG. 3.7. Dirichlet boundary condition for block-centred grid.

3.4.2.2 *Boundary Conditions of the 2nd Kind*

For the pressure equation, the conditions of the 2nd (Neumann) kind express flow rate across the boundary and can be used to specify production rate, known influx from aquifer or flow from parts of the reservoir outside the simulated domain. An alternative approach, to be discussed in the next section, is to express flow across boundaries through the source term $q(x, t)$.

Let the boundary condition at $x = 0$ be

$$\frac{\partial U}{\partial x} = \mathrm{f}_2(t) \tag{3.100}$$

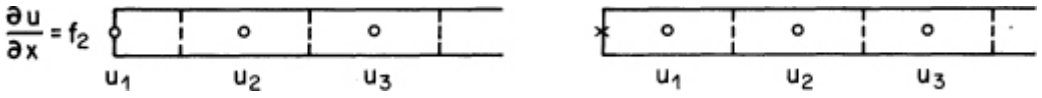


FIG. 3.8. Neumann boundary condition. Left, point-distributed grid; right, block-centred grid.

An $O(\Delta x)$ method is to approximate the derivative using inside points (Fig. 3.8)

$$\mathrm{f}_2(t^n) \simeq (u_2^n - u_1^n)/\Delta x \tag{3.101}$$

This is a poor approximation for the derivative at the boundary, especially for the block-centred grid.

A second-order method is the commonly used 'reflection technique'. According to Fig. 3.9 introduce an auxiliary point outside the boundary, with u_0 as the value of the unknown at this point. Let us first consider the point-distributed grid shown in Fig. 3.9 (left). The boundary condition of eqn. (3.100) is discretised using central difference at $x = 0$:

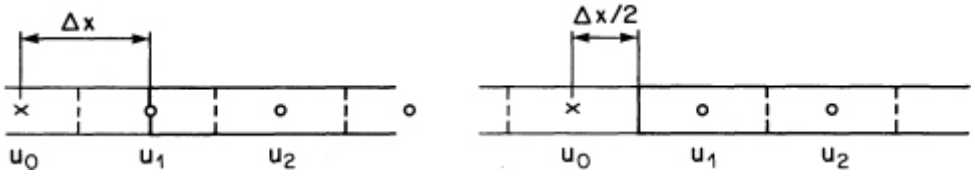


FIG. 3.9. Reflection technique. Left, point-distributed grid; right, block-centred grid.

$$\mathrm{f}_2(t^n) = (u_2^n - u_0^n)/2\,\Delta x + O(\Delta x^2) \tag{3.102}$$

and this equation is used to eliminate u_0^n from the difference equation written for the point $x = 0$.

The same procedure can be applied to the block-centred grid, shown in Fig. 3.9 (right):

$$f_2(t^n) = (u_1^n - u_0^n)/\Delta x + O(\Delta x^2) \tag{3.103}$$

The truncation error of eqn. (3.103) is exactly one-half of the error of eqn. (3.102). However, we have seen in Section 3.3.2, that these errors act as sources in the equations for errors of solution, viz. eqn. (3.42). For reasons that will become clear when we consider an arbitrary spacing, *the effect of the source term is proportional to the volume of the boundary block*, which is larger (exactly twice for same Δx) for the block-centred grid. Consequently, error introduced due to the boundary approximation should be about the same for both grids.

3.4.2.3 *Boundary Conditions of the 3rd Kind*

The boundary conditions of the 3rd kind are obtained by a combination of the previous two conditions:

$$a\frac{\partial u}{\partial x} + bu = f_3(t) \tag{3.104}$$

Mixed boundary conditions are quite common in the heat transfer literature. In fluid flow problems they occur in the following situation:

Consider reservoir I which is connected at $x = 0$ with another reservoir of known average pressure behaviour denoted by $U_{II}(t)$. We want to include the influence of the second reservoir through the boundary condition at $x = 0$ (Fig. 3.10). The flow from reservoir II to reservoir I will be

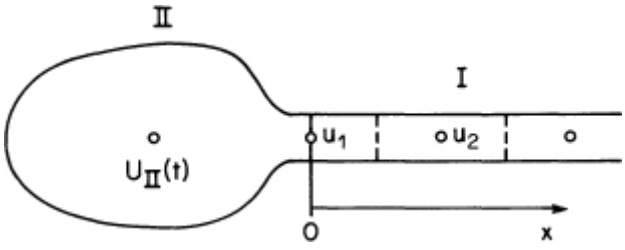


FIG. 3.10. Boundary conditions of the 3rd kind.

$$q_{II \to I}(t) = b[U_{II}(t) - u_1] \tag{3.105}$$

where b is a proportionality constant similar to the productivity index. On the other hand, inside reservoir I at $x \to 0$, the flow rate must satisfy Darcy's Law

$$q_{II \to I}(t) = a\frac{\partial U}{\partial x} \tag{3.106}$$

These two equations combined together give the boundary condition of the type (3.104)

$$a\frac{\partial U}{\partial x} + bU = bU_{II}(t)$$

Similar equations arise in the treatment of individual wells and aquifers (see Chapter 9). In the point-distributed grid, eqn. (3.104) can be approximated by

$$a[u_2^n - U_{II}^n]/2\Delta x + bu_1^n = bU_{II}^n \tag{3.107}$$

where we have used the reflection point.

In the block-centred grid, we encounter some difficulty in approximating bu at $x = 0$ and equations become more complex.

3.4.2.4 *Boundary Conditions of the* 4*th Kind*

These conditions, also called 'cyclic boundary' conditions, do not normally arise in one-dimensional problems. They are, however, important in multidimensional problems; we introduce them here for use in subsequent chapters. An example of a problem with these boundary conditions would be a reservoir in the form of a ring (Fig. 3.11), with x-co-ordinate along the circular centre-line. Then the point $x = 0$ is identical with the point $x = L$.

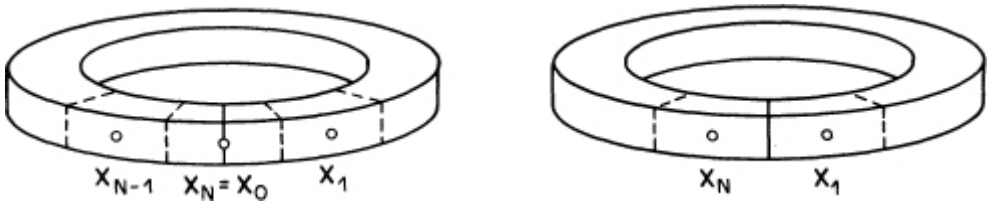


FIG. 3.11. Boundary conditions of the 4th kind. Left, point-distributed grid; right, block-centred grid.

The continuity of pressure and flow at $x = 0$ requires

$$U(L,t) = U(0,t) \tag{3.108a}$$

$$\frac{\partial U}{\partial x}(L,t) = \frac{\partial U}{\partial x}(0,t) \tag{3.108b}$$

For the point-distributed grid (Fig. 3.11, left), we specify

$$u_N = u_0 \tag{3.109}$$

Consequently, the difference equations for $i = 1$ and $i = N$ will be

$$\begin{aligned}(u_N - 2u_1 + u_2)/\Delta x^2 &= \frac{\partial u_1}{\partial t} + q_1 \\ (u_{N-1} - 2u_N + u_1)/\Delta x^2 &= \frac{\partial u_N}{\partial t} + q_N\end{aligned} \tag{3.110}$$

The boundary condition (3.108b) is not used, because $\partial U/\partial x$ is approximated at N - ½ and ½.

For the block-centred grid (Fig. 3.11, right), the condition (3.108a) is not relevant because there is no grid point at $x = 0$, and (3.108b) is satisfied if $\partial u/\partial x$ at $x = 0$ is approximated by the same term in the equations for points 1 and N. Then the two equations are:

$$\begin{aligned}(u_N - 2u_1 + u_2)/\Delta x^2 &= \frac{\partial u_1}{\partial t} + q_1 \\ (u_{N-1} - 2u_N + u_1)/\Delta x^2 &= \frac{\partial u_N}{\partial t} + q_N\end{aligned} \tag{3.111}$$

Cyclic boundary conditions also arise in two- and three-dimensional problems in cylindrical co-ordinates, discussed in later chapters.

3.5 DISCRETISATION OF ONE-DIMENSIONAL FLOW EQUATIONS IN CARTESIAN CO-ORDINATES

Let us first consider eqn. (3.3) for a slightly compressible fluid

$$\mathrm{A}\,U \equiv \frac{\partial}{\partial x}\left[\lambda(x, U)\frac{\partial U}{\partial x}\right] - c(x, U)\frac{\partial U}{\partial t} - q(x,t) \tag{3.112}$$

where

$$\lambda(x, U) = k(x)/B\mu(U), c(x, U) = \frac{\phi(x)c_f}{B^\circ} + \frac{\phi^\circ c_R}{B(U)}$$

There are two features that arise in solving eqn. (3.112), which have not been discussed in previous sections: (1) use of irregular grid, and (2) treatment of variable coefficients λ and c.

3.5.1 Difference Equations for Irregular Grid

The use of irregular grid spacing is essential in reservoir simulation. In many practical problems it is necessary to refine the grid in certain parts of the reservoir in order to obtain desired accuracy. For example, local refinement is necessary around the well in single-well (coning type) simulations. On the other hand, it is often possible to use coarser grid over areas such as aquifers and large gas caps where pressure and saturations change slowly. Irregular grid is also advantageous in cross-sectional and 3-D simulations of stratified reservoirs where the vertical grid is chosen according to reservoir stratification. In practice, we always want to keep the grid as coarse as possible (especially in 3-D simulations), and therefore the accuracy of finite-difference approximations is very important.

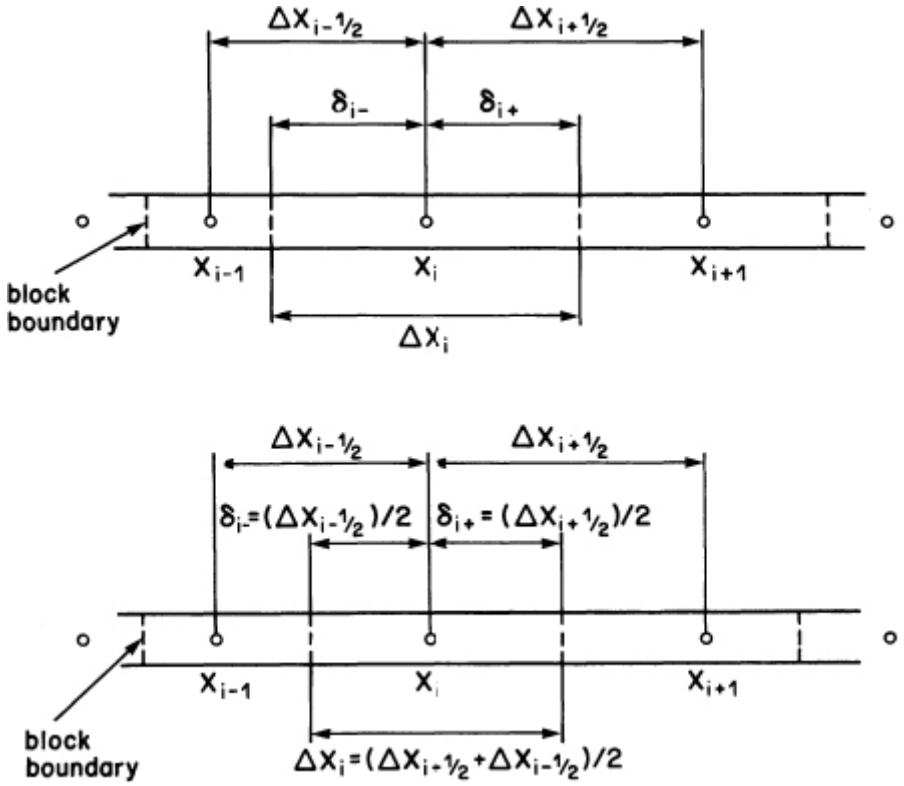


FIG. 3.12. Top, block-centred grid; bottom, point-distributed grid.

The use of irregular grid in reservoir simulation has been discussed in two papers by Settari and Aziz (1972, 1974). The related question of irregular boundaries has been discussed by, for example, Greenspan (1965) and Collatz (1966). However, it is only of minor importance in reservoir simulation because the physical boundaries are neither distinct nor accurately known.

Consider now difference approximations to the operator A over an irregular grid. Our discussion will closely follow Settari and Aziz (1972), where additional details may be found. The geometrical quantities involved in the construction of the block-centred and point-distributed grids are shown in Fig. 3.12. According to the principle outlined in Section 3.4.1, in the block-centred grid the block sizes Δx_i are selected first and then the grid points are centred between the boundaries, i.e.

$$\delta_{i-} = \delta_{i+} = \Delta x_i/2$$

Therefore $\Delta x_{i+1/2} = \frac{1}{2}(\Delta x_i + \Delta x_{i+1})$.

In the point-distributed grid the grid points are selected first and the block boundaries are placed half-way between the grid points:

$$\delta_{i+} = \delta_{i+1,-} = \Delta x_{i+1/2}/2$$
$$\delta_{i-} = \delta_{i-1,+} = \Delta x_{i-1/2}/2$$

The block sizes are then $\Delta x_i = \frac{1}{2}(\Delta x_{i+1/2} + \Delta x_{i-1/2}) = \delta_{i+} + \delta_{i-}$.

In addition to the definition of block boundaries we also need interblock transmissibilities $\lambda_{i+\frac{1}{2}}$. The problem of suitable definition of $\lambda_{i+\frac{1}{2}}$ will be dealt with in Section 3.5.3. As it has been shown by Settari and Aziz (1972), our conclusions here will not be affected by the treatment of transmissibilities.

In the case of the block-centred grid, the finite-difference approximation of A is

$$\tilde{\mathrm{L}}_1 u_i = \frac{1}{\Delta x_i}\left[\lambda_{i+1/2}\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}} + \lambda_{i-1/2}\frac{u_{i-1}-u_i}{\Delta x_{i-1/2}}\right] \tag{3.113}$$

If $\lambda_{i\pm\frac{1}{2}}$ is approximated as follows

$$\lambda_{i\pm1/2} = \tfrac{1}{2}(\lambda_i + \lambda_{i\pm1}) \tag{3.114}$$

the finite-difference operator in place of $\tilde{\mathrm{L}}_1$ is

$$L_1 u_i = \frac{1}{\Delta x_i}\left[\frac{\lambda_i}{2}\left(\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}} + \frac{u_{i-1}-u_i}{\Delta x_{i-1/2}}\right) + \frac{\lambda_{i+1}}{2}\left(\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}}\right) + \frac{\lambda_{i-1}}{2}\left(\frac{u_{i-1}-u_i}{\Delta x_{i-1/2}}\right)\right] \tag{3.115}$$

Similarly, for the point-distributed grid, the starting point is the operator

$$\tilde{L}_2 u_i = \frac{2}{\Delta x_{i+1/2} + \Delta x_{i-1/2}}\left[\lambda_{i+1/2}\left(\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}}\right) + \lambda_{i-1/2}\left(\frac{u_{i-1}-u_i}{\Delta x_{i-1/2}}\right)\right] \tag{3.116}$$

which changes for the approximation eqn. (3.114) to

$$L_2 u_i = \frac{2}{\Delta x_{i+1/2} + \Delta x_{i-1/2}}\left[\lambda_i\left(\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}} + \frac{u_{i-1}-u_i}{\Delta x_{i-1/2}}\right) + \frac{\Delta x_{i+1/2}}{2}\left(\frac{\lambda_{i+1}-\lambda_i}{\Delta x_{i+1/2}}\right)\left(\frac{u_{i+1}-u_i}{\Delta x_{i+1/2}}\right) + \frac{\Delta x_{i-1/2}}{2}\left(\frac{\lambda_i-\lambda_{i-1}}{\Delta x_{i-1/2}}\right)\left(\frac{u_i-u_{i-1}}{\Delta x_{i-1/2}}\right)\right] \tag{3.117}$$

The local truncation errors of the operators L_1 and L_2 are found to be

$$\begin{aligned} R_1(U_i) = &-\frac{\Delta x_{i+1} - 2\Delta x_i + \Delta x_{i-1}}{4\Delta x_i}(\lambda U')'_i \\ &-\frac{(\Delta x_{i+1/2})^2 - (\Delta x_{i-1/2})^2}{12\Delta x_i}[2\lambda U''' + 3(\lambda' U')']_i \\ &-\frac{(\Delta x_{i+1/2})^3 + (\Delta x_{i-1/2})^3}{24\Delta x_i}[\lambda U^{\mathrm{IV}} + 2\lambda' U''' + 3\lambda'' U'' + 2\lambda''' U']_i \\ &+ O(\Delta x^3) \end{aligned} \tag{3.118}$$

and

$$R_2(U_i) = -\frac{(\Delta x_{i+1/2} - \Delta x_{i-1/2})}{6}[2\lambda U''' + 3(\lambda' U'')']_i \tag{3.119}$$
$$-\frac{(\Delta x_{i+1/2}^3 + \Delta x_{i-1/2}^3)}{24\Delta x_i}[\lambda U^{IV} + 2\lambda' U''' + 3\lambda'' U'' + 2\lambda''' U']_i$$
$$+ O(\Delta x^3)$$

We observe that for the same position of grid points the error R_1 differs from R_2 by the additional term:

$$\frac{\Delta x_{i+1} - 2\Delta x_i + \Delta x_{i-1}}{4\Delta x_i}(\lambda U')_i' \tag{3.120}$$

This term is generally of zero order, which implies (in the sense of the definition introduced in Section 3.2.3), that L_1 is an *inconsistent* approximation. In contrast to this, L_2 is always at least of first order and therefore *consistent*. Note also that for regular spacing both approximations become $O(\Delta x^2)$ and for λ equal to a constant they reduce to the familiar form (eqn. (3.15)) discussed previously. On the other hand, *making the problem linear* (λ = *constant*) *with irregular grid does not improve the order of the approximation*. We note here in passing that for the nonlinear case the manner of approximating $\lambda_{i \pm 1/2}$ may lower the order of operator **L** when some approximation different from eqn. (3.114) is used (see Section 3.5.3). The essential fact is that the additional error terms introduced due to the approximation of λ are identical for both operators L_1 and L_2 (Appendix B of Settari and Aziz, 1972).

Let us now explore the question: 'What is the significance of the truncation error analysis just presented?' The point-distributed approximation L_2 is consistent and therefore for any stable approximation of the right side of eqn. (3.112) it will be *convergent*. However, there is no *a priori* guarantee of convergence for the block-centred operator L_1. It should be emphasised that inconsistency does not necessarily imply divergence as $\|\Delta x\| \to 0$. The actual order of convergence depends on the manner in which the grid spacing is refined. For example, one can construct a sequence of spacings, that will make the term (3.120) identically zero; such spacings satisfy the discrete Laplacian:

$$\Delta x_{i+1} - 2\Delta x_i + \Delta x_{i-1} = 0$$

Other spacings may be constructed, for which the term (3.120) will actually be $O(\Delta x)$ or $O(\Delta x^2)$. On the other hand, spacings may also be constructed for which this term

remains of $O(\Delta x^{\circ})$, but only at a finite number of points. Because the discretisation errors act as sources in the equation for errors of solution (see Section 3.2.3), their effect is 'smoothed out'. Based on numerical experiments, we conclude that for any reasonably smooth variation of spacings the operator L_1 will also be convergent *in the limit.*

Although the effect of the zero-order term vanishes in the limit this term may cause significant errors when grid sizes are finite. We will show this for the case of nonlinear gas flow in a horizontal, incompressible porous medium:

$$\frac{\partial}{\partial x}\left(\lambda(p)\frac{\partial p}{\partial x}\right)+\frac{\mathrm{B}T}{\mathrm{M}}S(t,x)=c(p)\frac{\partial p}{\partial t}\qquad x\varepsilon(0,L) \tag{3.121}$$

where

$$\lambda=\frac{kp}{Z\mu}\qquad c=\frac{\phi\mathrm{A}}{Z}\left(1-\frac{p}{Z}\frac{\mathrm{d}Z}{\mathrm{d}p}\right)$$

In the above equation T represents temperature, M is the molecular weight of gas, and A and B are conversion constants. Culham and Varga (1971) presented a special form of the source term $S(t, x)$, for which eqn. (3.121) has an exact solution with a sharp pressure peak near $x = L$. This problem was solved numerically with operators L_1 and L_2 using a spacing refined around this peak. The time truncation error was suppressed by a choice of sufficiently small Δt, the effect of boundaries was eliminated and the l_∞ - norm of errors was evaluated as a function of time. The results are shown in Fig. 3.13 for two grids of 25 and 50 nodes.

Both operators are convergent, but the block-centred operator has much larger errors. Also, the errors exhibit different behaviour in time: errors with L_1 increase while those with L_2 decrease with increasing time.

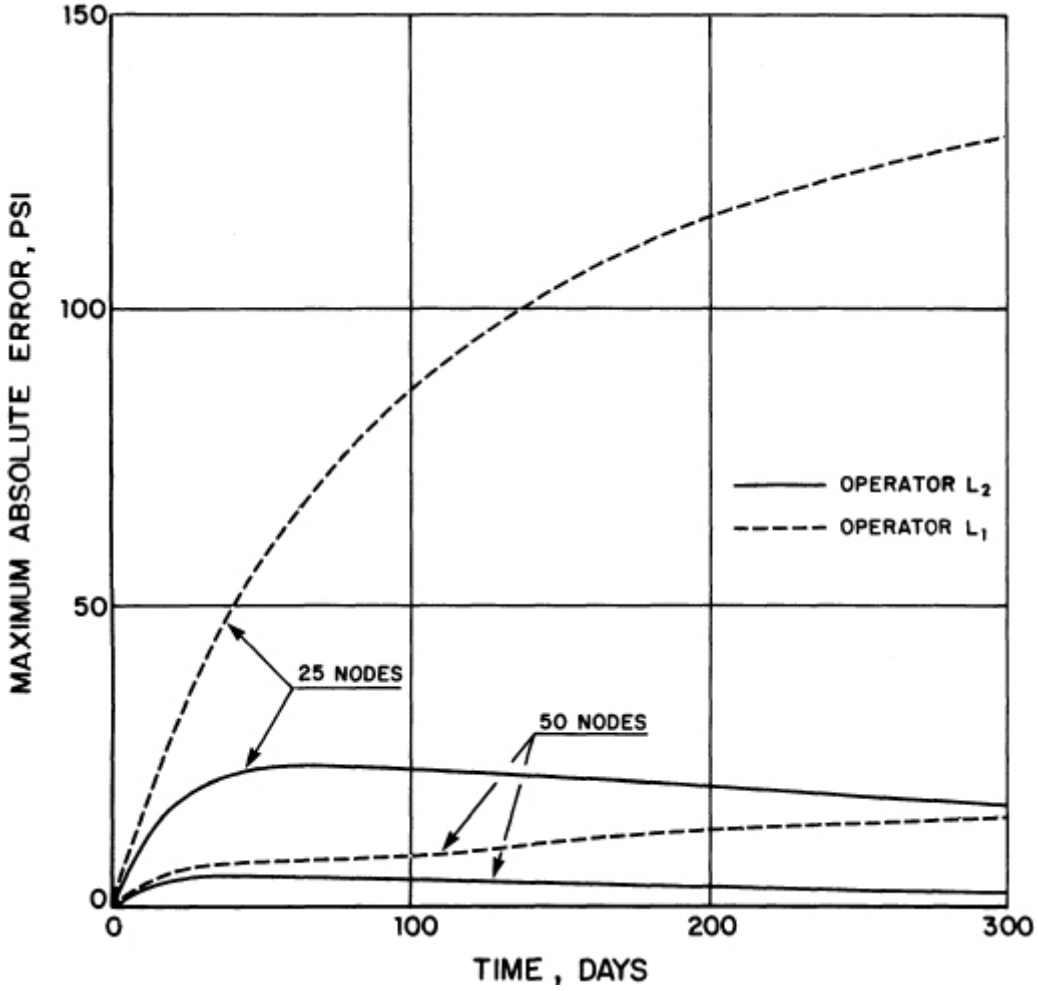


FIG. 3.13. Effects of grid refinement on error obtained with operators L_1 and L_2.

The results displayed in Fig. 3.13 are significant because in practice discretisation errors are rarely determined by mesh-refinement experiments. In the above example, L_1 would require, say, 40 blocks while 25 would be enough for L_2. In practice it is likely that 25 blocks would also be used for L_1, resulting in large errors for this example.

In conclusion, the use of the point-distributed grid as opposed to the block-centred grid is recommended. Since the only difference between the two schemes is in the location of block boundaries, the recommended scheme is no harder to use. In addition, it offers definite advantages for the treatment of boundary conditions, in particular for single-well problems (Chapter 9).

3.5.2 Difference Equations in Matrix Form

The left side of eqn. (3.112) may be discretised and the resulting difference equations may be written in a matrix form:

$$-\mathbf{T}\boldsymbol{u} = \mathbf{B}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \mathbf{Q} \tag{3.122}$$

where

$$\boldsymbol{u} = (u_1, u_2, \ldots, u_N)^{\mathrm{T}}$$

according to the convention discussed earlier. If we use directly equations of the form

$$\mathrm{L}u_i = c_i \,\mathrm{d}u_i/\mathrm{d}t + q_i \qquad i = 1, \ldots, N \tag{3.123}$$

then the matrix **T** will not be symmetric (except for a uniform grid), because the $i, j+1$ element is not the same as the $i+1, j$ element

$$\frac{\lambda_{i+1/2}}{\Delta x_i \, \Delta x_{i+1/2}} \neq \frac{\lambda_{i+1/2}}{\Delta x_{i+1} \, \Delta x_{i+1/2}}$$

A symmetric form is desirable for theoretical as well as practical reasons. This is achieved by multiplying the ith equation by Δx_i, leading to the following form

$$\frac{\lambda_{i+1/2}}{\Delta x_{i+1/2}}(u_{i+1} - u_i) + \frac{\lambda_{i-1/2}}{\Delta x_{i-1/2}}(u_{i-1} - u_i) = \Delta x_i c_i \,\mathrm{d}u_i/\mathrm{d}t + \Delta x_i q_i \tag{3.124}$$

The transition from eqn. (3.123) to (3.124) has also an important physical meaning: while the left side of eqn. (3.123) represents derivative of fluxes (i.e., derivative of flow rates per unit area), the corresponding terms in eqn. (124) represent flow rates in and out of block i. Likewise, the right side of eqn. (3.123) is the rate of change of mass in a unit volume and the right side of eqn. (3.124) is the rate of change of mass in the *volume* of the block i. This becomes obvious when we realise that the differential equation can be multiplied by an arbitrary cross-sectional area $A = \Delta y \, \Delta z$. Therefore we can write eqn. (3.124) as

$$\begin{aligned} q_{i+1/2} - q_{i-1/2} &= \mathrm{T}_{i+1/2}(u_{i+1} - u_i) - \mathrm{T}_{i-1/2}(u_i - u_{i-1}) \\ &= V_i c_i \frac{\mathrm{d}u_i}{\mathrm{d}t} + Q_i \end{aligned} \tag{3.125}$$

where

$$\mathrm{T}_{i+1/2} = \frac{\lambda_{i+1/2} A}{\Delta x_{i+1/2}} \tag{3.126}$$

$$V_i = \Delta x_i A$$

are the discrete transmissibility between i and $i + 1$ and volume of the block i, respectively. It is possible to partially account for multidimensional effects by allowing area A to be a function of x. This modification of eqn. (3.125) is the usual form used in reservoir simulation.

It is worth noting that eqn. (3.125) is often derived directly from elementary mass conservation considerations for the block i:

$$(Flow\text{-}in) - (Flow\ out) = Gain$$

Such an approach is closer to engineer's thinking in terms of 'material balances' and it is linked with the mathematician's approach via the integral method of derivation of difference equations (Section 3.2.1).

Consider now the boundary conditions at $x = 0$. The most common case of no-flow boundary can be treated by the reflection technique discussed in Section 3.3.4 and shown in Fig. 3.9. Let us write eqn. (3.125) for $i = 1$. In the point-distributed grid the 'reflection technique' requires that $u_0 = u_2$, $\mathrm{T}_{1/2} = \mathrm{T}_{1+1/2}$ (Fig. 3.9, left). Therefore we obtain

$$2\mathrm{T}_{1+1/2}(u_2 - u_1) = \Delta x_1 A c_1 \frac{\mathrm{d}u_1}{\mathrm{d}t} + \Delta x_1 A q_1$$

and after dividing by 2

$$\mathrm{T}_{1+1/2}(u_2 - u_1) = V_1 c_1 \frac{\mathrm{d}u_1}{\mathrm{d}t} + V_1 q_1 \tag{3.127}$$

where $V_1 = (\Delta x_1/2)A$ is the correct volume of the first block.

On the other hand, for the block-centred grid the condition is $u_0 = u_1$ (which is equivalent to $\mathrm{T}_{1/2} = 0$), again giving eqn. (3.127), but this time with $V_1 = \Delta x_1 A$.

Assuming no-flow boundary conditions at $x = 0$ and $x = L$, the matrices **T** and **B** in eqn. (3.122) will be

$$\mathbf{T} = \begin{bmatrix} T_{3/2} & -T_{3/2} & & & & \\ -T_{3/2} & (T_{3/2}+T_{5/2}) & -T_{5/2} & & & \\ & \cdot & \cdot & \cdot & & \\ & & -T_{i-1/2} & (T_{i-1/2}+T_{i+1/2}) & -T_{i+1/2} & \\ & & & \cdot & \cdot & \cdot \\ & & & & -T_{N-1/2} & T_{N-1/2} \end{bmatrix} \tag{3.128}$$

$$\mathbf{B} = \begin{bmatrix} V_1c_1 & & & & & \\ & V_2c_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & V_ic_i & \\ & & & & & \cdot \\ & & & & & & \cdot \\ & & & & & & & V_Nc_N \end{bmatrix} \tag{3.129}$$

and the vector $\boldsymbol{Q}$ will be of the form

$$\boldsymbol{Q} = [Q_1, \ldots, Q_i, \ldots, Q_N]^{\mathrm{T}} \tag{3.130}$$

We note that matrix **T** is a tridiagonal, symmetric matrix.

3.5.3 Treatment of Variable Coefficients

There is no unique way of choosing the values $\lambda_{i+½}$, c_i and q_i. In general, they should be chosen such that they give the most accurate values possible for the flow rate (accumulation and influx into the block). However, sometimes (as in the case of multiphase flow) their choice is dictated by the numerical techniques used. Here we will derive some useful formulae for simple cases.

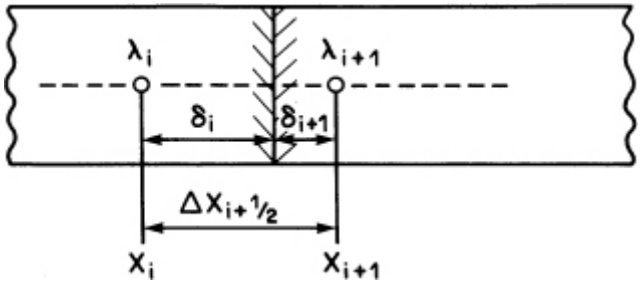


FIG. 3.14. Averaging of transmissibility—vertical discontinuity.

Suppose that the transmissibility is piecewise constant with interface between i and $i + 1$ (not necessarily at the block boundary, Fig. 3.14). The flow rate between i and $i + 1$ is

$$-q = A\frac{(p_{\text{int}} - p_i)}{\delta_i}\lambda_i = A\frac{(p_{i+1} - p_{\text{int}})}{\delta_{i+1}}\lambda_{i+1}$$

where p_{int} is the pressure at the interface. We want to find a mean transmissibility $\lambda_{i+\frac{1}{2}}$ which will give the same flow rate between i and $i+ 1$:

$$-q = A\frac{(p_{i+1} - p_i)}{\Delta x_{i+1/2}}\lambda_{i+1/2} \tag{3.131}$$

Elimination of p_{int} from these equations yields

$$\lambda_{i+1/2} = \frac{(\delta_i + \delta_{i+1})}{\dfrac{\delta_{i+1}}{\lambda_{i+1}} + \dfrac{\delta_i}{\lambda_i}} \tag{3.132}$$

Therefore, $\lambda_{i+\frac{1}{2}}$ is the *harmonic mean* value of λ_i and λ_{i+1}.

Consider now the case when the reservoir is composed of two layers of different permeability as shown on Fig. 3.15. The total flow between x_i and $x_{\text{i+i}}$ is

$$-q = -(q_1 + q_2) = A\frac{\delta_1}{\delta}\frac{(p_{i+1} - p_i)}{\Delta x_{i+1/2}}\lambda_1 + A\frac{\delta_2}{\delta}\frac{(p_{i+1} - p_i)}{\Delta x_{i+1/2}}\lambda_2$$

Comparison of the above equation with eqn. (3.131) gives

$$\lambda_{i+1/2} = \left(\frac{\delta_1\lambda_1 + \delta_2\lambda_2}{\delta_1 + \delta_2}\right) \tag{3.133}$$

i.e., in this case $\lambda_{i+\frac{1}{2}}$ is a weighted *arithmetic mean* of λ_1 and λ_2.

The analogies of eqns. (3.132) and (3.133) to problems in heat conduction (heat conductivity in composite walls) and electricity (series and parallel resistors) are well known.

In practice, the value of absolute permeability k is usually assigned to the centre of the block and the pressure dependent properties are evaluated from the block pressure p_i. The transmissibilities λ_i are then assumed to be constant within all blocks, and they are used to compute $\lambda_{i\pm\frac{1}{2}}$. Such an approximation is satisfactory if the properties do not change too much between two adjacent grid blocks. When the contrast in permeability between blocks is large and the data uncertain, large errors may result. The representative value of transmissibility should then reflect variations of k between grid points based on statistical interpretation of data (Toronyi and Farouq Ali, 1974).

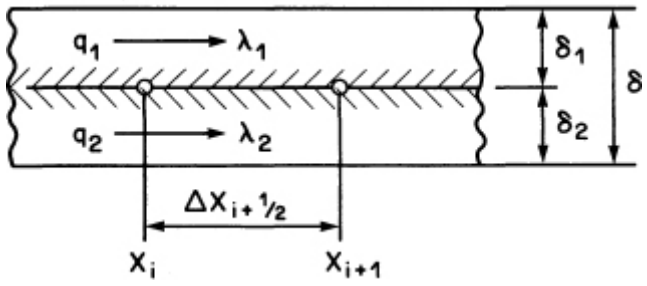


FIG. 3.15. Averaging of transmissibility—layered system.

Treatment of $c(x)$ and $q(x, t)$ poses similar problems. The value of c_i and q_i should be chosen such that they are integral mean values:

$$c_i V_i = \int_{\Delta x_i} Ac(x)\,\mathrm{d}x \qquad q_i V_i = \int_{\Delta x_i} Aq(x)\,\mathrm{d}x$$

Since the compressibilities c_f, c_R were assumed constant, we can write $c_\mathrm{i} = \phi_\mathrm{i} c_\mathrm{f}/B^\mathrm{o} + \phi^\mathrm{o} c_R/B$ where ϕ_i is the representative porosity of the block i:

$$\phi_i = \frac{1}{V_i}\int_{V_i} \phi\,\mathrm{d}V \tag{3.134}$$

The data required for the use of eqn. (3.134) are rarely known. The situation is better with the production term q_i. The sources and sinks usually represent injection and production wells and as such can be better approximated by point sources (Dirac δ-functions) rather than distributed sources (Fig. 3.16).

As long as the entire non-zero source function lies within one grid block, the value of the total source strength

$$Q_i = q_i V_i = \int_{V_i} q \, \mathrm{d}V$$

is independent of the actual distribution of the source within the block. Since only the total rate Q_i is required in the difference equations, the function $q(x)$ is immaterial. This fact also shows the limit of 'resolution' of the finite-difference method: *the location of a well can be changed within one grid block without affecting the answers.*

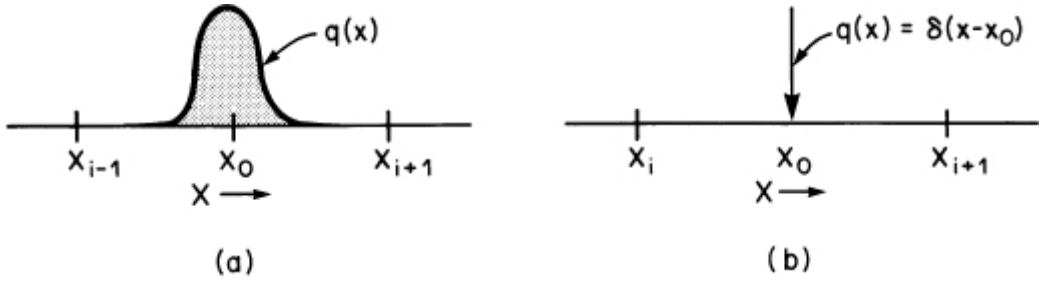


FIG. 3.16. Representation of source terms.

3.6 DISCRETISATION OF ONE-DIMENSIONAL FLOW EQUATIONS IN RADIAL CYLINDRICAL CO-ORDINATES

The equivalent of eqn. (3.112) in radial co-ordinate of cylindrical geometry is (eqn. (3.4)):

$$\mathrm{C}U \equiv \frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda(r, U)\frac{\partial U}{\partial r}\right) - c(r)\frac{\partial U}{\partial t} - q(r, t) \tag{3.135}$$

This equation can be transformed into a form similar to eqn. (3.112) by a transformation of the space co-ordinate. Define $\rho = \ln r$; then $r\lambda\partial U/\partial r = \lambda\partial U/\partial\rho$ and after multiplication by r^2 eqn. (3.135) is transformed into

$$\mathrm{A}U \equiv \frac{\partial}{\partial\rho}\left(\lambda(\exp(\rho), U)\frac{\partial U}{\partial\rho}\right) - \exp(2\rho)\left(c(\exp(\rho))\frac{\partial U}{\partial t} - q(\exp(\rho), t)\right) \tag{3.136}$$

since $r = \ln^{-1}\rho = \exp(\rho)$.

Either of the above equations describes flow in a reservoir towards an isolated well, under the assumption that all reservoir properties as well as boundary conditions are symmetric with respect to the axis of the well. The radius r_w is the radius of the well and r_e is the external (drainage) radius (Fig. 3.17). Analytical solutions for simple cases are well known (Craft and Hawkins, 1959, ERCB, 1975). They have logarithmic character with pressure gradient rapidly increasing towards the wellbore.

When eqn. (3.135) is solved numerically, smaller and smaller grid increments are necessary as $r \rightarrow r_w$ in order to maintain uniform accuracy. This results in highly irregular grid. (The ratio of largest to smallest Δr is typically of the order of 10^2.)

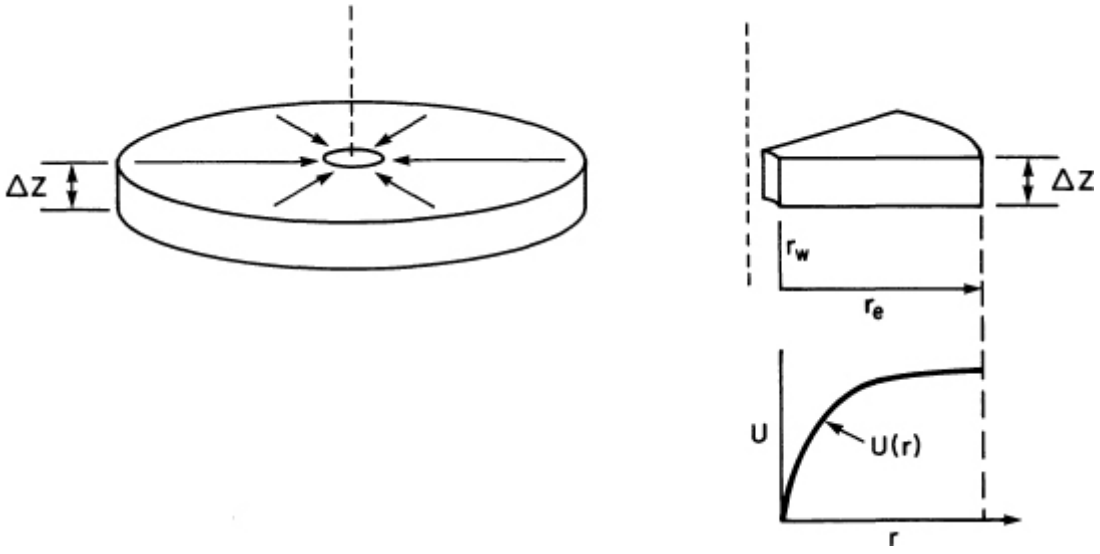


FIG. 3.17. One-dimensional radial flow.

If we choose to solve instead eqn. (3.136), equal spacing in ρ is usually most suitable. Since for steady-state Darcy flow in a homogeneous medium

$$U - U_w \sim \ln (r/r_w) \sim \rho - \rho_w$$

equal spacing in ρ will result in equal pressure drop between grid points. This reasoning is commonly used for constructing irregular grid in r. Suppose that we employ point-distributed grid of N grid points; then $r_1 = r_w$, $r_N = r_e$. The grid spacing should be regular in ρ, therefore

$$\Delta\rho = (\rho_e - \rho_w)/(N - 1) = \text{constant}$$

Since $\Delta\rho = \ln (r_{i+1}/r_i)$, we get

$$\frac{r_{i+1}}{r_i} = \left(\frac{r_e}{r_w}\right)^{1/(N-1)} \qquad i = 1, 2, \ldots, N \tag{3.137}$$

i.e., the co-ordinates of grid points increase in geometric progression.

Note that although the formulation (3.136) transforms the left-hand side into an operator similar to the one describing linear flow, the radial character of the equation has been transferred to the right-hand side (accumulation terms). Consequently, the transformed equation does not offer any significant advantage for numerical simulation. Most two- and three-dimensional simulators in radial co-ordinates use r co-ordinate directly (MacDonald and Coats, 1970, Letkeman and Ridings, 1970; Nolen and Berry, 1972; Sonier *et al*., 1973; Settari and Aziz, 1974a), although ρ has been employed by Bruce *et al*. (1953) and Akbar *et al.* (1974).

3.6.1 Difference Equations for Irregular Grid

The treatment of irregular grid for eqn. (3.135) is similar to the linear case discussed in Section 3.5.1. The operator C can be written as (1/x)A, if we define coefficient $\lambda(x,U) = x\lambda(x,U)$. Therefore, we can at once conclude that the results of the analysis in Section 3.5.1 can be applied directly to the operator C; and as before, the point-distributed grid is always consistent while the block-centred grid is only consistent under special conditions (i.e., inconsistent in general). All consistent approximations for C U may be written as:

$$\mathbf{M}u_i = \frac{1}{x_i \Delta x_i}\left[\lambda_{i+1/2} x_{i+1/2}\left(\frac{u_{i+1} - u_i}{\Delta x_{i+1/2}}\right) + \lambda_{i-1/2} x_{i-1/2}\left(\frac{u_{i-1} - u_i}{\Delta x_{i-1/2}}\right)\right] \tag{3.138}$$

where

$$x_{i+1/2} = \mathrm{f}(x_i, x_{i+1}) \tag{3.139}$$

and f represents the averaging function.

However, when $x = r$, i.e., x is a radial co-ordinate, it is not clear that the arithmetic mean is the best choice for locating the boundaries between grid points. As discussed at length by Settari and Aziz (1974), several choices will produce consistent approximations:

(a) *Arithmetic mean*

$$r_{i+1/2} = \tfrac{1}{2}(r_i + r_{i+1}) \tag{3.140}$$

(b) *Geometric mean*

This choice results when the arithmetic mean is applied to the transformed equation (eqn. (3.136)):

$$r_{i+1/2} = (r_i r_{i+1})^{1/2} \tag{3.141}$$

(c) *Logarithmic mean*

From now on it is necessary for us to utilise certain physical concepts; for this reason we will often use the variable p, which represents pressure, in place of u.

In a steady-state flow with λ equal to a constant, the exact pressure drop between i and $i + 1$ is obtained by integrating Darcy's Law

$$q = -2\pi r \lambda \frac{\mathrm{d}p}{\mathrm{d}r}$$

which results in

$$q_{\mathrm{E}} = -\frac{\lambda 2\pi\,\Delta z}{\ln\,(r_{i+1}/r_i)}(p_{i+1} - p_i) \tag{3.142}$$

The discretised Darcy velocity $q_{i+½}$ is just the first term inside the brackets in eqn. (3.138), multiplied by $2\pi\Delta z$:

$$-q_{i+1/2} = 2\pi\Delta z r_{i+1/2}\lambda_{i+1/2}\left(\frac{p_{i+1} - p_i}{r_{i+1} - r_i}\right) \tag{3.143}$$

Equation (3.143) gives the exact pressure drop for λ equal to a constant, provided

$$r_{i+1/2} = r^{\mathrm{L}}_{i+1/2} = \frac{r_{i+1} - r_i}{\ln\,(r_{i+1}/r_i)} \tag{3.144}$$

which is the well-known 'logarithmic mean radius'.

Numerical analysis alone does not provide any further basis for the selection of block boundaries. Such a basis can be obtained by considering some desirable approximation properties for the difference equations approximating eqn. (3.135). Let us utilise eqn. (3.138) in eqn. (3.135) to obtain:

$$\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left[\lambda_{i+1/2}r_{i+1/2}\left(\frac{p_{i+1}-p_i}{r_{i+1}-r_i}\right)+\lambda_{i-1/2}r_{i-1/2}\left(\frac{p_{i-1}-p_i}{r_i-r_{i-1}}\right)\right] \quad (3.145)$$

$$= c_i\frac{\mathrm{d}p_i}{\mathrm{d}t}+q_i$$

To achieve a symmetric form and give the terms in the equation additional physical meaning, we multiply the equation by $\Delta z 2\pi r_i(r_{i+1/2}-r_{i-1/2})$ and utilise eqn. (3.143) to obtain

$$-(q_{i+1/2}-q_{i-1/2}) = C_{v_i}c_i\frac{\mathrm{d}p_i}{\mathrm{d}t}+C_{v_i}q_i \quad (3.146)$$

where

$$C_{v_i} = 2\pi r_i(r_{i+1/2}-r_{i-1/2})\Delta z \quad (3.147)$$

In eqn. (3.146) the $q_{i\pm1/2}$ are the flows in and out of the block i (which is a circular ring) and C_{Vi} is the discretised volume of the block.

We may now require that:

(a) the discretised equation gives exact flow rate for a given pressure drop for λ = constant, and
(b) the discretised volume be equal to the actual volume of the block, $V_i = \pi\left(r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}}\right)\Delta z$.

These two conditions can be written as:

$$q_{i+1/2} = q_E\Delta z \quad C_{vi} = V_i \quad (3.148)$$

and will be satisfied if

$$r_{i+1/2} = r^L_{i+1/2} \qquad 2r_i(r_{i+1/2}-r_{i-1/2}) = r^2_{i+1/2}-r^2_{i-1/2} \quad (3.149)$$

Unfortunately, relations (3.148) cannot be satisfied simultaneously by any choice of boundaries. Nevertheless, as discussed below, a different discretisation may be obtained that satisfies these conditions.

Let us denote $\rho = r^2$, then eqn. (3.135) is transformed to

$$4\frac{\partial}{\partial\rho}\left(\rho\lambda\frac{\partial p}{\partial\rho}\right)=c\frac{\partial p}{\partial t}+q$$

which can be approximated by the following consistent scheme:

$$\frac{4}{(\rho_{i+1/2}-\rho_{i-1/2})}\left[\lambda_{i+1/2}\rho_{i+1/2}\frac{p_{i+1}-p_i}{\Delta\rho_{i+1/2}}+\lambda_{i-1/2}\rho_{i-1/2}\frac{p_{i-1}-p_i}{\Delta\rho_{i-1/2}}\right] \tag{3.150}$$

$$=c_i\frac{\mathrm{d}p_i}{\mathrm{d}t}+q_i$$

After multiplication by

$$C_{vi}=\pi(\rho_{i+1/2}-\rho_{i-1/2})\Delta z \tag{3.151}$$

we obtain again eqn. (3.146), but with $q_{i\pm1/2}$ defined as

$$q_{i+1/2}=4\pi\Delta z\lambda_{i+1/2}\rho_{i+1/2}\left(\frac{p_{i+1}-p_i}{\rho_{i+1}-\rho_i}\right) \tag{3.152}$$

The conditions (3.148) are now equivalent to

$$\rho_{i+1/2}-\rho_{i-1/2}=r^2_{i+1/2}-r^2_{i-1/2}$$

$$2\frac{\rho_{i+1/2}}{\rho_{i+1}-\rho_i}=r^{\mathrm{L}}_{i+1/2}\frac{1}{(r_{i+1}-r_i)}$$

The first condition is always trivially satisfied and the second can be used to solve for $\rho_{i+1/2}$:

$$\rho_{i+1/2}=\frac{\rho_{i+1}-\rho_i}{\ln\left(\frac{\rho_{i+1}}{\rho_i}\right)}=\rho^{\mathrm{L}}_{i+1/2} \tag{3.153}$$

Therefore, the boundary must be logarithmic in r^2 rather than in r when the discretisation (3.150) is used. Note that the use of eqn. (3.153) reduces eqn. (3.152) to eqn. (3.143) with $r_{i+1/2}=r^L_{i+1/2}$. In other words, *the boundary for calculating interblock transmissibility is logarithmic in r but the boundary for calculating block volume is logarithmic in* r^2. It should be noted that the difference between the

discretisation (3.150) and (3.145) is only in the definition of C_v and will therefore be only apparent in unsteady flow problems.

Alternatively, it is possible to use the discretisation (3.143) and replace C_v by V. Such difference equations are not derivable by Taylor series method, but can be related to the integral method.

3.6.2 Difference Equations in Matrix Form

The eqn. (3.146) can be written in the form of eqn. (3.125):

$$T_{i-1/2}(p_{i-1} - p_i) + T_{i+1/2}(p_{i+1} - p_i) = V_i c_i \frac{dp_i}{dt} + V_i q_i$$

With the choice of the discretisation (3.150), we will have

$$T_{i+1/2} \equiv 2\pi r^L_{i+1/2} \frac{\Delta z}{r_{i+1} - r_i} \lambda_{i+1/2} \tag{3.154}$$

$$V_i = \pi(r^2_{i+1/2} - r^2_{i-1/2})\,\Delta z \tag{3.155}$$

where

$$r^2_{i+1/2} = \frac{r^2_{i+1} - r^2_i}{\ln\left(\frac{r_{i+1}}{r_i}\right)^2} \tag{3.156}$$

$$r^L_{i+1/2} = \frac{r_{i+1} - r_i}{\ln\left(\frac{r_{i+1}}{r_i}\right)} \tag{3.157}$$

When written in matrix form, we obtain

$$\mathbf{T}\boldsymbol{p} = \mathbf{B}\frac{d\boldsymbol{p}}{dt} + \boldsymbol{Q}$$

where the matrix **T** will again be *symmetric*.

It is interesting to note here that one may wish to use a slightly different starting point in the derivation of difference equations, discussed above, namely to expand eqn. (3.135) as

$$CU \equiv \frac{\partial}{\partial r}\left(\lambda(r, U)\frac{\partial U}{\partial r}\right) + \frac{1}{r}\lambda(r, U)\frac{\partial U}{\partial r} = c(r)\frac{\partial U}{\partial t} + q(r, t) \tag{3.158}$$

and then use a finite-difference approximation. Unfortunately, this approach produces *nonsymmetric* difference equations (see Varga, 1962, p. 193 where he mentions another means of obtaining the symmetric form by integration).

3.6.3 Treatment of Variable Coefficients

All remarks made in Section 3.5.3 can be repeated here, with the obvious modification for cylindrical geometry. For example, the porosity ϕ_i can be defined by

$$\phi_i = \frac{\Delta z}{V_i} 2\pi \int_{r_{i-1/2}}^{r_{i+1/2}} \phi(r) r \, dr \tag{3.159}$$

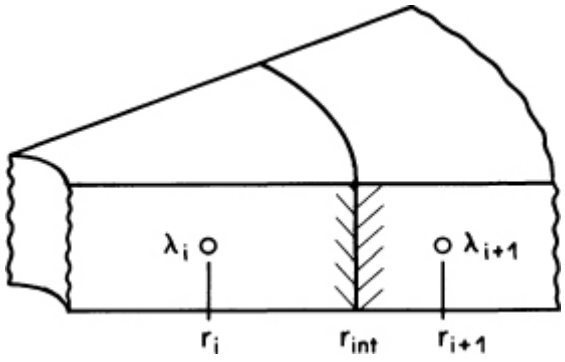


FIG. 3.18. Averaging of transmissibilities—vertical discontinuity.

The formulae for weighting of λ corresponding to eqn. (3.132) and eqn. (3.133) are:

(a) *For vertical interface* (Fig. 3.18)

$$\lambda_{i+1/2} = \frac{\ln(r_{i+1}/r_i)}{\dfrac{\ln(r_{int}/r_i)}{\lambda_i} + \dfrac{\ln(r_{i+1}/r_{int})}{\lambda_{i+1}}} \tag{3.160}$$

(b) *For horizontal interface* (*layered transmissibility* (Fig. 3.19))

$$\lambda_{i+1/2} = \frac{\delta_1}{\Delta z}\lambda_1 + \frac{\delta_2}{\Delta z}\lambda_2 \tag{3.161}$$

which is the same as eqn. (3.133).

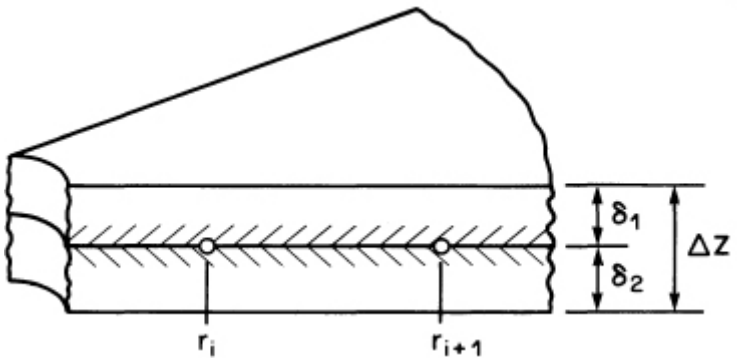


FIG. 3.19. Averaging of transmissibilities—layered system.

3.7 SOME PROPERTIES OF FINITE-DIFFERENCE EQUATIONS

3.7.1 Existence of Solution and Material Balance

The matrix equation for both linear and radial flow has the common form

$$-\mathbf{T}\boldsymbol{u} = \mathbf{B}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{Q} \tag{3.162}$$

where **T** is a symmetric tridiagonal matrix with positive diagonal elements and **B** is a diagonal matrix with positive entries. For Neumann boundary conditions **T**, **B** and $\boldsymbol{Q}$ have been given in Section 3.5.2. For Dirichlet boundary conditions only the equations for the boundary points will be modified. For example, consider the condition $u_{N+1} = u_{\mathrm{L}}$ at $x = L$. Then the matrix **T** will be

$$\mathbf{T} = \begin{bmatrix} \mathrm{T}_{3/2} & -\mathrm{T}_{3/2} & & & & \\ -\mathrm{T}_{3/2} & (\mathrm{T}_{3/2} + \mathrm{T}_{5/2}) & -\mathrm{T}_{5/2} & & & \\ & \cdot & \cdot & & & \\ & & -\mathrm{T}_{i-1/2} & (\mathrm{T}_{i-1/2} + \mathrm{T}_{i+1/2}) & -\mathrm{T}_{i+1/2} & \\ & & & \cdot & \cdot & \cdot \\ & & & & -\mathrm{T}_{N-1/2} & (\mathrm{T}_{N-1/2} + \mathrm{T}_{N+1/2}) \end{bmatrix} \tag{3.163}$$

and the vector $\boldsymbol{Q}$ will include the boundary term

$$\boldsymbol{Q} = \begin{bmatrix} Q_1 \\ \vdots \\ Q_i \\ \vdots \\ Q_N - T_{N+1/2} u_L \end{bmatrix} \tag{3.164}$$

where $Q_i = V_i q_i$ is the total source rate for the block i.

We observe that for every row of matrix **T** the diagonal entry is equal to the sum of the absolute values of the off-diagonal entries, except for the last row in which the diagonal entry is larger. Also, it is easy to see that **T** is an irreducible matrix (see Appendix A). According to the definition of diagonal dominance, we have:

(a) **T** *is diagonally dominant with zero row sums for Neumann boundary conditions.*
(b) **T** *is irreducibly diagonally dominant if at least one boundary condition is of Dirichlet type.*

Diagonal dominance is an important property; it follows immediately that **T** is *singular* in case (a) and *positive definite* in case (b) (Taussky, 1949; Varga, 1962, Chapter 1.5; Appendix A).

3.7.1.1 *Elliptic Problems*

We can immediately apply the above result to steady-state or incompressible problems, for which eqn. (3.162) reduces to $-\mathbf{T}\boldsymbol{u} = \mathbf{Q}$.

(a) *Neumann problem.* Since T is singular, solution may or may not exist. Let us sum the equations keeping in mind that **T** and $\boldsymbol{Q}$ are for this case given by eqn. (3.128) and (3.130). All flow terms $T_{i+1\,2}(u_{i+1} - u_i)$ will cancel in the process and before adding the last equation we obtain

$$T_{N-1/2}(u_{N-1} - u_N) = \sum_{i=1}^{N-1} Q_i \tag{3.165a}$$

and the last equation is

$$-T_{N-1/2}(u_{N-1} - u_N) = Q_N \tag{3.165b}$$

These two equations can be satisfied if and only if

$$\sum_{i=1}^{N} Q_i = 0 \tag{3.166}$$

We can then choose u_N arbitrarily, which will then determine $u_{(N-1)},\ldots,u_1$ uniquely.

The condition (3.166) can be looked upon as the *material balance condition.* The term 'material balance' is an engineering expression for conservation of mass of the entire system which is obtained by applying Green's theorem to the conservation equation. For Neumann problems, this is also the existence condition for the solution of differential equations (Mikhlin and Smolitskiy, 1967). In a closed incompressible system with no flow across the boundaries, the conservation of mass requires that the production be balanced by injection ($\int q\ dx = 0$), which in finite-difference form is exactly (3.166). The non-uniqueness of the solution is the consequence of the assumption of incompressibility: *because none of the properties are pressure-dependent, the pressure level is immaterial.*

(b) *Dirichlet boundary conditions*. In this case a unique solution exists. We can again derive a material balance equation which must be satisfied by the solution. By summing finite-difference equations for all points (where $\mathbf{T}$ and $\boldsymbol{q}$ are now given by eqns. (3.163) and (3.164)), we obtain

$$\mathrm{T}_{N+1/2}(u_L - u_N) + \sum_{i=1}^{N} Q_i = 0 \tag{3.167}$$

Obviously, $\mathbf{T}_{\mathrm{N+½}}(\boldsymbol{u_L} - u_N) = \boldsymbol{Q}_L$ is the flow rate across the boundary, which must be balanced by the production and injection.

3.7.1.2 *Parabolic Problems*

To establish existence of solution, we have to consider a particular time approximation. For the implicit method

$$-\mathbf{T}\boldsymbol{u}^{n+1} = \frac{\mathbf{B}}{\Delta t}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}) + \boldsymbol{Q}^{n+1} \tag{3.168}$$

the matrix to be inverted is $\mathbf{T} + \mathbf{B}/\Delta t$. Since $\mathbf{B}$ has positive entries, $\mathbf{T} + \mathbf{B}/\Delta t$ is *strictly diagonally dominant* and therefore *positive definite*. This means that eqn. (3.168) has a unique solution (see Appendix A). The same is true of the Crank–Nicolson method

$$-\tfrac{1}{2}(\mathbf{T}\boldsymbol{u}^n + \mathbf{T}\boldsymbol{u}^{n+1}) = \frac{\mathbf{B}}{\Delta t}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) + \tfrac{1}{2}(\boldsymbol{Q}^n + \boldsymbol{Q}^{n+1})$$

where the matrix to be inverted is $(\frac{1}{2}\mathbf{T} + \mathbf{B}/\Delta t)$.

The concept of material balance is also naturally extended to time-dependent problems. The mass of fluid in the block i is $m_i = V_i(\phi/B)_i$ and the rate of change of mass due to compressibility is

$$\frac{\mathrm{d}m_i}{\mathrm{d}t} = c_i V_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = V_i \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\phi}{B}\right)_i \tag{3.169}$$

Again by summing up the equations (3.162) we obtain

$$Q_L + \sum_{i=1}^{N} \frac{\mathrm{d}m_i}{\mathrm{d}t} + \sum_{i=1}^{N} Q_i = 0 \tag{3.170}$$

which is the continuous-in-time form of the material balance equation. When the time is discretised, this equation takes on the form

$$Q_L \Delta t + \sum_{i=1}^{N} (m_i^{n+1} - m_i^n) + \Delta t \sum_{i=1}^{N} Q_i = 0 \tag{3.171}$$

which expresses exactly the material balance between t^n and t^{n+1} if Q_i and Q_L are constant during that period.

Whether or not a particular time discretisation satisfies eqn. (3.171) depends on the definition of the operator $\Delta_t(cu)$ approximating $\Delta t(cV\ \partial \mathrm{u}/\partial \mathrm{t})_i$ From eqn. (3.169) it follows that *any one-level time approximation will satisfy eqn.* (3.171) *if* Δ_t satisfies

$$\Delta_t(c_i u_i) = m_i^{n+1} - m_i^n = V_i\left[\left(\frac{\phi}{B}\right)_i^{n+1} - \left(\frac{\phi}{B}\right)_i^{n}\right] \tag{3.172}$$

Thus, the approximation that suggests itself

$$\Delta_t(c_i u_i) = V_i[(c_i u_i)^{n+1} - (c_i u_i)^n]$$

does *not* satisfy the condition of eqn. (3.172) as can be easily verified. In order to obtain the correct expansion for slightly compressible fluid, we write $m_i^{n+1} - m_i^n$ as

$$m_i^{n+1} - m_i^n = V_i\left[\frac{(\phi^{n+1} - \phi^n)}{B^{n+1}} + \left(\frac{1}{B^{n+1}} - \frac{1}{B^n}\right)\phi^n\right]_i$$

$$\equiv V_i\left[\left(\frac{\phi}{B}\right)^{n+1} - \left(\frac{\phi}{B}\right)^n\right]_i$$

Using

$$\phi^{n+1} - \phi^n = \phi^\circ c_R(u^{n+1} - u^n)$$

and

$$\left(\frac{1}{B^{n+1}} - \frac{1}{B^n}\right) = \frac{c_f}{B^\circ}(u^{n+1} - u^n)$$

we obtain

$$m_i^{n+1} - m_i^n = V_i\left(\frac{\phi^n c_f}{B^\circ} + \frac{\phi^\circ c_R}{B^{n+1}}\right)_i (u^{n+1} - u^n)_i \tag{3.173}$$

$$= V_i \bar{c}_i (u^{n+1})(u^{n+1} - u^n)_i$$

Another equally valid approximation is

$$m_i^{n+1} - m_i^n = V_i\left(\frac{\phi^{n+1} c_f}{B^\circ} + \frac{\phi^\circ c_R}{B^n}\right)_i (u^{n+1} - u^n)_i \tag{3.174}$$

$$= V_i \bar{\bar{c}}_i (u^{n+1})(u^{n+1} - u^n)_i$$

Obviously, eqn. (3.173) is preferable if $c_R = 0$ and eqn. (3.174) if $c_f = 0$, because then the corresponding coefficient becomes explicit.

With either definition, any two-level method, in particular explicit, backward difference and Crank–Nicolson will satisfy the material balance (3.171).

3.7.1.3 *General Remarks on Material Balance*

We have seen that there are two essential properties of difference equations that ensure material balance:

(a) *Flow coefficient matrix* **T** *be symmetric with zero row sums except for the boundaries.*

Although this is a sufficient condition, material balance may be preserved also for some schemes with nonsymmetric matrices (these will be encountered in Chapter 5).

(b) *The expansion of time derivative terms be carried out in a form that satisfies definition* (3.172).

In general, difference schemes that satisfy conservation of some quantity are called 'conservative schemes'. It should be noted that non-conservative schemes do not necessarily give meaningless results. For example, if c_f is small, we can replace B^{n+1} in eqn. (3.173) by B^n, which avoids iteration on c and will be an adequate approximation. Similarly, one might use the non-symmetric form of matrix **T** mentioned in Section 3.6.2.

However, when the equations are strongly nonlinear, non-conservative schemes may not only cause large material balance errors, but also induce instability. This has been observed in numerical fluid dynamics (Arakawa, 1966). Therefore, conservative schemes are recommended especially for nonlinear equations.

3.7.2 Treatment of Nonlinearities

Whenever the differential equation is nonlinear, the transmissibilities and/or coefficient of the time-derivative term will be functions of the solution (i.e., the dependent variable). For this case, eqn. (3.125) may be written as:

$$\begin{aligned} &\mathrm{T}_{i+1/2}(x_i, x_{i+1}, u_i, u_{i+1})(u_{i+1} - u_i) + \mathrm{T}_{i-1/2}(x_i, x_{i-1}, u_i, u_{i-1})(u_{i-1} - u_i) \\ &\qquad = V_i c_i(x_i, u_i)\frac{\mathrm{d}u_i}{\mathrm{d}t} + V_i q_i \qquad i = 1, \ldots, N \end{aligned} \tag{3.175}$$

where we have assumed that $\mathbf{T}_{i+½}$ is a function of u and x at i and $i + 1$ only. Thus, the above semi-discretisation produces the following nonlinear matrix equation:

$$-\mathbf{T}(\boldsymbol{u})\boldsymbol{u} = \mathbf{B}(\boldsymbol{u})\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{q} \tag{3.176}$$

Depending on how the time derivative is approximated, linear or nonlinear algebraic equations are obtained. Nonlinear equations may be linearised in different ways or solved iteratively. In general, the nonlinearities present in single-phase flow are less severe than those in multiphase flow. Consequently, some of the methods to be discussed here, which are necessary for multiphase flow problems (Chapter 5), are not normally required for single-phase problems.

3.7.2.1 *Explicit Time Approximations*

The forward difference formula which satisfies material balance is

$$-\mathbf{T}(\boldsymbol{u}^n)\boldsymbol{u}^n = \frac{1}{\Delta t}\mathbf{B}(\boldsymbol{u}^{n+1})(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) + \boldsymbol{Q} \tag{3.177}$$

Notice that the matrix **T** can be calculated explicitly, therefore the only nonlinearity is in matrix **B** and it appears because of the material balance condition, eqn. (3.173). Equation (3.177) represents N scalar nonlinear equations of the type $f(u_i^{n+1}) = 0$, which can be solved by any of the standard methods (e.g., Ortega and Rheinboldt, 1970; Ostrowski, 1973). The nonlinearity is weak; therefore even the simplest methods should be satisfactory. For example, to perform single iteration by substitution, we may rearrange the ith equation of eqn. (3.177) as:

$$\begin{aligned} u_i^{n+1} &= \frac{\Delta t}{V_i \bar{c}_i(u^{n+1})}\{-Q_i + [\mathrm{T}^n_{i+1/2}(u_{i+1} - u_i)^n + \mathrm{T}^n_{i-1/2}(u_{i-1} - u_i)^n]\} + u_i^n \\ &= \mathrm{f}(u_i^{n+1}) \end{aligned} \tag{3.178}$$

and iterate as (note that $u^{(v)}$ is always at the unknown level of time)

$$u_i^{(v)} = f(u_i^{(v-1)}) \qquad v = 1, 2, \ldots,$$

with the starting value $f(u_i^{(0)}) = f(u_i^n)$ until some specified convergence criterion is satisfied, such as

$$\left|\frac{u_i^{(v)} - u_i^{(v+1)}}{u_i^{(v+1)}}\right| < \varepsilon$$

Other methods (e.g., Regula Falsi, Newton's method) are equally applicable.

Let us now investigate stability of the explicit method. For the linear case we have the stability condition $\alpha = \Delta t/h2 \leq ½$. It is easy to see that for a linear equation of the

form

$$\lambda \partial^2 u/\partial x^2 = c\, \partial u/\partial t$$

the corresponding condition is:

$$C\alpha = \left(\frac{\lambda}{c}\right)\frac{\Delta t}{h^2} \leq \frac{1}{2}$$

It is possible to generalise these results for eqn. (3.175) using the concept of difference equations of positive type (Forsythe and Wasow, 1960; Section 14.1). Let us write eqn. (3.175) as:

$$u_i^{n+1} = \frac{\Delta t}{V_i c_i}\left[\mathrm{T}_{i-1/2}u_{i-1}^n + \left(\frac{V_i c_i}{\Delta t} - \mathrm{T}_{i-1/2} - \mathrm{T}_{i+1/2}\right)u_i^n + \mathrm{T}_{i+1/2}u_{i+1}^n - q_i V_i\right]$$

If all coefficients on the right are positive, the difference approximation is called '*of positive type*' and will be stable. This leads to the condition

$$\frac{\mathrm{T}_{i+1/2} + \mathrm{T}_{i-1/2}}{V_i c_i}\Delta t \leq 1$$

Since this must hold for all *i,* the stability of the explicit method will be maintained if

$$\max_i\left[\frac{(\mathrm{T}_{i+1/2} + \mathrm{T}_{i-1/2})}{V_i c_i}\right]\Delta t \leq 1 \tag{3.179}$$

at every time step. Note that the stability limit is time dependent and decreases as the compressibility of the system decreases. Unfortunately, the stability restriction (3.179) leads to unpractically small time steps for compressibilities typical for reservoir fluids (see Exercise 3.10).

Other explicit methods that are unconditionally stable have been used (Sheffield, 1970), but their application has been of limited value. Explicit methods of Runge–Kutta type are easily applied, since they always generate equations with coefficients evaluated at a known time level. We have found that for the gas-flow problem cited in Section 3.5.1 the standard fourth-order Runge–Kutta procedure has excellent stability

and compares favourably with forward- and backward-difference methods in terms of computer effort for a given accuracy.

3.7.2.2 *Implicit Time Approximations*

Let us consider the backward-difference approximation to eqn. (3.176):

$$-\mathbf{T}(\boldsymbol{u}^{n+1})\boldsymbol{u}^{n+1} = \frac{1}{\Delta t}\mathbf{B}(\boldsymbol{u}^{n+1})(\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}) + \boldsymbol{Q} \tag{3.180}$$

In the above equation the nonlinearity is due to the fact that $\mathbf{T}(\boldsymbol{u}^{n+1})$ and $\mathbf{B}(\boldsymbol{u}^{n+1})$ are matrices with elements which are functions of $\boldsymbol{u}^{n+1}$. We use the following notation in the discussion to follow:

$$\mathbf{T}(\boldsymbol{u}^{n+1}) \equiv \mathbf{T}^{n+1}$$

$$\mathbf{B}(\boldsymbol{u}^{n+1}) \equiv \mathbf{B}^{n+1}$$

The superscript on a matrix or a vector reflects the time or iteration level at which elements of the matrix are evaluated. For example, $\mathbf{T}^{(v)}$ means that the elements of $\mathbf{T}$ are evaluated using $\boldsymbol{u}^{(v)}$.

Simple iteration. The simplest method for the solution of eqn. (3.180) is obtained by writing

$$\left(\mathbf{T}^{(v-1)} + \frac{1}{\Delta t}\mathbf{B}^{(v-1)}\right)\boldsymbol{u}^{(v)} = \frac{1}{\Delta t}\mathbf{B}^{(v-1)}\boldsymbol{u}^{n} - \boldsymbol{Q} \qquad v = 1, 2, \ldots$$

with

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}^{n}$$

as the initial guess. The iteration is continued until convergence is obtained. Each iteration requires the same amount of work as the solution over a time step for the corresponding linear problem. For the simple single-phase flow problem being considered here, the nonlinearity is weak and the iteration process converges rapidly.

Newton's Method. A more powerful iteration scheme, which also works for strongly nonlinear problems to be introduced later, is Newton's method (or Newton–Raphson method). The nonlinear system of equations given by eqn. (3.180) may be written as

$$\left(\mathbf{T}^{n+1} + \frac{1}{\Delta t}\mathbf{B}^{n+1}\right)\boldsymbol{u}^{n+1} - \frac{1}{\Delta t}\mathbf{B}^{n+1}\boldsymbol{u}^{n} + \boldsymbol{Q} = \boldsymbol{0} \tag{3.182a}$$

or by calling the left-hand side of the above equation function $\boldsymbol{f}$ we can write

$$\boldsymbol{f}(\boldsymbol{u}^{n+1}) \equiv \boldsymbol{f}^{n+1} = \boldsymbol{0} \tag{3.182b}$$

A nonlinear system of equations of the form given above may be solved by Newton's method (e.g. Henrici, 1962) as defined below:

$$\boldsymbol{u}^{(\nu)} - \boldsymbol{u}^{(\nu-1)} = -[\mathbf{F}^{(\nu-1)}]^{-1}\boldsymbol{f}^{(\nu-1)} \qquad \nu = 1, 2, \ldots \tag{3.183}$$

where $\mathbf{F}$ is the Jacobian matrix of the vector function $\boldsymbol{f}$:

$$\mathbf{F}^{(\nu)} = \left(\frac{\partial f_i}{\partial u_j}\right)^{(\nu)} \tag{3.184}$$

and it is the vectorial extension of derivative used in the classical Newton's method for single equation. Equation (3.183) may be written in the following form which is more convenient for computational purposes:

$$\mathbf{F}^{(\nu-1)}\boldsymbol{\delta}^{\nu} = -\boldsymbol{f}^{(\nu-1)} \tag{3.185a}$$

And

$$\boldsymbol{u}^{(\nu)} = \boldsymbol{u}^{(\nu-1)} + \boldsymbol{\delta}^{(\nu)} \qquad \text{for } \nu = 1, 2, \ldots \tag{3.185b}$$

with

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}^{n}$$

As the iteration proceeds both $\boldsymbol{\delta}$ and $\boldsymbol{f}$ approach $\boldsymbol{0}$, provided the method converges.

Several variants of Newton's method are possible and they are discussed, for example, by Ortega and Rheinboldt (1970) and Ostrowski (1973). The standard method given above is also referred to as the *variable tangent method*.

A *variable secant* or *chord* method is obtained if the elements of F are evaluated by

$$\frac{\partial f_i^{(\nu)}}{\partial u_j} = \frac{f_i(\ldots, u_j^{(\nu)}, \ldots) - f_i(\ldots, u_j^{(\nu-1)}, \ldots)}{u_j^{(\nu)} - u_j^{(\nu-1)}} \tag{3.186}$$

Two other methods are obtained if the Jacobian in the above two methods is computed only once and assumed constant for further iterations. These are called *fixed tangent* and *fixed secant* methods; such iterative schemes are called *stationary*. For either method we can write

$$\mathbf{F}^n \boldsymbol{\delta}^{(\nu)} = -\boldsymbol{f}^{(\nu-1)} \qquad \nu = 1, 2, \ldots \tag{3.187}$$

where the **F** matrix is evaluated using $\boldsymbol{u}$ at the old time level n. This scheme reduces the work required per iteration but the rate of convergence is also reduced.

Let us consider in detail how Newton's method is applied. If we look at eqn. (3.185), it is clear that the main problem facing us is the evaluation of matrix **F.** From eqn. (3.175) we see that **F** will have the same sparse structure as matrix **T** since for a given row i, only

$$\frac{\partial f_i}{\partial u_{i-1}} \qquad \frac{\partial f_i}{\partial u_i} \qquad \text{and} \qquad \frac{\partial f_i}{\partial u_{i+1}}$$

will be non-zero. For the ith equation being considered we have

$$\frac{\partial f_i}{\partial u_{i-1}} = \mathrm{T}_{i-1\,2} + \frac{\partial \mathrm{T}_{i-1\,2}}{\partial u_{i-1}}(u_{i-1} - u_i) \tag{3.188a}$$

$$\frac{\partial \mathrm{f}_i}{\partial \mathrm{u}_i} = -(\mathrm{T}_{i-1\,2} + \mathrm{T}_{i+1\,2}) + \frac{\partial \mathrm{T}_{i-1\,2}}{\partial u_i}(u_{i-1} - u_i) + \frac{\partial \mathrm{T}_{i+1\,2}}{\partial u_i}(u_{i+1} - u_i) - \frac{V_i}{\Delta t}\left[c_i + \frac{\partial c_i}{\partial u_i}(u_i - u_i^n)\right] \tag{3.188b}$$

$$\frac{\partial f_i}{\partial u_{i+1}} = \mathrm{T}_{i+1\,2} + \frac{\partial \mathrm{T}_{i+1\,2}}{\partial u_{i+1}}(u_{i+1} - u_i) \tag{3.188c}$$

A convenient way of writing **F** is

$$\mathbf{F} = \mathbf{T} + \mathbf{T}' + \frac{1}{\Delta t}(\mathbf{B} + \mathbf{B}') \tag{3.189}$$

where the non-zero elements of $\mathbf{T}'$ and $\mathbf{B}'$ are

$$t'_{i,i-1} = \frac{\partial \mathbf{T}_{i-1\,2}}{\partial u_{i-1}}(u_{i-1} - u_i) \tag{3.190a}$$

$$t'_{i,i} = \frac{\partial \mathbf{T}_{i-1\,2}}{\partial u_i}(u_{i-1} - u_i) + \frac{\partial \mathbf{T}_{i+1\,2}}{\partial u_i}(u_{i+1} - u_i) \tag{3.190b}$$

$$t'_{i,i+1} = \frac{\partial \mathbf{T}_{i+1\,2}}{\partial u_{i+1}}(u_{i+1} - u_i) \tag{3.190c}$$

$$b'_{i,i} = V_i \frac{\partial c_i}{\partial u_i}(u_i - u_i^n) \tag{3.191}$$

The matrix equation for Newton's method (3.185a) may now be written as

$$\left[\mathbf{T} + \frac{1}{\Delta t}\mathbf{B} + \mathbf{T}' + \frac{1}{\Delta t}\mathbf{B}'\right]^{(\nu)} \boldsymbol{\delta}^{(\nu+1)} = -\mathbf{T}^{(\nu)}\boldsymbol{u}^{(\nu)} - \frac{1}{\Delta t}\mathbf{B}^{(\nu)}(\boldsymbol{u}^{(\nu)} - \boldsymbol{u}^n) - \boldsymbol{Q} \tag{3.192}$$

Let us now consider a special case of *linear* equations, where $\mathbf{T}' = \mathbf{B}' \equiv 0$. With $\boldsymbol{u}^{(0)} = \boldsymbol{u}^n$, the first iteration of (3.192) becomes

$$\left(\mathbf{T} + \frac{1}{\Delta t}\mathbf{B}\right)(\boldsymbol{u}^{(1)} - \boldsymbol{u}^n) = -\mathbf{T}\boldsymbol{u}^n - \boldsymbol{Q} \tag{3.193}$$

Comparison with (3.180) shows that these two equations are equivalent if $\boldsymbol{u}^{(1)} = \boldsymbol{u}^{n+1}$. Therefore the first iteration of Newton's method (3.183) *corresponds to the solution of a linearised version of the boundary value problem.*

Similar treatment is possible for the Crank–Nicolson method. In order that this method remains $O(\Delta t^2)$, in the nonlinear case all nonlinearities must be evaluated at the $n + \frac{1}{2}$ level:

$$-\tfrac{1}{2}(\mathbf{T}^{n+1/2}\boldsymbol{u}^{n+1} + \mathbf{T}^{n+1/2}\boldsymbol{u}^n) = \frac{1}{\Delta t}\mathbf{B}^{n+1/2}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) + \boldsymbol{Q} \tag{3.194}$$

We first note that eqn. (3.194) can be rewritten in the form of eqn. (3.182a):

$$\left(\mathbf{T}^{n+1/2} + \frac{2}{\Delta t}\mathbf{B}^{n+1/2}\right)(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) = -2(\mathbf{T}^{n+1/2}\boldsymbol{u}^n + \boldsymbol{Q}) \tag{3.195}$$

If the $n + \frac{1}{2}$ level is approximated by

$$\boldsymbol{u}^{n+1/2} = \tfrac{1}{2}(\boldsymbol{u}^n + \boldsymbol{u}^{n+1})$$

matrices $\mathbf{T}^{n+1/2}$ and $\mathbf{B}^{n+1/2}$ are again functions of $\boldsymbol{u}^{n+1}$ and we can write the Newton's method for this problem by direct analogy with eqn. (3.192):

$$\left[\mathbf{T}+\frac{2}{\Delta t}\mathbf{B}+\left(\mathbf{T}+\frac{2}{\Delta t}\mathbf{B}\right)'\right]^{(\nu)}(\boldsymbol{u}^{(\nu+1)}-\boldsymbol{u}^{(\nu)}) \tag{3.196}$$

$$=-2(\mathbf{T}^{(\nu)}\boldsymbol{u}^{(\nu)}+\boldsymbol{Q})+\left(\mathbf{T}+\frac{2}{\Delta t}\mathbf{B}\right)^{(\nu)}(\boldsymbol{u}^{(\nu)}-\boldsymbol{u}^{n})$$

Equation (3.194) can also be solved by a two-stage process. As shown in Section 3.3.2, it can be written as

$$-\mathbf{T}^{n+1/2}\boldsymbol{u}^{n}=\frac{2}{\Delta t}\mathbf{B}^{n+1/2}(\boldsymbol{u}^{n+1/2}-\boldsymbol{u}^{n})+\boldsymbol{Q} \tag{3.197a}$$

$$-\mathbf{T}^{n+1/2}\boldsymbol{u}^{n+1}=\frac{2}{\Delta t}\mathbf{B}^{n+1/2}(\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n+1/2})+\boldsymbol{Q} \tag{3.197b}$$

The first equation is of the explicit type, but the coefficient matrices are at the unknown $(n+\frac{1}{2})$ level of time. Even though this equation is nonlinear, it can be solved easier than the equations of implicit type (e.g. by simple iteration). The second equation, which is of implicit type, is then linear since $\boldsymbol{u}^{n+1/2}$ is now known.

3.7.2.3 *Methods of Linearisation*

Simple iteration. The simplest and a frequently used method of linearisation is obtained by allowing the nonlinearities to lag one time step behind. Thus, for the backward difference method we would solve:

$$-\mathbf{T}^{n}\boldsymbol{u}^{n+1}=\frac{1}{\Delta t}\mathbf{B}^{n}(\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n})+\boldsymbol{Q}$$

It is important to realise that this linearisation will decrease the order of approximation to $O(\Delta t)$ for higher-order methods and therefore their use may not be justified.

Extrapolation. Extrapolation of the solution to the desired time level can, however, preserve the order of convergence for higher order methods. Suppose we require $\boldsymbol{u}$ at the level $n+l$, where $0<l\leq 1$; then by linear extrapolation using $\boldsymbol{u}^{n-1}$ and $\boldsymbol{u}^{n}$, we have

$$\boldsymbol{u}(t^{n+l})\simeq\boldsymbol{u}^{n+l}=\boldsymbol{u}^{n}+\frac{\Delta t^{n}}{\Delta t^{n-1}}l(\boldsymbol{u}^{n}-\boldsymbol{u}^{n-1}) \tag{3.198}$$

Higher-order extrapolations are possible but they increase storage requirements. Because eqn. (3.198) is an $O(\Delta t)$ approximation to $\boldsymbol{u}(t^{n+l})$, second-order methods will preserve their order when matrices are evaluated at extrapolated time levels. This approach works well if the nonlinearities are not too strong and can be used successfully for single-phase flow and miscible flow. In multiphase flow, the explicit nature of the extrapolation causes stability limitations similar to those present in explicit (or linearised by lagging behind) formulations.

We note here that Douglas (1961) and Douglas and Jones (1963) presented a modified Crank–Nicolson procedure of predictor–corrector type, which is $O(\Delta t^2)$ and does not involve solving nonlinear equations.

Semi-implicit method. Finally, we describe the method of linearisation which is the basis of the 'semi-implicit' treatment of transmissibilities. Although this approach was originally developed to preserve stability in multiphase flow (see Chapter 5 for details), it applies equally well to any nonlinear problem.

Consider a typical term of $\mathbf{T}^{n+1}\boldsymbol{u}^{n+1}$ in the nonlinear eqn. (3.180):

$$\mathrm{T}_{i+1/2}(u_i^{n+1}, u_{i+1}^{n+1})(u_{i+1} - u_i)^{n+1} \tag{3.199}$$

Expanding $\mathbf{T}_{i+1/2}^{n+1}$ in Taylor series and retaining only the lowest order terms we have

$$\mathrm{T}_{i+1/2}^{n+1} = \mathrm{T}_{i+1/2}^{n} + \left(\frac{\partial \mathrm{T}_{i+1/2}}{\partial u_i}\right)^n (u_i^{n+1} - u_i^n) + \left(\frac{\partial \mathrm{T}_{i+1/2}}{\partial u_{i+1}}\right)^n (u_{i+1}^{n+1} - u_{i+1}^n) \tag{3.200}$$

Substitution of eqn. (3.200) in (3.199) will produce nonlinear terms which must be linearised. This is achieved in the following way:

$$\left(\frac{\partial \mathrm{T}_{i+1\ 2}}{\partial u_i}\right)^n (u_i^{n+1} - u_i^n)(u_{i+1} - u_i)^{n+1} \cong \left(\frac{\partial \mathrm{T}_{i+1/2}}{\partial u_i}\right)^n (u_i^{n+1} - u_i^n)(u_{i+1} - u_i)^n$$

where we have replaced $(u_{i+1} - u_i)^{n+1}$ by $(u_{i+1} - u_i)^n$. With the above approximation eqn. (3.199) is linearised since

$$\mathrm{T}_{i+1/2}^{n+1}(u_{i+1}-u_i)^{n+1} \cong \mathrm{T}_{i+1/2}^{n}(u_{i+1}-u_i)^{n+1} + \left[\left(\frac{\partial \mathrm{T}_{i+1/2}}{\partial u_i}\right)^n (u_{i+1}-u_i)^n\right](u_i^{n+1}-u_i^n) + \left[\left(\frac{\partial \mathrm{T}_{i+1/2}}{\partial u_{i+1}}\right)^n (u_{i+1}-u_i)^n\right](u_{i+1}^{n+1}-u_{i+1}^n) \tag{3.201}$$

Obviously, the first term is an element of $\mathbf{T}^n\mathbf{u}^{n+1}$, and the other two terms are seen to be elements of $\mathbf{T}'^n(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n)$, where $\mathbf{T}'$ is defined by eqns. (3.190). Linearisation of $\mathbf{B}^{n+1}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n)$ turns out to be just $\mathbf{B}^n(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n)$, because we have replaced $(u^{n+1} - u^n)_i$ by $(u^n - u^n)_i = 0$ in the nonlinear terms. Therefore the linearised form of eqn. (3.180) will be

$$\left[\mathbf{T} + \frac{1}{\Delta t}\mathbf{B} + \mathbf{T}'\right]^n (\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) = -\mathbf{T}^n\boldsymbol{u}^n - \boldsymbol{Q} \tag{3.202}$$

This can be compared with the Newton's iteration (eqn. (3.192)) for $v = 0$. Since it follows from eqn. (3.191) that $\mathbf{B}'^n \equiv 0$, eqn. (3.202) is identical with eqn. (3.192) if we set $\boldsymbol{u}^{(1)} = \boldsymbol{u}^{n+1}$. Therefore we have the following important result:

> *The linearisation* (3.201) *is identical with the first iteration of Newton's method.*

3.8 CONCLUDING REMARKS

In this chapter we have attempted to introduce most of the concepts required for reservoir simulation, that can be introduced in the context of a simple one-dimensional single-phase flow problem. Much of the material in this chapter is only important as a foundation for our future discussion of multiphase problems.

This chapter does not contain a survey of finite difference methods. Several excellent books that do this job were mentioned in the text.

The reader attempting to develop computer simulation programs for one-dimensional single-phase flow problems will also require the material in the next chapter. A thorough understanding of the material contained in this chapter is prerequisite to the understanding of more complicated problems.

EXERCISES

Exercise 3.1

Derive the 'one-sided' approximation to U' at the boundary:

$$U_1' = (-U_3 + 4U_2 - 3U_1)/2h + O(h^2) \tag{A}$$

Solution Outline

Method (*a*) Use Taylor series to expand U_2 and U_3 about U_1 and solve for U_1'.

Method (*b*) Show that the interpolating polynomial of 2nd degree going through U_1, U_2 and U_3 is

$$P_2 = U_1 + (U_2 - U_1)\alpha + (U_3 - 2U_2 + U_1)\frac{\alpha(\alpha - 1)}{2} \tag{B}$$

where $\alpha = x/h$. Then show that $U_1' \cong P_2'(x = 0)$ gives the above formula (A).

Exercise 3.2

Derive the difference equation (*eqn.* (*3.20*)) *by the variational method using Chapeau basis functions.*

Solution Outline

First show that the function $v(x)$ given by eqn. (3.24) is linear within each interval, i.e., for $x \in (x_i, x_{i+1})$

$$v(x) = c_i + \xi(c_{i+1} - c_i) \qquad \xi = (x - x_i)/h \tag{A}$$

Next, using eqn. (3.23) show that the functional to be minimised can be approximated by

$$I_A = \sum_i \left[\frac{1}{2h}(c_{i+1} - c_i)^2 - \frac{h}{2}(q_i c_i + q_{i+1} c_{i+1})\right] \tag{B}$$

Finally, show that the relationships $\partial I_A/\partial c_i = 0$ give

$$\frac{1}{h^2}\Delta^2 c_i - q_i = 0 \tag{C}$$

the desired equation.

Exercise 3.3

Consider the linear equation $\nabla^2 u = q$ *approximated by* $1/h^2\ \Delta^2 u_i = q_i$, *with boundary conditions* $u_0 = u_{N+1} = 0$. *Prove that* $||\boldsymbol{e}|| \to 0$ *as* $h \to 0$.

Solution Outline

The solution of the error equation (eqn. (3.42)) can be assumed to be of the form

$$e_i = \sum_k e_i^{(k)} \tag{A}$$

where $e_i^{(k)}$ satisfies

$$\frac{1}{h^2}\Delta^2 e_i^{(k)} = -\delta_{ik} R_i \tag{B}$$

The solution of (B) is

$$e_k^{(k)} = R_k \frac{k(N+1-k)}{N+1} h^2$$

$$e_i^{(k)} = \frac{i}{k} e_k^{(k)} \qquad i = 1, 2, \ldots, k-1$$

$$e_i^{(k)} = \frac{(N+1-i)}{(N+1-k)} e_k^{(k)} \qquad i = k+1, \ldots, N$$

and therefore

$$\max_{i,k} |e_i^{(k)}| < C \max_k |R_k|.$$

Since $R_i = O(h^2)$, the desired result follows.

Exercise 3.4

Derive the stability conditions for (*a*) *the forward, and* (*b*) *the backward difference approximations by the Fourier series method.*

Solution Outline

(a) Substitute the formal solution

$$u_i^{n(m)} = \xi^n \exp(\sqrt{-1}\, mih) \tag{A}$$

into eqn. (3.30) to obtain

$$\xi = 1 + \alpha(\exp(\sqrt{-1}mh) + \exp(-\sqrt{-1}mh) - 2) \tag{B}$$

Use of relations $\exp(\sqrt{-1}x) = \cos x + \sqrt{-1}\sin x$ and $1 - \cos x = 2\sin^2(\alpha/2)$ gives

$$\xi = 1 - 2\alpha[1 - \cos(mh)] = 1 - 4\alpha\sin^2\left(\frac{mh}{2}\right) \tag{C}$$

Considering all $m > 0$ and $h \to 0$, show that $|\xi| \leq 1$ if $\alpha \leq \frac{1}{2}$.

(b) Using the same procedure for eqn. (3.31) gives the amplification factor as

$$\xi = \frac{1}{1 + 2\alpha[1 - \cos(mh)]} = \frac{1}{1 + 4\alpha\sin^2(mh/2)} \tag{D}$$

and $|\xi| < 1$ for all α.

Exercise 3.5

Investigate the stability of (*a*) *the forward and* (*b*) *backward difference methods by the matrix method.*

Solution Outline

Let **M** be an $N \times N$ matrix

$$M = \begin{bmatrix} a & -b & & & & \\ -b & a & -b & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & -b & a & -b \\ & & & & -b & a \end{bmatrix}$$

where $a = 2b + c$, and $a, b, c > 0$. Show that the eigenvalues λ_s and eigenvectors $\boldsymbol{u}_s$ of **M** are

$$\lambda_s = -c - 4b \sin^2 \frac{s\pi}{2(N+1)} \qquad s = 1, 2, \ldots, N \tag{A}$$

$$\boldsymbol{u}_s = \begin{bmatrix} \sin\left(\dfrac{s\pi}{N+1}\right) \\ \vdots \\ \sin\left(\dfrac{is\pi}{N+1}\right) \\ \vdots \\ \sin\left(\dfrac{Ns\pi}{N+1}\right) \end{bmatrix}$$

(a) Use (A) with appropriate choice of constants to get eigenvalues of $\mathbf{B} = \alpha\mathbf{A} + \mathbf{I}$ for the forward difference method

$$\lambda_s = 1 - 4\alpha \sin^2 \frac{s\pi}{2(N+1)} \qquad s = 1, 2, \ldots, N \tag{B}$$

and stability condition from $|\lambda_i| < 1$.

(b) The eigenvalues of $\mathbf{B} = -(\alpha\mathbf{A} + \mathbf{I})^{-1}$ for the backward difference method are

$$\lambda_s = \frac{1}{1 + 4\alpha \sin^2\left(\dfrac{s\pi}{2(N+1)}\right)} \tag{C}$$

Hint: If $\mathbf{A}$ has eigenvalues λ_i, $\mathbf{A}^{-1}$ has eigenvalues $1/\lambda_i$.

Exercise 3.6

Investigate the consistency of the DuFort–Frankel approximation, eqn. (3.58).

Solution Outline

Show that the local truncation error is

$$R_i = O(\Delta t^2) + O(\Delta x^2) + O\left[\left(\frac{\Delta t}{\Delta x}\right)^2\right] \tag{D}$$

Formulate the condition for consistency as $\Delta t \to 0$. Show that for

α = constant, the approximation (3.58) will converge to the solution of

$$\frac{\partial^2 U}{\partial x^2} = \frac{\partial U}{\partial t} + \alpha^2 \frac{\partial^2 U}{\partial t^2} \tag{B}$$

The above result can be used in reverse. First, form the central difference approximation to $\alpha^2(\partial^2 U/\partial t^2)$. Then show that adding this term to the Richardson approximation results in the DuFort and Frankel formula. Such 'perturbation' techniques are often used to derive new methods.

Exercise 3.7

Determine the order of approximation of method (3.66).

Solution Outline

Expand the terms in Taylor series about u_i^{n+1} to obtain

$$R_i = -U^{\mathrm{IV}} \frac{h^2}{12} - U^{\cdots} \frac{\Delta t^2}{3} + O(h^3) + O(\Delta t^3)$$

Exercise 3.8

Derive the matrix form of Runge–Kutta method for the linear eqn. (3.69) *and discuss its computational advantage compared to the recursive algorithm.*

Solution Outline

Denote $\mathbf{M} = -(1/h^2)\mathbf{E}$, then coefficients $\boldsymbol{a}_i$ will be

$$\boldsymbol{a}_1 = \mathbf{M}\boldsymbol{u}^n$$

$$\boldsymbol{a}_2 = (\mathbf{M} + \tfrac{1}{2}\Delta t\mathbf{M}^2)\boldsymbol{u}^n$$

Substitution in eqn. (3.79) gives

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t[\mathbf{M} + \tfrac{1}{2}\Delta t\mathbf{M}^2 + \tfrac{1}{6}\Delta t^2\mathbf{M}^3 + \tfrac{1}{24}\Delta t^3\mathbf{M}^4]\boldsymbol{u}^n \\ &= \boldsymbol{u}^n + \Delta t\mathbf{B}(\Delta t)\boldsymbol{u}^n \end{aligned} \tag{A}$$

The recursive algorithm can be written as

$$\left.\begin{aligned} \boldsymbol{a}_1 &= \mathbf{M}\boldsymbol{u}^n \\ \boldsymbol{a}_2 &= \boldsymbol{a}_1 + \tfrac{1}{2}\Delta t\mathbf{M}\boldsymbol{a}_1 \\ \boldsymbol{a}_3 &= \boldsymbol{a}_1 + \tfrac{1}{2}\Delta t\mathbf{M}\boldsymbol{a}_2 \\ \boldsymbol{a}_4 &= \boldsymbol{a}_1 + \Delta t\mathbf{M}\boldsymbol{a}_3 \end{aligned}\right\} \tag{B}$$

In case of constant Δt, an algorithm based on (A) will be faster since **B** is still a tridiagonal matrix. Algorithm (B) will require approximately four times as much computations. If Δt is variable, both algorithms are comparable, since **B** can be evaluated by

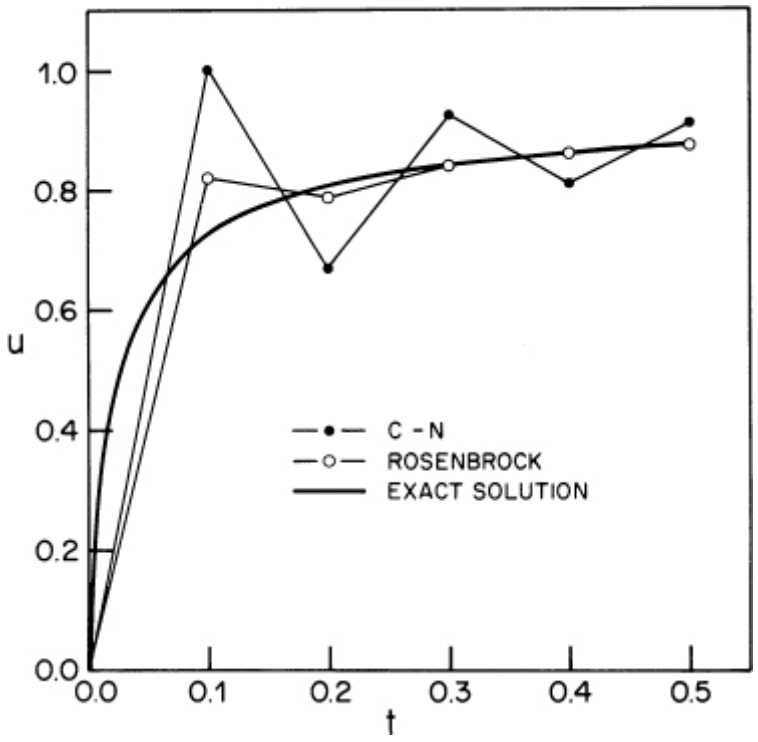
$$\mathbf{B} = \mathbf{M}[\mathbf{I} + \tfrac{1}{2}\Delta t\mathbf{M}(\mathbf{I} + \tfrac{1}{3}\Delta t\mathbf{M}(\mathbf{I} + \tfrac{1}{4}\Delta t\mathbf{M}))] \tag{C}$$

Exercise 3.9

Investigate the properties of the second-order Rosenbrock's scheme (3.89), (3.90) *and compare with the C–N scheme.*

Solution Outline

It can be shown that for $\Delta t \to \infty$, the amplification factor ξ approaches – 1 for the C–N scheme, and 0 for the Rosenbrock's scheme. Schemes for which $\xi \to 0$ as $\Delta t \to \infty$ are called *L*-acceptable (Ehle, 1973). Therefore C–N oscillates much more for large Δt as shown above for the simple diffusivity equation $\partial^2 u/\partial x^2 = \partial u/\partial t$ with initial condition $u(x,\ 0) = 0$ and boundary conditions $u(0,\ t) = 1$, $u_x(1,\ t) = 0$. The figure shows the solutions as a function of time at $x = 0.1$ computed with $\Delta x = 0.1$ *and* $\Delta t = 0.1$.



Exercise 3.10

Find the stability limit for the forward difference method for a one-dimensional problem with the following parameters: $\Delta x = 100ft$, $k = 100$ *md*, $\mu = 1$ *cp*, $B = 1$, $\phi = 0.1$, $c_f = 4 \times 10^{-6}$ *psia*$^{-1}$, $c_R = 3 \times 10^{-6}$ *psia*$^{-1}$.

Solution Outline

Equation (3.179) with the approximation

$$c \sim \frac{\phi}{B}(c_f + c_R) \tag{A}$$

gives for a constant cross-section

$$\Delta t \leq \frac{1}{2}\frac{\mu\phi\,\Delta x^2(c_f + c_R)}{0{\cdot}006328\,k} \tag{B}$$

where the conversion constant 0•006328 gives the flow in units of ft^3/day. Substitution of numerical values in (B) gives $\Delta t \leq 5{\bullet}53 \times 10^{-3}$ day.

CHAPTER 4

SOLUTION OF TRIDIAGONAL MATRIX EQUATIONS

4.1 INTRODUCTION

All of the one-dimensional flow problems discussed in the previous chapter (except for the case of periodic boundary conditions) resulted in a system of N simultaneous algebraic equations with the form:

$$\begin{aligned} a_1u_1 + b_1u_2 &= d_1 \\ c_iu_{i-1} + a_iu_i + b_iu_{i+1} &= d_i \qquad i = 2, 3, \ldots, N-1 \\ c_Nu_{N-1} + a_Nu_N &= d_N \end{aligned} \tag{4.1}$$

where coefficients a_i b_i and c_i are known and d_i on the right-hand side of the equations are also known. This system of equations may also be written in matrix form as:

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d} \tag{4.2}$$

where $\boldsymbol{u}^{\mathrm{T}} = [u_1, u_2, \ldots, u_N]$, $\boldsymbol{d}^{\mathrm{T}} = [d_1, d_2, \ldots, d_N]$ and $\mathbf{A}$ is a tridiagonal matrix:

$$\mathbf{A} = \begin{bmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & c_N & a_N \end{bmatrix} \tag{4.3}$$

For problems with boundary conditions of the fourth kind (cyclic or periodic boundary conditions) the system of equations has a slightly different form:

$$\begin{aligned} c_1u_N + a_1u_1 + b_1u_2 &= d_1 \\ c_iu_{i-1} + a_iu_i + b_iu_{i+1} &= d_i \qquad i = 2, 3, \ldots, N-1 \\ c_Nu_{N-1} + a_Nu_N + b_Nu_1 &= d_N \end{aligned} \tag{4.4}$$

This system of equations may be written in matrix form as:

$$\mathbf{C}\boldsymbol{u} = \boldsymbol{d} \tag{4.5}$$

where vectors $\boldsymbol{u}$ and $\boldsymbol{d}$ are defined as before, and

$$\mathbf{C} = \begin{bmatrix} a_1 & b_1 & & & c_1 \\ c_2 & a_2 & b_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ b_N & & & c_N & a_N \end{bmatrix} \tag{4.6}$$

The remainder of this chapter deals with methods for the solution of eqns. (4.2) and (4.5). These equations may be solved either by direct elimination or by some iterative method. In this chapter we will only consider the former class of method and leave the discussion of iterative methods for later chapters. The reason for this choice is that in one-dimensional problems no known iterative method can compete with direct elimination.

In all cases it is assumed that the solution exists. The existence of the solution has been discussed in the previous chapter.

4.2 METHODS OF SOLUTION

In this section we will first present one method for the general cases of eqns. (4.2) and (4.5). Following this, some algorithms that are applicable under certain special conditions will be presented. It is not possible to give all methods for the solution of various forms of eqns. (4.2) and (4.5); we will, however, deal with methods that are most important for the numerical solution of practical partial differential equations.

4.2.1 Thomas' Algorithm

The most popular method for the solution of eqn. (4.2) is derived by writing **A** as a product of two matrices:

$$\mathbf{A} = \mathbf{W}\mathbf{Q} \tag{4.7}$$

where **W** is a lower triangular matrix and **Q** an upper triangular matrix. For the special tridiagonal matrix **A** being considered the two matrices **Q** and **W** may be written as:

$$\mathbf{W} = \begin{bmatrix} w_1 & & & & \\ c_2 & w_2 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & c_N & w_N \end{bmatrix} \tag{4.8}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & q_1 & & & \\ & 1 & q_2 & & \\ & & \cdot & \cdot & \\ & & & \cdot & q_{N-1} \\ & & & & 1 \end{bmatrix} \tag{4.9}$$

We note that the lower diagonal of **W** has the same elements as the lower diagonal of **A**, and the elements of the main diagonal of **Q** have been arbitrarily selected to be all ones. Now if the elements of **WQ** are equated to the elements of **A**, term-by-term, we obtain $2N-1$ equations for the unknowns:

$$w_1, w_2, \ldots, w_N$$

and

$$q_1, q_2, \ldots, q_{N-1}$$

The desired relationships are:

$$w_1 = a_1 \tag{4.10}$$

$$q_{i-1} = b_{i-1}/w_{i-1} \qquad i = 2, 3, \ldots, N \tag{4.11}$$

$$w_i = a_i - c_i q_{i-1} \qquad i = 2, 3 \ldots, N \tag{4.12}$$

Substituting eqn. (4.7) into (4.2) we have

$$\mathbf{WQ}\boldsymbol{u} = \boldsymbol{d} \tag{4.13}$$

Let

$$\mathbf{Q}\boldsymbol{u} = \boldsymbol{g} \tag{4.14}$$

Then

$$\mathbf{W}\boldsymbol{g} = \boldsymbol{d} \tag{4.15}$$

Since **W** is a lower triangular matrix, the first equation of the system (4.15) has only one unknown and it can be solved for g_1:

$$g_1 = \frac{d_1}{w_1} \tag{4.16}$$

The remaining equations in the system (4.15) can be solved by forward elimination:

$$g_i = \frac{d_i - c_i g_{i-1}}{w_i} \qquad i = 2, 3, \ldots, N \tag{4.17}$$

Now we observe that $\boldsymbol{g}$ is the right-hand-side vector for the system (4.14) with the coefficient matrix **Q** now known from (4.10) to (4.12). Since **Q** is an upper triangular matrix the last equation in this system has only one unknown and it can be solved first:

$$u_N = g_N$$

The remaining equations can now be solved by backward substitution:

$$u_i = g_i - q_i u_{i+1} \qquad i = N-1, N-2, \ldots, 1 \tag{4.18}$$

For computational purposes it is not necessary to store w_i and the algorithm may be written as:

1. Set $q_1 = b_1/a_1, \quad g_1 = d_1/a_1$
2. Compute for $i = 2, 3, \ldots, N$

$$p_i = a_i - c_i q_{i-1}$$

$$q_i = \frac{b_i}{p_i}$$

$$g_i = \frac{d_i - c_i g_{i-1}}{p_i}$$

3. Set $u_N = g_N$

4. Compute for $i = N-1, N-2, \ldots, 1$

$$u_i = g_i - q_i u_{i+1}$$

The above algorithm requires five multiplications or divisions and three subtractions per grid point. This algorithm, in this or slightly different form, has been discussed in many texts (e.g., Lapidus, 1962; Richtmyer and Morton, 1967; Ames, 1969; Von Rosenberg, 1969). We present a Fortran computer program for Thomas' algorithm in Appendix **B**. The algorithm is, in essence, a form of Gaussian elimination.

When using this algorithm we must be sure that

$$a_1 \neq 0 \tag{4.19}$$

and

$$p_i = a_i - c_i q_{i-1} \neq 0 \tag{4.20}$$

In cases where the above conditions are not satisfied the system size can be reduced. One must also guard against serious build-up of round-off error, in cases where $|p_i|$ is small. This problem may be handled by the process of elimination called 'pivoting'.

Ahlberg *et al*. (1967) have extended Thomas' algorithm to solve eqn. (4.5). The procedure reduces to the Thomas algorithm when c_1 and b_N are zero. The algorithm may be written as:

1. Set $q_1 = b_1/a_1 \quad g_1 = d_1/a_1 \quad s_1 = -c_1/a_1 \; t_N = 1 \quad v_N = 0$

2. Compute for $i = 2, 3, \ldots, N$

$$p_i = a_i - c_i q_{i-1}$$

$$s_i = -\frac{c_i s_{i-1}}{p_i}$$

$$q_i = \frac{b_i}{p_i}$$

$$g_i = \frac{d_i - c_i g_{i-1}}{p_i}$$

3. Compute for $i = N-1, N-2, \ldots, 1$

$$t_i = -q_i t_{i+1} + s_i$$

$$v_i = g_i - q_i v_{i+1}$$

4. Compute $u_N = \frac{d_N - c_N v_{N-1} - b_N v_1}{a_N + t_1 b_N + t_{N-1} c_N}$

5. Compute for $i = N-1, N-2, \ldots, 1$

$$u_i = v_i + t_i u_N$$

The above procedure requires eight multiplications or divisions and five subtractions per grid point.

4.2.2 Tang's Algorithm

Tang (1969) has presented an algorithm for the solution of eqn. (4.5). This algorithm may be derived in a manner similar to the procedure discussed for the Thomas algorithm. The algorithm is:

1. Set $\zeta_1 = 0$ $\beta_1 = -1$ $\gamma_1 = 0$
2. Compute

$$\zeta_2 = \frac{d_1}{b_1} \qquad \beta_2 = \frac{a_1}{b_1} \qquad \gamma_2 = \frac{c_1}{b_1}$$

3. Compute for $i = 2, 3, \ldots, N-1$

$$\zeta_{i+1} = (d_i - a_i \zeta_i - c_i \zeta_{i-1})/b_i$$

$$\beta_{i+1} = -(a_i \beta_i + c_i \beta_{i-1})/b_i$$

$$\gamma_{i+1} = -(a_i \gamma_i + c_i \gamma_{i-1})/b_i$$

4. Compute

$$A = \zeta_N/(1 + \gamma_N)$$

$$B = \beta_N/(1 + \gamma_N)$$

$$C = (d_N - c_N \zeta_{N-1})/(a_N - c_N \gamma_{N-1})$$

$$D = (b_N - c_N \beta_{N-1})/(a_N - c_N \gamma_{N-1})$$

5. Compute first and last values of the solution vector

$$u_1 = (A - C)/(B - D)$$

$$u_N = (BC - AD)/(B - D)$$

6. Compute intermediate values of the solution vector for $i = 2, 3, \ldots, N-1$

$$u_i = \zeta_i - \beta_i u_1 - \gamma_i u_N$$

The algorithm may also be applied to the solution of eqn. (4.2). In this case

$$c_1 = b_N = 0$$

and the coefficient matrix is tridiagonal. The step (6) above reduces to the following:

6a. Note that $\gamma_i = 0$ for all i and for $i = 2, \ldots, N-1$

$$u_i = \zeta_i - \beta_i u_1$$

Tang claims that this method is superior to the Thomas algorithm under certain conditions. It is clear from step (6a) above that round-off error should not accumulate in the application of this step. This algorithm requires 11 multiplications or divisions and six additions or subtractions per grid point. The reduced form of this algorithm for the tridiagonal matrix equation results in a reduction of four multiplications or divisions and two subtractions per grid point. This results in work requirement of approximately one-and-a-half times the work for Thomas' algorithm. A computer program for Tang's algorithm is included in Appendix **B**.

Another very similar algorithm for eqn. (4.5) has been presented by Evans (1971). It requires about the same amount of computer work per grid point (11 multiplications or divisions and six additions or subtractions) as Tang's algorithm. However, in the final step of the elimination process u_i is computed from u_{i+1} and u_N. Hence it does not have the same advantage as Tang's method as far as the accumulation of round-off error is concerned.

4.2.3 Solution of Symmetric Tridiagonal Matrix Equations

As discussed in the last chapter, in many practical problems it is possible to write the finite difference equation as

$$\mathbf{S}\boldsymbol{u} = \boldsymbol{d} \tag{4.21}$$

where the coefficient matrix is tridiagonal and symmetric:

$$\mathbf{S} = \begin{bmatrix} a_1 & b_1 & & & & \\ b_1 & a_2 & b_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & b_{N-2} & a_{N-1} & b_{N-1} \\ & & & & b_{N-1} & a_N \end{bmatrix} \tag{4.22}$$

For this case it is possible to write

$$\mathbf{S} = \mathbf{W}\mathbf{W}^{\mathbf{T}}$$

where $\mathbf{W}$ has the form

$$\mathbf{W} = \begin{bmatrix} w_1 & & & & & \\ q_1 & w_2 & & & & \\ & \cdot & \cdot & & & \\ & & q_{i-1} & w_i & & \\ & & & \cdot & \cdot & \\ & & & & q_{N-1} & w_N \end{bmatrix}$$

and $\mathbf{W}^{\mathbf{T}}$ is the transpose of $\mathbf{W}$.

An algorithm that takes advantage of this special form of the matrix (attributed to Choleski) may be executed by the following steps (Fox, 1964; Westlake, 1968; Wilkinson and Reinsch, 1971):

1. Set $w_1 = \sqrt{a_1}$
2. Compute for $i = 1,\ldots, N-1$

$$q_i = \frac{b_i}{w_i}$$

$$w_{i+1} = \sqrt{a_{i+1} - q_i^2}$$

3. Set

$$g_N = \frac{d_N}{w_N}$$

4. Compute for $i = N-1,\ldots, 1$

$$g_i = (d_i - q_i g_{i+1})/w_i$$

5. Set $u_1 = g_1/w_1$
6. Compute for $i = 2,\ldots, N$

$$u_i = (g_i - q_i u_{i-1})/w_i$$

Choleski's algorithm requires considerably more work (six multiplications or divisions, one square root and three additions or subtractions per grid point) than the Thomas algorithm. This is surprising at first, since we would expect to do less work for the symmetric case. We will show in Chapter 8 that symmetric decomposition as discussed above becomes attractive for band matrices of larger band-width.

4.2.4 Special Cases of Non-Unique Solution

Under certain cases the matrix equation to be solved is singular and a unique solution does not exist. This is the case for elliptic (steady-state flow) problems with Neumann boundary conditions. For this case a solution exists if $\sum d_i = 0$ and in order to make the solution unique one of the elements of $\boldsymbol{u}$ must be specified arbitrarily at some boundary point (see Exercise 4.1). The coefficient matrix may be made symmetric by the methods discussed in the previous chapter. The form of the matrix is the same as the matrix **S** given by eqn. (4.22) with elements defined by:

$$\begin{aligned} a_1 &= -b_1 \\ a_i &= -(b_i + b_{i-1}) \qquad i = 2, \ldots, N-1 \\ a_N &= -b_{N-1} \end{aligned}$$

This system may be solved by the following algorithm:

1. Let $u_N = 0$ (or any arbitrary value)
2. Compute for $i = N-1,\ldots, 1$

$$u_i = -\sum_{k=1}^{i} d_k/b_i - u_{i+1} \tag{4.23}$$

This process requires only one division and one addition per grid point (since $\sum_{i=1}^{N-1} d_i = -d_N$ and subsequent sums can be obtained by a single subtraction).

A unique solution does not exist for the elliptic case with periodic (circular) boundary conditions. The solution becomes unique if u_i is specified at any point i and the remaining unknowns are solved for, with u_i as the boundary condition at both ends. It is possible to develop a special elimination algorithm for this case as outlined in Exercise 4.2.

4.2.5 Other Special Cases

Computational work may be further reduced if eqn. (4.2) is to be solved repeatedly for different ***d*** with no change in **A**. In this case the elements of decomposition matrices **W** and **Q** defined by eqns. (4.10) to (4.12) may be saved and the Thomas algorithm modified accordingly. This process requires only three multiplications or divisions and two additions or subtractions per grid point after **W** and **Q** have been computed once.

Further reductions in the work are possible if **A** is symmetric and eqn. (4.2) is to be solved repeatedly (Cuthill and Varga, 1959). This procedure requires only two multiplications or divisions and two additions or subtractions per grid point.

Other special cases have been considered by Evans and Forrington (1963) and Bakes (1965). These are, however, of no importance for the problems considered in this book.

EXERCISES

Exercise 4.1

Consider the equation

$$\frac{\partial}{\partial x} K\left(\frac{\partial u}{\partial x}\right) = q(x) \qquad x \in (0, L)$$

with boundary conditions $(\partial u/\partial x) = 0$ *at* $x = 0$ *and* $x = L$ *for which the corresponding matrix equation is of the form* (4.21). *Find* (*a*) *the conditions for the existence of solution and* (*b*) *show how the solution could be made unique by specifying u at an arbitrary point.*

Solution Outline

(a) By successively adding the rows of the equation, find the condition for the existence of solution

$$\sum_i d_i = 0 \tag{A}$$

which is equivalent to eqn. (3.166). Show that if (A) is satisfied, all solutions of $\mathbf{S}\boldsymbol{u} = \boldsymbol{d}$ are

$$\boldsymbol{u} = \boldsymbol{u}_\mathrm{p} + \boldsymbol{c} \tag{B}$$

where $\boldsymbol{u}_\mathrm{p}$ is a particular solution and $\boldsymbol{c} = [c, c, \ldots]^\mathrm{T}$ is a constant vector. (*Hint:* show that $\boldsymbol{Sc} = \boldsymbol{d}$.)

(b) If the solution is specified at an interior point the matrix **S** becomes strongly diagonally dominant but *reducible* (Varga, 1962). (Replace the finite-difference equation for point i by $u_i = U$ and analyse the resulting matrix.) The physical meaning of reducibility is that the problem can be separated into two independent problems. The matrix **S** in this case can be reduced by eliminating the row and column i.

The reduced equation is:

$$\left[\begin{array}{c|c} \mathbf{S}_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{S}_2 \end{array}\right]\left[\begin{array}{c} \boldsymbol{u}_1 \\ \hline \boldsymbol{u}_2 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \end{array}\right]$$

i.e.,

$$\mathbf{S}_1\boldsymbol{u}_1 = \boldsymbol{g}_1$$
$$\mathbf{S}_2\boldsymbol{u}_2 = \boldsymbol{g}_2$$

where

$$\boldsymbol{u}_1 = [u_1, u_2, \ldots, u_{i-1}]^\mathrm{T}$$
$$\boldsymbol{u}_2 = [u_{i+1}, \ldots, u_N]^\mathrm{T}$$
$$g_{1j} = d_j \qquad j = 1, \ldots, i-2 \qquad g_{2j} = d_j \qquad j = i+2, \ldots, N$$
$$g_{1i-1} = d_{i-1} - b_{i-1}U \qquad g_{2i+1} = d_{i+1} - b_iU$$

and $\mathbf{S}_1$, and $\mathbf{S}_2$ are submatrices of **S**.

If the solution is specified at $i = 1$ or $i = N$, the size of matrix can be reduced to $N - 1$. The reduced matrix is diagonally dominant and irreducible and the problem has a

unique solution.

Exercise 4.2

Develop the algorithm analogous to the algorithm (4.23) for elliptic problems with periodic boundary conditions, given by $\mathbf{Au} = \boldsymbol{d}$, *where*

$$\mathbf{A} = \begin{bmatrix} a_1 & b_1 & & & & & b_N \\ b_1 & a_2 & b_2 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ b_N & & & & & b_{N-1} & a_N \end{bmatrix} \qquad \begin{aligned} a_1 &= -(b_1 + b_N) \\ a_i &= -(b_{i-1} + b_i) \\ i &= 2, \ldots, N \end{aligned}$$

Consider:

(a) *The case when* $b_1 = b_2 = \cdots = b$.

(b) *The general case.*

(c) *Computing requirements.*

Solution Outline

Adding rows will transform $\mathbf{A}$ and $\boldsymbol{d}$ into

$$\mathbf{A}' = \begin{bmatrix} a_1 & b_1 & & & & & & & b_N \\ -b_N & -b_2 & b_2 & & & & & & b_N \\ -b_N & & -b_3 & b_3 & & & & & b_N \\ \cdot & & & \cdot & \cdot & & & & \cdot \\ -b_N & & & & -b_i & b_i & & & b_N \\ \cdot & & & & & \cdot & \cdot & & \cdot \\ \cdot & & & & & & \cdot & \cdot & \cdot \\ -b_N & & & & & & & -b_{N-1} & (b_{N-1} + b_N) \\ b_N & & & & & & & b_{N-1} & a_N \end{bmatrix}$$

$$\boldsymbol{d}^1 = \begin{bmatrix} d_1 \\ d_1 + d_2 \\ \vdots \\ \sum_{j=1}^{i} d_j \\ \vdots \\ \sum_{j=1}^{N-1} d_j \\ d_N \end{bmatrix}$$

The last two rows give the existence condition $\sum_i^N d_j = 0$. Choose now

$$u_N = 0 \tag{A}$$

then $u_1,\ldots, u_{N\text{-}1}$ are obtained by solving $\mathbf{A^1}\boldsymbol{u} = \boldsymbol{d^1}$ reduced by dropping the last row and the last column.

(a) If $b_1 = b_2 = \cdots = b_N$, the algorithm is

$$u_1 = \sum_{k=1}^{N-1} \left(\sum_{j=1}^{k} d_j \right) \Big/ Nb \tag{B}$$

$$u_i = \sum_{k=i}^{N-1} \left(\sum_{j=1}^{k} d_j \right) \Big/ b - (N-i)u_1$$

(*Hint:* Sum all rows $j = 1,\ldots, N-1$ to obtain u_1, sum rows $j = 1,\ldots, i$ to get u_i.)

(b) For the case when all b values are different, multiply each row by b_1/b_i, then proceed as in (a) above.

(c) With suitable arrangement of calculations, (B) requires one multiplication and three additions per unknown.

A more general system of this type is considered by Evans (1971a).

CHAPTER 5

MULTIPHASE FLOW IN ONE DIMENSION

5.1 INTRODUCTION

This chapter is the first dealing with multiphase flow. In comparison with single-phase flow, simulation of multiphase flow requires more powerful techniques, because it deals with a system of *coupled* nonlinear equations. We will introduce here basic solution techniques that are currently in use, as well as special techniques used in problems that are difficult to solve by standard techniques. Although such problems (e.g., coning and gas percolation) are more typical of multidimensional systems, it is easier to discuss techniques for handling them here.

Since the first two-phase paper by West *et al*. (1954), many papers on simulation of multiphase flow have been published. Progress in this area has been reviewed by Richardson and Stone (1973). In Sections 5.1 and 5.2, the two basic methods for solving multiphase equations are introduced: simultaneous solution method (SS), and implicit pressure-explicit saturation method (IMPES). Several methods for the treatment of nonlinearities are discussed in Section 5.5. A relatively new method known as the sequential method (SEQ) is introduced in Section 5.6.

When dealing with two-phase flow, we will denote the wetting and non-wetting phases by 'w' and 'n' and write eqns. (2.62) to (2.65) as:

$$\frac{\partial}{\partial x}\left[\lambda_l\left(\frac{\partial p_l}{\partial x} - \gamma_l \frac{\partial z}{\partial x}\right)\right] = \frac{\partial}{\partial t}\left(\phi \frac{S_l}{B_l}\right) + q_l \qquad l = \mathrm{w}, \mathrm{n} \tag{5.1a}$$

$$P_c = p_{\mathrm{n}} - p_{\mathrm{w}} \qquad S_{\mathrm{w}} + S_{\mathrm{n}} = 1 \tag{5.1b}$$

For three-phase flow, we will write the equations as:

$$\left.\begin{aligned}
&\frac{\partial}{\partial x}\left[\lambda_w\left(\frac{\partial p_w}{\partial x}-\gamma_w\frac{\partial z}{\partial x}\right)\right]=\frac{\partial}{\partial t}\left(\phi\frac{S_w}{B_w}\right)+q_w\\
&\frac{\partial}{\partial x}\left[\lambda_o\left(\frac{\partial p_o}{\partial x}-\gamma_o\frac{\partial z}{\partial x}\right)\right]=\frac{\partial}{\partial t}\left(\phi\frac{(1-S_w-S_g)}{B_o}\right)+q_o\\
&\frac{\partial}{\partial x}\left[R_s\lambda_o\left(\frac{\partial p_o}{\partial x}-\gamma_o\frac{\partial z}{\partial x}\right)\right]+\frac{\partial}{\partial x}\left[\lambda_g\left(\frac{\partial p_g}{\partial x}-\gamma_g\frac{\partial z}{\partial x}\right)\right]\\
&\qquad=\frac{\partial}{\partial t}\left[\phi R_s\frac{(1-S_w-S_g)}{B_o}+\phi\frac{S_g}{B_g}\right]+R_sq_o+q_g
\end{aligned}\right\} \tag{5.2a}$$

$$p_o-p_w=P_{cow}\qquad p_g-p_o=P_{cog} \tag{5.2b}$$

Since most of this chapter will deal with approximations at a given point in space, we will on occasion omit the spatial subscript *i*.

5.2 THE SIMULTANEOUS SOLUTION (SS) METHOD

The essence of this method is the writing of saturation derivatives on the right side of eqns. (5.1a) or eqns. (5.2a) in terms of pressure derivatives and the solution of the resulting equations for pressures. It was first proposed by Douglas *et al.* (1959) and later extended and further analysed by several investigators (Coats *et al.*, 1967; Coats, 1968; Sheffield, 1969).

5.2.1 The SS Method for Two-Phase Flow

The choice of time approximation is crucial for coupled equations. Since eqns. (5.1a) are of parabolic form, it is natural to try techniques of Chapter 3 (Section 3.3). However, such methods rarely work when applied to systems of coupled nonlinear equations of this type. As discussed by Peaceman (1967), the explicit method as well as the Peaceman–Rachford ADI method will always be unstable for problems of this type. One reliable method is the backward difference approximation. It has been shown to be stable for problems of this type by Douglas (1960) and it is the one most often used for reservoir simulations.

Let us therefore discretise the right side of eqn. (5.1a) by the backward difference approximation. The concept of time approximation that conserves mass was discussed

in Chapter 3 (Section 3.7.1). It is extended here to multiphase flow by the following definition of time difference

$$\Delta_t\left(\phi\frac{S_l}{B_l}\right) \equiv \left(\phi\frac{S_l}{B_l}\right)^{n+1} - \left(\phi\frac{S_l}{B_l}\right)^{n} \tag{5.3}$$

so that

$$\frac{\partial}{\partial t}\left(\phi\frac{S_l}{B_l}\right) \simeq \frac{1}{\Delta t}\Delta_t\left(\phi\frac{S_l}{B_l}\right) \tag{5.4}$$

We can now write the finite-difference approximations to eqn. (5.1a) as

$$[\Delta \mathrm{T}_l\,\Delta(p_l^{n+1} - \gamma_l z)]_i = \frac{1}{\Delta t}\Delta_t\left(V_p\frac{S_l}{B_l}\right)_i + Q_{li} \qquad l = \mathrm{w}, \mathrm{n} \tag{5.5}$$

where

$$\begin{aligned}\Delta \mathrm{T}\,\Delta(p - \gamma z) &\equiv \Delta \mathrm{T}\,\Delta p - \Delta \mathrm{T}\gamma\,\Delta z \\ &\equiv \mathrm{T}_{i+1/2}[p_{i+1} - p_i - \gamma_{i+1/2}(z_{i+1} - z_i)] \\ &\quad + \mathrm{T}_{i-1/2}[p_{i-1} - p_i - \gamma_{i-1/2}(z_{i-1} - z_i)]\end{aligned} \tag{5.6}$$

$V_{pi} = \phi_i A_i \triangle x_i = V_i\phi_i$, $\mathrm{T}_{i+1/2} = \lambda_{i+1/2}(A_{i+1/2}/\triangle x_{i+1/2})$ and Q_{li} are the block injection/production rates. These approximations are direct extensions of eqns. (3.125) and (3.126) for single-phase flow.

Expansion of the time derivative approximations. The essential step which leads to the SS formulation is to expand the right side of eqn. (5.5) in terms of p_w and p_n. The operator Δ_t can be expanded as

$$\begin{aligned}\Delta_t\left(\phi\frac{S_l}{B_l}\right) &= S_l^n\,\Delta_t\left(\frac{\phi}{B_l}\right) + \left(\frac{\phi}{B_l}\right)^{n+1}\Delta_t S_l \\ &= S_l^n\phi^n\,\Delta_t\left(\frac{1}{B_l}\right) + \left(\frac{\phi}{B_l}\right)^{n+1}\Delta_t S_l + S_l^n\frac{1}{B_l^{n+1}}\Delta_t\phi\end{aligned} \tag{5.7}$$

We define derivatives (slopes of nonlinearities) as follows:

$$b_l' = \left(\frac{1}{B_l}\right)' = \frac{\mathrm{d}\left(\frac{1}{B_l}\right)}{\mathrm{d}p_l} \tag{5.8}$$

$$S_l' = \mathrm{d}S_l/\mathrm{d}P_c \qquad l = \mathrm{w}, \mathrm{n}$$

$$\phi' = \mathrm{d}\phi/\mathrm{d}p \; (= \phi^\circ c_\mathrm{R})$$

If we assume that porosity depends on $p \cong \frac{1}{2}(p_w + p_n)$, we can write $\triangle_t \phi = \phi' \frac{1}{2} \; (\triangle_t p_w + \triangle_t p_n)$. Then eqn. (5.7) can be expressed as

$$\Delta_t\left(\phi \frac{S_l}{B_l}\right) = (\phi S_l)^n b_l' \, \Delta_t p_l + (\phi b_l)^{n+1} S_l'(\Delta_t p_n - \Delta_t p_\mathrm{w}) + \tfrac{1}{2} b_l^{n+1} S_l^n \phi'(\Delta_t p_\mathrm{n} + \Delta_t p_\mathrm{w}) \qquad l = \mathrm{w}, \mathrm{n} \tag{5.9}$$

Note that so far we have not assigned any time level to the derivatives S_l', etc. Equation (5.9) will conserve mass only if the expansions of terms like $\Delta_t S$ are exact, i.e.,

$$S_l' \, \Delta_t P_c \equiv S_l^{n+1} - S_l^n$$

This can be generally satisfied only if the derivative S_l' is defined as a *chord slope* between S^n and S^{n+1} (Peaceman, 1967; Coats, 1968):

$$S_l' = \frac{S_l^{n+1} - S_l^n}{P_c^{n+1} - P_c^n} \tag{5.10}$$

With the exception of capillary pressure that is a linear function of saturation, eqn. (5.10) defines an implicit coefficient and thus requires iteration. Similar implicit coefficients result from b_l' and ϕ'. Even if P_c and b_l were linear functions, expansion (5.9) will still have implicit coefficients because $(\phi b_l)^{n+1}$ in the second term and b^{n+1} in the third term are dated at the $n + 1$ level.

Matrix form of the SS equations. With the expansions given by eqn. (5.9) we can now write eqn. (5.5) as:

$$\begin{aligned} [\Delta \mathrm{T}_\mathrm{w}(\Delta p_\mathrm{w} - \gamma_\mathrm{w} \Delta z)]_i^{n+1} &= [d_{11} \, \Delta_t p_\mathrm{w} + d_{12} \, \Delta_t p_n]_i + Q_{\mathrm{w}\,i} \\ [\Delta \mathrm{T}_\mathrm{n}(\Delta p_\mathrm{n} - \gamma_\mathrm{n} \Delta z)]_i^{n+1} &= [d_{21} \, \Delta_t p_\mathrm{w} + d_{22} \, \Delta_t p_\mathrm{n}]_i + Q_{n\,i} \end{aligned} \tag{5.11}$$

where the coefficients d_{kl} are easily found from eqn. (5.9) (recognising $S_n' = -S_w'$):

$$d_{11} = \frac{V}{\Delta t}[(\phi S_w)^n b'_w - (\phi b_w)^{n+1} S'_w + \tfrac{1}{2} b_w^{n+1} S_w^n \phi'] \tag{5.12}$$

$$d_{12} = \frac{V}{\Delta t}[(\phi b_w)^{n+1} S'_w + \tfrac{1}{2} b_w^{n+1} S_w^n \phi']$$

$$d_{21} = \frac{V}{\Delta t}[(\phi b_n)^{n+1} S'_w + \tfrac{1}{2} b_n^{n+1} (1 - S_w^n)\phi']$$

$$d_{22} = \frac{V}{\Delta t}[(\phi(1 - S_w))^n b'_n - (\phi b_n)^{n+1} S'_w + \tfrac{1}{2} b_n^{n+1} (1 - S_w^n)\phi']$$

There are two difference equations (eqns. (5.11)) for each grid point which must be solved simultaneously, hence the name 'SS' method.

Let us order the unknowns in a vector:

$$\boldsymbol{P} = [p_{1w}, p_{1n}, \ldots, p_{iw}, p_{in}, \ldots, p_{Nw}, p_{Nn}]^T$$

The finite-difference equations for all grid points may now be written in a matrix form as

$$\mathbf{T}\boldsymbol{P}^{n+1} = \mathbf{D}(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) + \boldsymbol{G} + \boldsymbol{Q} = \mathbf{D}\,\Delta_t \boldsymbol{P} + \boldsymbol{G} + \boldsymbol{Q} \tag{5.13}$$

where **T** is the transmissibility matrix, **D** is the accumulation matrix, ***G*** is the vector of gravity terms (which we have assumed can be expressed explicitly at the time level ***n***), and ***Q*** is the source vector.

Since eqn. (5.13) is typical for multiphase flow, we will show its structure in detail and develop notation for block-structured matrices.

The *i*th 'block row' of matrix **T** is

Diagonal Block

$T_{w_{i-1/2}}$	0	$-(T_{w_{i-1/2}} + T_{w_{i+1/2}})$	0	$T_{w_{i+1/2}}$	0
0	$T_{n_{i-1/2}}$	0	$-(T_{n_{i-1/2}} + T_{n_{i+1/2}})$	0	$T_{n_{i+1/2}}$

The *i*th block of **D** is a 2 × 2 matrix

$$\mathbf{D}_i = \begin{bmatrix} d_{11_i} & d_{12_i} \\ d_{21_i} & d_{22_i} \end{bmatrix} \tag{5.14}$$

and the ith partitions of $\boldsymbol{G}$ and $\boldsymbol{Q}$ are

$$\boldsymbol{G}_i = \begin{bmatrix} (\Delta \mathrm{T}_w \gamma_w \Delta z)_i \\ (\Delta \mathrm{T}_n \gamma_n \Delta z)_i \end{bmatrix} \qquad \boldsymbol{Q}_i = \begin{bmatrix} Q_{w_i} \\ Q_{n_i} \end{bmatrix} \tag{5.15}$$

If we define (in addition to $\mathbf{D}_i$) 2×2 matrices

$$\mathbf{T}_{i-1/2} = \begin{bmatrix} \mathrm{T}_{w_{i-1/2}} & 0 \\ 0 & \mathrm{T}_{n_{i-1/2}} \end{bmatrix} \qquad \mathbf{T}_i = -(\mathbf{T}_{i-1/2} + \mathbf{T}_{i+1/2}) \tag{5.16}$$

and two-element vectors

$$\boldsymbol{P}_i = \begin{bmatrix} p_{w_i} \\ p_{n_i} \end{bmatrix} \qquad \Delta_t \boldsymbol{P}_i^n = \begin{bmatrix} \Delta_t p_{w_i}^n \\ \Delta_t p_{n_i}^n \end{bmatrix} \equiv \begin{bmatrix} p_{w_i}^{n+1} - p_{w_i}^n \\ p_{n_i}^{n+1} - p_{n_i}^n \end{bmatrix}$$

then, eqn. (5.13) can be written in expanded block form as

$$\begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_{1+1/2} & & & & & \\ \cdot & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \mathbf{T}_{i-1/2} & \mathbf{T}_i & \mathbf{T}_{i+1/2} & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & & \mathbf{T}_{N-1/2} & \mathbf{T}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 \\ \cdot \\ \cdot \\ \boldsymbol{P}_i \\ \cdot \\ \cdot \\ \boldsymbol{P}_N \end{bmatrix}^{n+1} \tag{5.17}$$

$$= \begin{bmatrix} \mathbf{D}_1 & & & & \\ & \cdot & & & \\ & & \cdot & & \\ & & & \mathbf{D}_i & \\ & & & & \cdot \\ & & & & & \cdot \\ & & & & & & \mathbf{D}_N \end{bmatrix} \begin{bmatrix} \Delta_t \mathbf{P}_1 \\ \cdot \\ \cdot \\ \Delta_t \mathbf{P}_i \\ \cdot \\ \cdot \\ \Delta_t \mathbf{P}_N \end{bmatrix}^n + \begin{bmatrix} \boldsymbol{G}_1 \\ \cdot \\ \cdot \\ \boldsymbol{G}_i \\ \cdot \\ \cdot \\ \boldsymbol{G}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{Q}_1 \\ \cdot \\ \cdot \\ \boldsymbol{Q}_i \\ \cdot \\ \cdot \\ \boldsymbol{Q}_N \end{bmatrix}$$

The above form is useful because many matrix algorithms can be readily extended to block-structured matrices by formal substitution of matrix operations for arithmetic operations (Chapter 6).

Equation (5.13) is sometimes written in a residual form, which is more convenient from the computational point of view, especially in connection with iterative methods. Let us define for any $\boldsymbol{P}^k$ an $\boldsymbol{R}^k$ by

$$\boldsymbol{R}^k = \mathbf{T}\boldsymbol{P}^k - \mathbf{D}(\boldsymbol{P}^k - \boldsymbol{P}^n) - \boldsymbol{G} \tag{5.18}$$

Then the solution (5.13) satisfies

$$\boldsymbol{R}^{n+1} = \boldsymbol{Q}$$

Using $\boldsymbol{R}^n$, we can write eqn. (5.13) as

$$(\mathbf{T} - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\boldsymbol{R}^n + \boldsymbol{Q} \tag{5.19}$$

Often, we have some approximation $\boldsymbol{P}^k$ to $\boldsymbol{P}^{n+1}$ such as the last iteration vector; then we can write

$$(\mathbf{T} - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{k}) = -\boldsymbol{R}^{k} + \boldsymbol{Q} \tag{5.20}$$

(Some authors include $\boldsymbol{Q}$ in the definition of $\boldsymbol{R}$ such that $\boldsymbol{R}^{n+1} = \boldsymbol{0}$.)

Remarks on the SS method.

(a) The SS method as presented here *requires non-zero capillary pressure* since the equations for p_{w} and p_{n} are coupled through S'_w. As the capillary pressure slope decreases, the matrix **D** becomes dominant and the system of equations becomes singular (Exercise 5.1). Therefore, if one wishes to simulate the case of zero capillary pressure, a small 'dummy' P_c curve (best choice is linear) must be used. Fortunately, the value of $\mathrm{d}P_c/\mathrm{d}S$ which is necessary for the SS solution to remain meaningful is small enough so that it does not affect the answers (Coats, 1968). A variant of the SS method which is not restricted to non-zero P_c is given in the next section.

(b) After the time step is completed, S_{w} must be updated. This step involves the treatment of the nonlinearities due to capillary pressure which will be discussed in Section 5.5.

(c) Since the number of unknowns is $N \times$ the number of phases, the SS method becomes expensive for multidimensional problems. The real value of the SS approach is in conjunction with implicit treatment of transmissibilities (Section 5.5).

5.2.2 Extension of the SS Method to Three-Phase Flow

The expansions of the right-hand side for the oil and water equations from eqns. (5.2) are

$$\begin{aligned}\Delta_t\left(\phi\frac{S_{\mathrm{o}}}{B_{\mathrm{o}}}\right) = {} & \phi^n(1 - S_{\mathrm{w}} - S_{\mathrm{g}})^n b'_{\mathrm{o}}\,\Delta_t p_{\mathrm{o}} + (\phi b_{\mathrm{o}})^{n+1}[-S'_{\mathrm{w}}(\Delta_t p_{\mathrm{o}} - \Delta_t p_{\mathrm{w}}) \\ & - S'_{\mathrm{g}}(\Delta_t p_{\mathrm{g}} - \Delta_t p_{\mathrm{o}})] + b_{\mathrm{o}}^{n+1}(1 - S_{\mathrm{w}} - S_{\mathrm{g}})^n \phi'\,\Delta_t p_{\mathrm{w}}\end{aligned} \tag{5.21}$$

$$\Delta_t\left(\phi\frac{S_{\mathrm{w}}}{B_{\mathrm{w}}}\right) = \phi^n S_{\mathrm{w}}^n b'_{\mathrm{w}}\,\Delta_t p_{\mathrm{w}} + (\phi b_{\mathrm{w}})^{n+1} S'_{\mathrm{w}}(\Delta_t p_{\mathrm{o}} - \Delta_t p_{\mathrm{w}}) + b_{\mathrm{w}}^{n+1} S_{\mathrm{w}}^n \phi'\,\Delta_t p_{\mathrm{w}} \tag{5.22}$$

where we have now assumed that ϕ depends on p_{w}. For the gas equation, the operator satisfying material balance is

$$\Delta_t\left(\phi\frac{S_g}{B_g}+R_s\phi\frac{S_o}{B_o}\right)=\Delta_t\left(\phi\frac{S_g}{B_g}\right)+R_s^n\,\Delta_t\left(\phi\frac{S_o}{B_o}\right)+\left(\phi\frac{S_o}{B_o}\right)^{n+1}\Delta_t R_s \tag{5.23}$$

The first term is

$$\Delta_t\left(\phi\frac{S_g}{B_g}\right)=\phi^n S_g^n b_g'\,\Delta_t p_g+(\phi b_g)^{n+1}S_g'(\Delta_t p_g-\Delta_t p_o)+b_g^{n+1}S_g^n\phi'\,\Delta_t p_w \tag{5.24}$$

The second term is given by eqn. (5.21) and $\Delta_t R_s$ is expressed as

$$\Delta_t R_s=R_s'\,\Delta_t p_o \qquad \text{where} \qquad R_s'=\frac{R_s^{n+1}-R_s^n}{p_o^{n+1}-p_o^n}$$

It is equally appropriate to expand the second term on the left side of eqn. (5.23) as

$$\Delta_t\left(R_s\phi\frac{S_o}{B_o}\right)=(R_s b_o)^n\,\Delta_t(\phi S_o)+(\phi S_o)^{n+1}\,\Delta_t(R_s b_o)$$

using the definition

$$(R_s b_o)'=\frac{(R_s b_o)^{n+1}-(R_s b_o)^n}{p_o^{n+1}-p_o^n}$$

The difference equations for oil and water are again given by eqn. (5.5), where l = o, w and eqn. (5.21) or (5.22) as appropriate is used for the right-hand side. The difference approximation for the gas equation may be written as

$$\begin{aligned}\Delta[\mathrm{T}_g(\Delta p_g^{n+1}-\gamma_g\,\Delta z)]_i+\Delta[R_s\mathrm{T}_o(\Delta p_o^{n+1}-\gamma_o\,\Delta z)]_i \\ =\frac{1}{\Delta t}\Delta_t\left\{V_p\left[\frac{S_g}{B_g}+R_s\frac{S_o}{B_o}\right]\right\}+(R_sQ_o)_i+Q_{g_i}\end{aligned} \tag{5.25}$$

where

$$\begin{aligned}\Delta[R_s\mathrm{T}_o(\Delta p_o-\gamma_o\,\Delta z)]_i=(R_s\mathrm{T}_o)_{i+1/2}[p_{o_{i+1}}-p_{o_i}-\gamma_{o_{i+1/2}}(z_{i+1}-z_i)] \\ +(R_s\mathrm{T}_o)_{i-1/2}[\mathrm{p}_{o_{i-1}}-p_{o_i}-\gamma_{o_{i-1/2}}(z_{i-1}-z_i)]\end{aligned} \tag{5.26}$$

An alternative formulation of the gas equation is obtained if the oil equation is multiplied by R_s and subtracted from the gas equation. This can be done with the differential or difference equation with the same result (see Exercise 5.2).

5.2.3 Other Formulations of the SS Method

For brevity, we will only consider two-phase flow.

Formulation without explicit gravity terms. The definition of pseudopotentials Φ_w, Φ_n (see Section 2.4.3 of Chapter 2) by

$$\Phi_l = \int_{p^0}^{p} \frac{dp}{\gamma_l} - z \qquad l = w, n \tag{5.27}$$

transforms eqn. (5.1a) into

$$\frac{\partial}{\partial x}\left(\lambda_l \gamma_l \frac{\partial \Phi_l}{\partial x}\right) = \frac{\partial}{\partial t}\left(\phi \frac{S_l}{B_l}\right) + q_l \qquad l = w, n \tag{5.28}$$

which may be approximated by the following difference equations:

$$[\Delta T_l \, \Delta \Phi_l^{n+1}]_i = \frac{1}{\Delta t} \Delta_t \left(V_p \frac{S_l}{B_l}\right)_i + Q_{l_i} \qquad l = w, n \tag{5.29}$$

where

$$T_{l_{i+1/2}} = (\gamma_l \lambda_l)_{i+1/2} \frac{A_{i+1/2}}{\Delta x_{i+1/2}}$$

The right-hand side must be expanded in terms of Φ_l values. Since $\triangle_t \Phi_l = \triangle_t p_l / \gamma_l$ where γ_l is the average γ over the time interval n to $n+1$, we get

$$\begin{aligned} \Delta_t\left(\phi \frac{S_l}{B_l}\right) = {} & \bar{\gamma}_l [(\phi S_l)^n b_l' + b_l^{n+1} S_l^n \phi'] \, \Delta_t \Phi_l \\ & + \bar{\gamma}_l (\phi b_l)^{n+1} S_l' (\Delta_t \Phi_n - \Delta_t \Phi_w) \qquad l = w, n \end{aligned} \tag{5.30}$$

We have assumed here that

$$\Delta_t \phi \simeq \phi' \, \Delta_t p_o \simeq \phi' \, \Delta_t p_w$$

which is a good approximation if P_c is small compared to changes in reservoir pressure. In matrix form, this formulation becomes

$$\mathbf{T}\Phi^{n+1} = \mathbf{D}(\Phi^{n+1} - \Phi^n) + \mathbf{Q}$$

where matrix **D** has elements defined by eqn. (5.30).

The conversion between Φ_l and p_l is best accomplished by establishing the correspondence

$$\Phi_l \leftrightarrow \int \frac{\mathrm{d}p_l}{\gamma_l} = \hat{p}_l(p_l)$$

in tabular form, since for any z, $\Phi_l = \hat{p}_l\,(p_l) - z$.

Formulation in terms of p, S. Let us choose p_n, S_w for the dependent variables. Equations (5.1) are

$$\frac{\partial}{\partial x}\left[\lambda_w\left(\frac{\partial p_n}{\partial x} - \gamma_w \frac{\partial z}{\partial x}\right)\right] - \frac{\partial}{\partial x}\left[\lambda_w \frac{\partial P_c}{\partial x}\right] = \frac{\partial}{\partial t}\left(\phi \frac{S_w}{B_w}\right) + q_w \tag{5.31}$$

$$\frac{\partial}{\partial x}\left[\lambda_n\left(\frac{\partial p_n}{\partial x} - \gamma_n \frac{\partial z}{\partial x}\right)\right] = \frac{\partial}{\partial t}\left(\phi \frac{(1 - S_w)}{B_n}\right) + q_n$$

Now expand the right-hand sides of the above equations with the assumption that $\phi^{n+1} = \phi^n + \phi' \triangle_t p_n$ and by using the relation

$$\Delta_t p_w = \Delta_t p_n - \Delta_t P_c = \Delta_t p_n - P_c' \Delta_t S_w$$

Where

$$P_c' = \mathrm{d}P_c/\mathrm{d}S_w$$

Then

$$\Delta_t\left(\phi \frac{S_w}{B_w}\right) = (\phi b_w)^{n+1} \Delta_t S_w + (\phi S_w)^n b_w' [\Delta_t p_n - P_c' \Delta_t S_w] + S_w^n b_w^{n+1} \phi' \Delta_t p_n$$

$$\Delta_t\left(\phi \frac{S_n}{B_n}\right) = -(\phi b_n)^{n+1} \Delta_t S_w + (1 - S_w^n)(\phi^n b_n' + \phi' b_n^{n+1}) \Delta_t p_n$$

Similarly the term $\lambda_w(\partial P_c/\partial x)$ can be expressed as $\lambda_w P_c{}'(\partial S_w/\partial x)$. If we order the unknowns as indicated in the following vector

$$\boldsymbol{P} = [p_{n_1}, S_{w_1}, \ldots, p_{n_i}, S_{w_i}, \ldots, p_{n_N}, S_{w_N}]^T$$

we can write the backward finite-difference approximation to eqn. (5.31) again in the form of eqn. (5.13).

The matrix **D** has now blocks $\mathbf{D}_i$ with elements

$$d_{11} = \frac{V}{\Delta t}[S_w^n(\phi^n b'_w + \phi' b_w^{n+1})] \tag{5.32}$$

$$d_{12} = \frac{V}{\Delta t}[(\phi b_w)^{n+1} - (\phi S_w)^n b'_w P'_c]$$

$$d_{21} = \frac{V}{\Delta t}[(1 - S_w^n)(\phi^n b'_n + \phi' b_n^{n+1})]$$

$$d_{22} = \frac{V}{\Delta t}[-(\phi b_n)^{n+1}]$$

and the matrix **T** has blocks

$$\mathbf{T}_{i+1/2} = \begin{bmatrix} \mathrm{T}_{w_{j+1/2}} & -(\mathrm{T}_w P'_c)_{j+1/2} \\ \mathrm{T}_{n_{j+1/2}} & 0 \end{bmatrix} \qquad \mathbf{T}_i = -(\mathbf{T}_{i+1/2} + \mathbf{T}_{i-1/2}) \tag{5.33}$$

(Note the asymmetric structure of **T** matrix as opposed to the symmetric form of eqn. (5.16).) This formulation *is applicable also for* $P_c \equiv 0$, in which case $P'_c \equiv 0$. Equivalent formulations can be written in any other pair of p and S or equivalent variables (Exercise 5.3 and 5.4).

5.3 THE IMPLICIT PRESSURE–EXPLICIT SATURATION (IMPES) METHOD

This method originated in the works of Sheldon *et al.* (1959), and Stone and Gardner (1961). Its basic idea is to obtain a single pressure equation by a combination of the flow equations. After the pressure has been advanced in time, the saturations are updated explicitly. A similar numerical procedure has been developed also for Navier–Stokes equations where it is called the 'primitive variable' method (Fox and Deardoff, 1972). The IMPES method was also proposed by Soviet authors (Danilov *et al.*, 1968).

We will first give the standard derivation of three-phase IMPES equations, as found in Breitenbach *et al.* (1969) and Coats (1968). Following this, some variations of the method will be presented.

5.3.1 IMPES Method for Three-Phase Flow

The finite-difference equations, discretising eqn. (5.1) can be written in terms of p_o and saturations as

$$\Delta[T_w(\Delta p_o - \Delta P_{cow} - \gamma_w \Delta z)] = C_{1p}\Delta_t p_w + \sum_l C_{1l}\Delta_t S_l + Q_w$$

$$\Delta[T_o(\Delta p_o - \gamma_o \Delta z)] = C_{2p}\Delta_t p_o + \sum_l C_{2l}\Delta_t S_l + Q_o$$

$$\Delta[T_g(\Delta p_o + \Delta P_{cog} - \gamma_g \Delta z)] + \Delta[R_s T_o(\Delta p_o - \gamma_o \Delta z)]$$
$$= C_{3p}\Delta_t p_g + \sum_l C_{3l}\Delta_t S_l + R_s Q_o + Q_g$$

The basic assumption of the IMPES method is that the *capillary pressure in the flow terms on the left side of the equations does not change over a time step*. Then the terms involving ΔP_{cow} and ΔP_{cog} can be evaluated explicitly at the old (nth) time level and also $\Delta_t p_w = \Delta_t p_o = \Delta_t p_g$. We can therefore denote p_o by p and write

$$\Delta[T_w(\Delta p^{n+1} - \gamma_w \Delta z - \Delta P^n_{cow})] = C_{1p}\Delta_t p + C_{1w}\Delta_t S_w + Q_w \qquad (5.34)$$

$$\Delta[T_o(\Delta p^{n+1} - \gamma_o \Delta z)] = C_{2p}\Delta_t p + C_{2o}\Delta_t S_o + Q_o$$

$$\Delta[T_g(\Delta p^{n+1} - \gamma_g \Delta z + \Delta P^n_{cog})] + \Delta[R_s T_o(\Delta p^{n+1} - \gamma_o \Delta z)]$$
$$= C_{3p}\Delta_t p + C_{3o}\Delta_t S_o + C_{3g}\Delta_t S_g + R_s Q_o + Q_g$$

where the coefficients C are found from expansions (5.7), etc.:

$$C_{1p} = \frac{V}{\Delta t}[(S_w\phi)^n b'_w + S^n_w b^{n+1}_w \phi'] \tag{5.35}$$

$$C_{1w} = \frac{V}{\Delta t}(\phi b_w)^{n+1}$$

$$C_{2p} = \frac{V}{\Delta t}[(S_o\phi)^n b'_o + S^n_o b^{n+1}_o \phi'] \Big|$$

$$C_{2o} = \frac{V}{\Delta t}(\phi b_o)^{n+1}$$

$$C_{3p} = \frac{V}{\Delta t}[R^n_s(S^n_o \phi^n b'_o + S^n_o b^{n+1}_o \phi') + S^n_g \phi^n b'_g + S^n_g b^{n+1}_g \phi' + (\phi S_o b_o)^{n+1} R'_s]$$

$$C_{3o} = \frac{V}{\Delta t}[R^n_s(\phi b_o)^{n+1}]$$

$$C_{3g} = \frac{V}{\Delta t}(\phi b_g)^{n+1}$$

We now wish to combine the three equations (5.34) in such a way that all terms with $\Delta_t S_l$ disappear. This is achieved by multiplying the water equation by A, gas equation by B and adding all three equations. The right side of the resulting equation is:

$$(AC_{1p} + C_{2p} + BC_{3p})\Delta_t p + (-AC_{1w} + C_{2o} + BC_{3o})\Delta_t S_o + (-AC_{1w} + BC_{3g})\Delta_t S_g$$

and A and B are found from

$$-AC_{1w} + C_{2o} + BC_{3o} = 0$$
$$-AC_{1w} + BC_{3g} = 0$$

which yields

$$B = C_{2o}/(C_{3g} - C_{3o}) \tag{5.36}$$
$$A = BC_{3g}/C_{1w}$$

Therefore the pressure equation will be

$$\begin{aligned}
&\Delta[\mathrm{T_o}(\Delta p^{n+1} - \gamma_\mathrm{o}\,\Delta z)]_i + A_i\,\Delta[\mathrm{T_w}(\Delta p^{n+1} - \gamma_\mathrm{w}\,\Delta z)]_i \\
&\quad + B_i\,\Delta[\mathrm{T_o}R_\mathrm{s}(\Delta p^{n+1} - \gamma_\mathrm{o}\,\Delta z) + \mathrm{T_g}(\Delta p^{n+1} - \gamma_\mathrm{g}\,\Delta z)]_i \\
&= (C_{2p} + AC_{1p} + BC_{3p})_i\,\Delta_t p + A_i\,\Delta(\mathrm{T_w}\,\Delta P^n_{\mathrm{cow}})_i \\
&\quad - B_i\,\Delta(\mathrm{T_g}\,\Delta P^n_{\mathrm{cog}})_i + Q_\mathrm{o} + A_i Q_{\mathrm{w}\,i} + B_i(R_\mathrm{s}Q_\mathrm{o} + Q_\mathrm{g})_i
\end{aligned} \tag{5.37}$$

This is a finite-difference equation of the type obtained from a single parabolic equation and can be written as

$$\mathbf{T}\boldsymbol{P}^{n+1} = \mathbf{D}(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) + \boldsymbol{G} + \boldsymbol{Q}$$

where **T** is a tridiagonal matrix while **D** is a diagonal matrix. In this case the vector ***G*** includes gravity *and* capillary terms.

After the pressure solution is obtained, the saturations are explicitly updated by substituting the results in the first two equations (5.34). When S_l^{n+1} are known, new capillary pressures P_{cow}^{n+1} and P_{cog}^{n+1} are calculated, which are explicitly used at the next time step.

Like for the SS method, many of the coefficients on the right side are at the unknown level of time and iteration is necessary. Note that the multiplication factors A and B must also be updated during iteration.

It is easy to derive the special cases for two-phase flow.

5.3.2 Other Derivations of the IMPES Method

Different variables. Any SS formulation can be used as a starting point for IMPES. The coefficients A and B will be slightly different depending on the choice of dependent variables. For example, consider the SS formulation in p and S for two-phase flow (Section 5.2.3):

$$\begin{aligned}
\Delta[\mathrm{T_w}(\Delta p - \gamma_\mathrm{w}\,\Delta z - \Delta P_c)^{n+1}] &= d_{11}\,\Delta_t p + d_{12}\,\Delta_t S_\mathrm{w} + Q_\mathrm{w} \\
\Delta[\mathrm{T_n}(\Delta p - \gamma_\mathrm{n}\,\Delta z)^{n+1}] &= d_{21}\,\Delta_t p + d_{22}\,\Delta_t S_\mathrm{w} + Q_n
\end{aligned} \tag{5.38}$$

Multiply the first equation by A, add the equations and obtain an A that eliminates the $\Delta_t \mathrm{S_w}$ terms. This gives $A = -d_{22}/d_{12}$ and inspection of eqn. (5.32) shows that d_{12} will always be positive as long as $P_c' \leq 0$, so that the process can be carried out.

The IMPES solution now consists of two steps:

(a) With P_c evaluated explicitly, solve $\boldsymbol{P}^{n+1}$, implicitly from the following equation:

$$-(d_{22}/d_{12})\,\Delta[\mathrm{T_w}\,\Delta p^{n+1}] + \Delta[\mathrm{T_n}\,\Delta p^{n+1}] \tag{5.39}$$
$$= [-(d_{22}/d_{12})d_{11} + d_{21}]\,\Delta_t p - (d_{22}/d_{12})Q_\mathrm{w} + Q_\mathrm{n}$$
$$-(d_{22}/d_{12})\,\Delta[\mathrm{T_w}(\gamma_\mathrm{w}\,\Delta z + \Delta P_c^n)] + \Delta \mathrm{T_n}\gamma_\mathrm{n}\,\Delta z$$

(b) Solve explicitly for $\Delta_t S_\mathrm{w}$, for example, from the water equation:

$$\Delta_t S_\mathrm{w} = (1/d_{12})[-d_{11}\,\Delta_t p - Q_\mathrm{w} + \Delta \mathrm{T_w}(\Delta p^{n+1} - \gamma_\mathrm{w}\,\Delta z - \Delta P_c^n)] \tag{5.40}$$

Direct use of the fractional flow equations. The most direct way to obtain an IMPES formulation is through the 'hyperbolic' form of the flow equations derived in Chapter 2 (Section 2.5.2). For clarity, let us consider two-phase flow with ϕ = constant and zero sources and sinks. Then the pressure equation in differential form is obtained from eqn. (2.94):

$$b_\mathrm{n}\frac{\partial}{\partial x}\left[\lambda_\mathrm{w}\left(\frac{\partial p}{\partial x} - \gamma_\mathrm{w}\frac{\partial z}{\partial x} - \frac{\partial P_c}{\partial x}\right)\right] + b_\mathrm{w}\frac{\partial}{\partial x}\left[\lambda_\mathrm{n}\left(\frac{\partial p}{\partial x} - \gamma_\mathrm{n}\frac{\partial z}{\partial x}\right)\right] \tag{5.41}$$
$$= \phi\left[b_\mathrm{n}S_\mathrm{w}\frac{\partial b_\mathrm{w}}{\partial t} + b_\mathrm{w}(1 - S_\mathrm{w})\frac{\partial b_\mathrm{n}}{\partial t}\right]$$

where $p = p_n$. We discretise the flow terms in the usual manner and approximate $\partial b/\partial t$ by $(b'/\Delta t)\,\Delta_t p$. Finally, b_n and b_w are approximated by B_n^{n+1} and B_w^{n+1}. This choice then leads to difference equations *identical* to eqn. (5.39) as shown below. The manipulations indicated above yield

$$b_\mathrm{n}^{n+1}\,\Delta[\mathrm{T_w}(\Delta p^{n+1} - \gamma_\mathrm{w}\,\Delta z - \Delta P_c^n)] + b_\mathrm{w}^{n-1}\,\Delta[\mathrm{T_n}(\Delta p^{n+1} - \gamma_\mathrm{n}\,\Delta z)] \tag{5.42}$$
$$= \frac{V_p}{\Delta t}\,[b_\mathrm{n}^{n+1}S_\mathrm{w}^n b'_\mathrm{w} + b_\mathrm{w}^{n+1}(1 - S_\mathrm{w}^n)b'_\mathrm{n}]\,\Delta_t p$$

Now for ϕ equal to a constant, we get from eqn. (5.32)

$$A = -d_{22}/d_{12} = b_\mathrm{n}^{n+1}/b_\mathrm{w}^{n+1}$$

and it is now easy to verify that eqn. (5.39) is identical with eqn. (5.42). It should be now obvious how to proceed with derivation of IMPES from three-phase equations expressed in terms of p, S_w, S_g, etc. (Exercise 5.5).

In IMPES, we do not use the saturation equation (eqn. 2.95), also called the 'fractional flow equation'. As we will see in Section 5.6, use of this equation leads to the sequential solution method.

5.4 ANALYSIS OF SS AND IMPES METHODS

In the two previous sections, we have derived the SS and IMPES methods without any reference to the treatment of nonlinearities involved or the existence of solution. In this section we will examine the properties of these methods in their basic form when all of the coefficients in the difference equations are evaluated at the old time level (lagged behind). The first convergence analysis of the SS method was presented for two-phase flow by Douglas (1960). Later, Coats (1968) analysed the stability of both SS and IMPES methods, and Sheffield (1969) discussed the existence of solution for the SS method.

5.4.1 Stability

There are two possible stability limitations which can be analysed independently.

The first comes from the explicit treatment of primary variables. Because the SS method treats all primary variables implicitly, it is in this respect unconditionally stable. However, the IMPES method treats capillary pressure explicitly and has therefore a stability limit depending on the magnitude of dP_c/dS.

The second limit results from the explicit treatment of transmissibilities, which are the strongest nonlinearities involved. Because this treatment is identical for both the SS and IMPES formulations, so is the stability limit.

For simplicity, we will treat only the two-phase case. The three-phase development is outlined in Exercise 5.6. Our analysis follows Coats (1968).

5.4.1.1 *Stability with Respect to P_c*

For this analysis, we assume that the dP_c/dS and transmissibilities are constant and for simplicity we also neglect gravity effects.

Stability of the IMPES method. Let us first consider the IMPES method for incompressible flow. Under the above assumptions eqns. (5.38) may be written as

$$\mathrm{T}_{\mathrm{w}} \Delta^2 p^{n+1} = \frac{V_p}{\Delta t} \Delta_t S_{\mathrm{w}} + \mathrm{T}_{\mathrm{w}} P_c' \Delta^2 S_{\mathrm{w}}^n + Q_{\mathrm{w}} \tag{5.43a}$$

$$\mathrm{T}_{\mathrm{n}} \Delta^2 p^{n+1} = -\frac{V_p}{\Delta t} \Delta_t S_{\mathrm{w}} + Q_n \tag{5.43b}$$

Denote now the errors in S_{w} and p by e_1 and e_2:

$$e_1^n = S_w^n - S_{w\,exact}^n \qquad e_2^n = p^n - p_{exact}^n$$

As shown in Chapter 3 (Section 3.2.3) the errors e_1 and e_2 satisfy the difference equations (5.43a) and (5.43b), provided we neglect the truncation errors. The resulting error equations are:

$$T_w \Delta^2 e_2^{n+1} = \frac{V_p}{\Delta t} \Delta_t e_1 + T_w P_c' \Delta^2 e_1^n \tag{5.44a}$$

$$T_n \Delta^2 e_2^{n+1} = -\frac{V_p}{\Delta t} \Delta_t e_1 \tag{5.44b}$$

Adding eqns. (5.44) and solving for e_2^{n+1} gives the error in the pressure equation as

$$\Delta^2 e_2^{n+1} = [T_w/(T_n + T_w)] P_c' \Delta^2 e_1^n \tag{5.45}$$

Following the Fourier series stability analysis (Chapter 3, Section 3.2.3), we seek e_l in the form

$$e_{li}^n = \xi_l^n \exp(\sqrt{-1}\alpha_l i) \qquad l = 1, 2$$

where $\alpha_l = m_l \Delta x, m_l > 0$. By suppressing the subscript i we can write

$$\Delta^2 e_l = -\gamma_l e_l$$

where

$$\gamma_l = 4 \sin^2 \frac{\alpha_l}{2}$$

Substitution of this in eqn. (5.45) gives

$$e_2^{n+1} = P_c'(T_w/T_T)(\gamma_1/\gamma_2) e_1^n \tag{5.46}$$

where $T_T = T_w + T_n$.

Substitution of e_2^{n+1} in eqn. (5.44b) yields

$$e_1^{n+1} = e_1^n \left(\frac{T_w T_n}{T_T} P_c' \gamma_1 \frac{\Delta t}{V_p} - 1 \right) = e_1^n \xi_1$$

The condition $|\xi_1| < 1$ leads to

$$\Delta t < \frac{1}{2} \frac{V_p}{|P_c'| \dfrac{T_w T_n}{T_T}} \tag{5.47}$$

Here we have used the fact that $P_c' < 0$, $\gamma_1 > 0$ $\max_i |\lambda_i| < 1$ and $\max \gamma_1 < 4$. It is easy to see that if e_1 is bounded, e_2 will also be bounded according to eqn. (5.46). The condition (5.47) must be satisfied for all points i and any saturation S_w.

Therefore, for a regular grid we obtain the following stability condition:

$$\Delta t < \tfrac{1}{2} \Delta x^2 \min_i \left(\frac{\phi_i}{k_i} \right) \frac{1}{\max\limits_{S_w} |P_c'|} \min_{i, S_w} \left(\frac{\mu_n}{k_{rn}} + \frac{\mu_w}{k_{rw}} \right) \tag{5.48}$$

Let us now investigate the rôle of compressibility on the stability of the IMPES method. For this purpose, we define $\bar{T}_l = B_l T_l$ and write the equivalent of eqn. (5.43) as

$$\bar{T}_w \Delta^2 p^{n+1} = C_{wp} \Delta_t p + C_s \Delta_t S_w + \bar{T}_w P_c' \Delta^2 S_w^n \tag{5.49a}$$

$$\bar{T}_n \Delta^2 p^{n+1} = C_{np} \Delta_t p - C_s \Delta_t S_w \tag{5.49b}$$

where

$$C_{lp} = \frac{V_p}{\Delta t} S_l b_l' B_l \qquad C_s = \frac{V_p}{\Delta t}$$

Applying the Fourier stability method, we get the equations for errors e_1, and e_2:

$$C_s e_1^{n+1} + (\bar{T}_w \gamma_2 + C_{wp}) e_2^{n+1} = (C_s + \bar{T}_w P_c' \gamma_1) e_1^n + C_{wp} e_2^n \tag{5.50a}$$

$$C_s e_1^{n+1} - (\bar{T}_n \gamma_2 + C_{np}) e_2^{n+1} = C_s e_1^n - C_{np} e_2^n \tag{5.50b}$$

In order to find the amplification factors, we must write the equations in matrix form. Thus,

$$\mathbf{A} e^{n+1} = \mathbf{C} e^n \tag{5.51}$$

and

$$\boldsymbol{e}^{n+1} = \mathbf{A}^{-1}\mathbf{C}\boldsymbol{e}^{n} = \mathbf{B}\boldsymbol{e}^{n}$$

After some matrix manipulations (Exercise 5.6), we find that

$$\mathbf{B} = \frac{1}{G_{\mathrm{T}}}\begin{bmatrix} G_{\mathrm{T}} + \dfrac{G_{\mathrm{n}}}{C_{\mathrm{s}}}C_{pc} & (G_{\mathrm{w}}C_{\mathrm{n}p} + G_{\mathrm{n}}C_{\mathrm{w}p})/C_{\mathrm{s}} \\ C_{pc} & C_{\mathrm{w}p} + C_{\mathrm{n}p} \end{bmatrix}$$

where

$$G_l = \bar{\mathrm{T}}_l\gamma_2 + C_{lp} \qquad G_{\mathrm{T}} = G_{\mathrm{w}} + G_{\mathrm{n}}$$
$$C_{pc} = \bar{\mathrm{T}}_{\mathrm{w}}P_c'\gamma_1$$

Stability requires that $\max_i |\lambda_i| < 1$ where λ_i are the eigenvalues of **B**

(Chapter 3, Section 3.2.3.3). The eigenvalues are obtained by solving $|\mathbf{B} - \lambda\mathbf{I}| = 0$ which yields

$$\lambda_{1,2} = \frac{1}{2G_{\mathrm{T}}}[X \pm \sqrt{X^2 - 4Y}] \tag{5.52}$$

where

$$X = G_{\mathrm{T}} + (C_{pc}G_{\mathrm{n}})/C_{\mathrm{s}} + C_{\mathrm{w}p} + C_{\mathrm{n}p} \tag{5.53}$$
$$Y = [G_{\mathrm{T}} + (C_{pc}G_{\mathrm{n}})/C_{\mathrm{s}}](C_{\mathrm{w}p} + C_{\mathrm{n}p}) - (G_{\mathrm{w}}C_{\mathrm{n}p} + G_{\mathrm{n}}C_{\mathrm{w}p})\,C_{pc}/C_{\mathrm{s}}$$

Because the compressibility of fluids at reservoir conditions is small (b'_w, b'_n«1), the condition C_{np}, $C_{\mathrm{w}p}$ « C_s will hold and Y « X in eqn. (5.52), since all terms of Y are multiplied by some compressibility. Therefore, the dominant eigenvalue of **B** is

$$\lambda_1 = \frac{1}{2G_{\mathrm{T}}}[X + \sqrt{X^2 - 4Y}] \simeq \frac{1}{G_{\mathrm{T}}}[X - Y/X]$$

Note that for $C_{\mathrm{w}p} = C_{\mathrm{n}p} = 0$, the above equation gives stability condition $|X/G_{\mathrm{T}}| < 1$ which leads to the previously derived condition (eqn. (5.47)). The value of (X/G_{T}) which gives this condition is negative. Therefore, when we look for the solution of

$$\left|\frac{1}{G_{\mathrm{T}}}(X - Y/X)\right| < 1$$

we can make the approximation $X/G_T \cong -1$ in the first term:

$$\left|\frac{X + Y/G_T}{G_T}\right| < 1 \tag{5.54}$$

To solve this inequality in general is difficult; however, consider a special case when only the non-wetting phase is compressible and let $C_{wp} = C_p$, $C_{np} = 0$. We can then again use the fact that the solution is close to the incompressible case and therefore

$$\frac{X}{G_T} \simeq \frac{1}{G_T}\left(G_T + \frac{G_n}{G_T} C_{pc}\right) \simeq -1$$

Then we can evaluate Y from eqn. (5.53) as

$$Y \simeq -G_T C_p - G_n(C_{pc}/C_s)C_p \simeq G_T C_p$$

and solution of eqn. (5.54) gives, for negative $X + Y/G_T$,

$$\frac{G_n}{G_T}\frac{|C_{pc}|}{C_s} < 2\left(1 + \frac{C_p}{G_T}\right)$$

which after substitution and approximation $G_l \cong T_l \gamma_2$ finally gives for the maximum value of $\gamma_2 \to 1$:

$$\Delta t \leq V_p\left(\frac{1}{2|P'_c|\dfrac{\bar{T}_w \bar{T}_n}{\bar{T}_T}} + \frac{S_w b'_w B_w}{\bar{T}_T}\right) \tag{5.55}$$

It is evident that the stability limit has increased compared to eqn. (5.47), but the improvement is in most cases negligible.

Stability analysis for three-phase flow follows the same lines but becomes somewhat more complicated even for the incompressible case (Coats, 1968). Here we summarise the final results of the analysis for three-phase incompressible flow.

If we denote e_1, e_2 and e_3 the errors in S_w, p and S_g, and define γ_l, $l = 1, 2, 3$ accordingly, we obtain after considerable manipulations (Exercise 5.6).

$$\Delta t \leq \frac{4V_p}{\gamma_1|P'_{cow}|T_w\dfrac{T_o + T_g}{T_T} + \gamma_3|P'_{cog}|T_g\dfrac{T_o + T_w}{T_T} + \dfrac{1}{T_T}\sqrt{X}} \tag{5.56}$$

where

$$X = [\gamma_1|P'_{cow}|T_w(T_o + T_g) - \gamma_3|P'_{cog}|T_g(T_o + T_w)]^2 + 4|P'_{cow}||P'_{cog}|T_w^2 T_g^2 \gamma_1 \gamma_3$$

It is easily verified that for regions where only two phases are flowing, the above expression reduces to eqn. (5.47). The analysis of three-phase compressible systems would result in complicated expressions. However, we can see from the analogy with the two-phase case discussed above that the addition of compressibility will relax the stability limit slightly. For practical purposes the stability limit derived for the incompressible case should be used even when the system is compressible.

Stability of the SS method. Because the SS method treats P_c implicitly, it is unconditionally stable with respect to the primary variables. This can be easily proved by Fourier analysis (Exercise 5.7).

5.4.1.2 *Stability with Respect to Transmissibilities*

For the purposes of this analysis, we assume incompressible, two-phase flow with zero capillary pressure and gravity forces. As shown previously, both IMPES and SS methods are identical in this case.

In a typical flow term,

$$u_{i+1/2} = |-T_{i+1/2}(p_{i+1} - p_i)$$

we need to evaluate the nonlinear transmissibility coefficient T at some point in space and time. While all the choices will be discussed in detail in Section 5.5, let us consider the upstream in space, explicit in time weighting, i.e.,

$$T_{i+1/2} = T_i^n \qquad \text{if flow is from } i \text{ to } i+1$$
$$T_{i+1/2} = T_{i+1}^n \qquad \text{if flow is from } i+1 \text{ to } i$$

Without loss of generality, we can assume that the flow is in the direction of positive x-axis. Then the flow equations are

$$T_{l_{i-1}}^n (p_{i-1} - p_i)^{n+1} + T_{l_i}^n (p_{i+1} - p_i)^{n+1} = \frac{V_{p_i}}{\Delta t}(S_l^{n+1} - S_l^n) \qquad l = \text{n, w} \tag{5.57}$$

Although it is possible to analyse this equation, it is more convenient to consider the equivalent system of eqns. (2.84) and (2.91). For a 1-D case, the solution of (2.84) is

trivial ($u_T = Q_T/A$) and we only have to consider the saturation equation

$$-\boldsymbol{u}_T \cdot \nabla f_w = -\boldsymbol{u}_T \frac{df_w}{dS_w} \nabla S_w = \phi \frac{\partial S_w}{\partial t} + q_w - f_w q_T$$

Discretisation of this equation for a block not containing a source gives

$$-Q_T \left[\left(\frac{df_w}{dS_w} \right)_i S_{w_i} - \left(\frac{df_w}{dS_w} \right)_{i-1} S_{w_{i-1}} \right]^n = \frac{V_{p_i}}{\Delta t} (S_w^{n+1} - S_w^n)_i \tag{5.58}$$

where the concept of upstream weighting is expressed by taking the saturations explicitly upstream. This form of the left side is conservative and therefore preferable to other discretisations such as

$$-Q_T \left(\frac{df_w}{dS_w} \right)_{i-1/2} (S_{w_i} - S_{w_{i-1}})^n$$

It is easy to show in this simple case that the treatment of S_w in eqn. (5.58) is exactly equivalent to the treatment of T_w in eqn. (5.57) (Exercise 5.8).

The error now satisfies the equation (neglecting variations in df_w/dS_w):

$$-Q_T \frac{df_w}{dS_w} (e_i - e_{i-1})^n = \frac{V_{p_i}}{\Delta t} (e^{n+1} - e^n)_i$$

Following the Fourier stability analysis, we let

$$e_i^n = \xi^n \exp(\sqrt{-1}\alpha i) \qquad \alpha = m\Delta x$$

and after substitution find

$$\xi = 1 + C[\exp(-\sqrt{-1}\alpha) - 1] \tag{5.59}$$

where

$$C = Q_T \frac{df_w}{dS_w} \frac{\Delta t}{V_p}$$

To find $|\xi|$, we express ξ in complex form as $\xi = a + \sqrt{-1}b$ and find

$$|\xi|^2 = a^2 + b^2 = 1 - 4C(1 - C)\sin^2\frac{\alpha}{2}$$

The stability condition $|\xi| < 1$ gives for $\sin^2\alpha/2 \simeq 1$

$$0 < C(1 - C) < \tfrac{1}{2}$$

which leads to the condition C < 1, i.e.,

$$\Delta t < \frac{V_p}{\dfrac{df_w}{dS_w} Q_T} \tag{5.60}$$

This inequality has a simple physical interpretation. It can be written as

$$\Delta t \frac{df_w}{dS_w} \frac{u_T}{\phi} < \Delta x \tag{5.61}$$

where we have divided by $A\phi$.

From the Buckley–Leverett theory of two-phase displacement we find that the term on the left is the velocity of the advance of a constant saturation surface:

$$\frac{u_T}{\phi}\frac{df_w}{dS_w} = \left(\frac{\partial x}{\partial t}\right)_{S_w = \text{constant}} = u_s$$

(e.g., Craft and Hawkins, 1959, p. 369). Then the stability condition can be expressed as

$$u_s \Delta t < \Delta x \tag{5.62}$$

which means that the flood front can only advance a distance of one grid block per time step or the throughput through any block per time step must be less than its pore volume. This stability limit is well known in the literature on numerical fluid dynamics (Richtmyer and Morton, 1967, p. 304).

5.4.2 Existence and Uniqueness of Solution

5.4.2.1 *The SS Method*

Let us consider the two-phase SS equations

$$(\mathbf{T} - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n}) = -\boldsymbol{R}^{n} + \boldsymbol{Q} \tag{5.63}$$

In the rest of this section, we will assume that the problem has no-flow boundary conditions which are discretised by the 'reflection' technique. Consequently, if a point $i + 1$ is outside the boundary, we set $\mathbf{T}_{i+½} = \boldsymbol{0}$ and $\mathrm{T}_{\mathbf{i}} = -(\mathrm{T}_{\mathbf{i}\,-\,½})$ in eqn. (5.17). For simplicity, we also assume zero rock compressibility. Then we can write the matrix **D** as

$$\mathbf{D}_{\mathbf{i}} = \mathbf{D}_{\mathbf{si}} + \mathbf{D}_{\mathbf{Bi}} = -\frac{V_p}{\Delta t}\begin{bmatrix} b_{\mathrm{w}}^{n+1}S'_{\mathrm{w}} & -b_{\mathrm{w}}^{n+1}S'_{\mathrm{w}} \\ -b_{\mathrm{n}}^{n+1}S'_{\mathrm{w}} & b_{\mathrm{n}}^{n+1}S'_{\mathrm{w}} \end{bmatrix} + \frac{V_p}{\Delta t}\begin{bmatrix} S_{\mathrm{w}}^{n}b'_{\mathrm{w}} & 0 \\ 0 & S_{\mathrm{n}}^{n}b'_{\mathrm{n}} \end{bmatrix} \tag{5.64}$$

We can now prove the following results:

Theorem 1. *Let* $S'_w < 0,\ b'_w > 0,\ b'_n > 0$. *Then eqn.* (*5.63*) *has a unique solution.*

Proof: Matrix **T** is symmetric, diagonally dominant with negative diagonal elements and positive non-diagonal elements. This implies that **T** is negative semidefinite. Matrix $\mathbf{D_s}$ also has zero row sum, but it is not symmetric. Under the assumptions made at least one element of $\mathbf{D_{Bi}}$ is positive. Therefore matrix $\mathbf{A} = \mathbf{T} - \mathbf{D}$ is strongly diagonally dominant for at least one row. Furthermore, **A** has at least one full subdiagonal and one full superdiagonal and it is easy to show that **A** is irreducible. Then **A** is nonsingular (Varga, 1962; Appendix, A.3.2). Note that if $0 < S_{wc} < S_{\mathrm{w}} < S_{\mathrm{wmax}} < 1$, **A** will be strictly diagonally dominant and irreducibility is not required.

The assumption of compressibility is essential for uniqueness, as indicated by the next theorem.

Theorem 2. Let $S'_w < 0,\ b'_w = 0,\ b'_n = 0$. *Then solution of eqn.* (*5.63*) *exists if and only if*

$$B_{\mathrm{w}}\sum_i Q_{\mathrm{w}\,i} + B_{\mathrm{n}}\sum_i Q_{\mathrm{n}\,i} = 0 \tag{5.65}$$

and it is determined up to an additive constant.

Proof: In the incompressible case matrix $\mathbf{D_B}$ is zero and therefore all rows of matrix **A** have zero sums. Since all diagonal elements are negative and the off-diagonal elements positive, it follows (Appendix, A.3.6) that **A** is singular. Multiply every wetting phase

equation by B_w, every non-wetting by B_n and denote the resulting matrix by $\hat{\mathbf{A}} = \hat{\mathbf{T}} - \hat{\mathbf{D}}$. It is easy to see that $\hat{\mathbf{A}}$ is symmetric. The symmetry and zero row sums of $\hat{\mathbf{T}}$ imply that

$$\sum_i (\mathbf{T}\boldsymbol{P})_{li} = \sum_i (\hat{\mathbf{T}}\boldsymbol{P})_{li} = 0 \qquad l = \mathrm{w},\ \mathrm{n} \tag{5.66}$$

for any vector $\boldsymbol{P}$. Here $(\hat{\mathbf{T}}\boldsymbol{P})_{li}$ denotes the result of multiplying the ith row of $\hat{\mathrm{T}}$ for the phase l by the vector $\boldsymbol{P}$. Since

$$\hat{\mathbf{D}}_{s_i} = \frac{V_{p_i}}{\Delta t} S'_{w_i} \begin{vmatrix} -1 & 1 \\ 1 & -1 \end{vmatrix}$$

it holds that

$$\sum_l \sum_i (\hat{\mathbf{D}}_s \boldsymbol{P})_{li} = 0$$

for any $\boldsymbol{P}$, which also implies that

$$\sum_l \sum_i \left| R_{li}^k = 0 \qquad \text{for any } \boldsymbol{P}^k \right. \tag{5.67}$$

If we now write the eqns. (5.63) as

$$(\hat{\mathbf{T}} - \hat{\mathbf{D}})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\hat{\boldsymbol{R}}^n + \hat{\boldsymbol{Q}} \tag{5.68}$$

sum them up and use the above results, we find that eqn. (5.65) must hold if a solution exists.

On the other hand it is easy to show that every principal minor of order $N-1$, where N is the order of $\mathbf{A}$, is irreducibly diagonally dominant and therefore the rank of $\mathbf{A}$ is $N-1$. This proves that the condition (5.65) is also sufficient. Since obviously any constant vector $\boldsymbol{P} = \boldsymbol{C}$ satisfies eqn. (5.68), all solutions have the form $\boldsymbol{P} = \boldsymbol{P}_1 + \boldsymbol{C}$, where $\boldsymbol{P}_1$ is one particular solution.

Physically, eqn. (5.65) represents conservation of mass in a closed incompressible system. The undetermined constant represents an arbitrary reference pressure, since in

an incompressible system the pressure level is immaterial.

We can now derive some useful relationships which are direct extensions of the results given in Chapter 3 (Section 3.7.1).

First, we can define the residual at the $n + 1$ level as

$$\boldsymbol{R}^{n+1} = (\mathbf{T} - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n}) + \boldsymbol{R}^{n} = \boldsymbol{Q} \tag{5.69}$$

Then from eqn. (5.66) we obtain the following result:

Theorem 3. Solution of eqn. (*5.63*)*, if it exists, satisfies*

$$\sum_i R_{li}^{n+1} = -\sum_i [\mathbf{D}(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n})]_{li} = \sum_i Q_{li}$$

where the term inside the second summation sign represents the l-phase part of the i*th element of the vector.*

The next theorem gives the relationship between matrix **D** and material balance errors. First, define a vector ***DS*** by

$$DS_{li}^{n,n+1} = \frac{V_{p_i}}{\Delta t}[(S_l b_l)^{n+1} - (S_l b_l)^{n}]_i \qquad l = \mathrm{w},\ \mathrm{n} \tag{5.70}$$

Note that $\Delta t\boldsymbol{DS}$ is a 'mass change' vector, and eqn. (5.70) is a direct extension of eqn. (3.172). Then the material balance error over a time step is by definition

$$E_l = \sum_i DS_{li}^{n,n+1} + \sum_i Q_{li} \qquad l = \mathrm{w},\ \mathrm{n} \tag{5.71}$$

Now, from Theorem 3 the following result is obtained directly:

Theorem 4. Solution of eqn. (*5.63*)*, if it exists, satisfies*

$$E_l = \sum_i DS_{li}^{n,n+1} - \sum_i [\mathbf{D}(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n})]_{li} \qquad l = \mathrm{w},\ \mathrm{n} \tag{5.72}$$

This theorem thus proves rigorously our previous assertion that the time derivatives defined by eqn. (5.3) will conserve mass, because in that case $\mathbf{D}(\mathbf{P}^{n+1} - \mathbf{P}^{n}) = \boldsymbol{DS}^{n,n+1}$.

On the other hand, eqn. (5.72) may give $E_l = 0$ accidentally even for an operator that is not conservative. Therefore, E_l should not be used to measure the convergence when solving the equations. Instead, one might use a norm, such as

$$\|E_l\| = \sum_i |E_{li}|$$

The analysis for three-phase flow is quite analogous. However, the matrices **T** and **D** are generally not diagonally dominant due to additional off-diagonal terms representing the mass transfer between oil and gas phase. Sheffield (1969) shows that **A** is non-singular for sufficiently small Δt, but his proof seems to be in error. The correct proof outlined in Exercise 5.9 shows that the three-phase SS equations have a unique solution if Δt satisfies the following condition

$$\Delta t(b - a) < V_p \frac{S_o}{B_o} R'_s \tag{5.73}$$

where a and b are certain constants. Note that the existence of solution for the case without mass transfer does not require a condition on Δt.

Generalisations of Theorems 3 and 4 are also straightforward, with the definition of the third element of ***DS*** (in a three-phase case l = o, w, g) as

$$DS_{g_i}^{n,n+1} = \frac{V_{p_i}}{\Delta t}[(S_g b_g)^{n+1} - (S_g b_g)^n + (S_o b_o R_s)^{n+1} - (S_o b_o R_s)^n]_i \tag{5.74}$$

5.4.2.2 *The IMPES Method*

The properties of IMPES can be analysed in the same manner as for the SS method. Since IMPES can be obtained from SS equations by algebraic manipulations after the P_c terms are fixed, all theorems stated previously are also valid for IMPES. This can also be shown directly. The two-phase IMPES equations, before their reduction to a single equation, are:

$$R_{w_i} = \Delta[T_w(\Delta p_n - \gamma_w \Delta z - \Delta P_c)]_i - \frac{V_{p_i}}{\Delta t}(S_w^n b'_w \Delta_t p_n + b_w^k \Delta_t S_w)_i = Q_{w_i}$$

$$R_{n_i} = \Delta[T_n(\Delta p_n - \gamma_n \Delta z)]_i - \frac{V_{p_i}}{\Delta t}(S_n^n b'_n \Delta_t p_n - b_n^k \Delta_t S_w)_i = Q_{n_i}$$

where we have again assumed $\Delta_t p_w = \Delta_t p_n$ and ϕ is constant; b_l^k denotes an approximation to b_l^{n+1}.

The finite difference material balance equation (Theorem 3), is now easily obtained by simply adding all equations for a given phase:

$$\sum_i R_{l_i} = -\sum_i \frac{V_{p_i}}{\Delta t}(S_l^n b_l' \Delta_t p_n + b_l^k \Delta_t S_l) = \sum_i Q_{l_i} \qquad l = w, n$$

Obviously, when $b_l^k = b_l^{n+1}$, the terms in the second sum become DS_{li}, defined by eqn. (5.70) and Theorem 4 also follows. Theorems 1 and 2 can also be obtained in a similar fashion.

5.4.3 Convergence

Because all the approximations we have used are consistent with the differential equations, convergence follows (for sufficiently small Δt) from stability (Chapter 3, Section 3.2.3). However, this is true only when the differential problem is properly posed and the operator is linear. These conditions are generally *not* satisfied and therefore convergence is not automatically assured. We will give an example in the next section, when a very reasonable difference scheme will converge to a wrong solution.

5.5 TREATMENT OF NONLINEARITIES

The nonlinearities in eqn. (5.20) appear in matrices **T** and **D** and implicitly also in the vector ***R***. A typical element of matrix **T** will be denoted for further discussion as

$$T_l = T_l[f_1(p_l), f_2(S_w)] = GC f_1 f_2 \tag{5.75}$$

where GC is the constant part of the transmissibility, $f_1 = 1/\mu B$, and $f_2 = k_{rl}$. Functions f_1 and f_2 may be approximated on different time levels and in different ways between grid points, generally as

$$T_{l_{i+1/2}}^{n+1} \simeq T_l[f_{1_{i_1}}^{k_1}, f_{2_{i_2}}^{k_2}]$$

where

$$i \le i_1, i_2 \le i+1 \qquad n \le k_1, k_2 \le n+1$$

The problem of approximating the i+½ level in the space co-ordinate is referred to as the 'weighting' problem. The problem of approximation of the $n + 1$ level in time is the problem of solution or local linearisation of the set of nonlinear equations.

A similar situation exists in approximating the matrix **D**.

All nonlinearities in eqn. (5.20) may be divided into two groups:

(*a*) *Weak nonlinearities*. All variables that are functions of the pressure of one phase only can be considered weak nonlinearities. These include B_l^{n+1}, $(1/B_l)'$, R_s, γ_l^{n+1} and μ_l^{n+1}. An example of actual pressure-dependent functions is presented in Chapter 2 (Fig. 2.4). The effect of weak nonlinearities depends on the degree of pressure change and disappears in problems in which pressure remains constant. It is generally satisfactory even in the case of varying pressure to evaluate pressure-dependent functions one step behind, i.e., as a function of p_l^n instead of p_l^{n+1}. Also, the approximation of i+½ level is not critical; e.g., we can use

$$f_{1\,i+1/2} \simeq \tfrac{1}{2}(f_{1\,i} + f_{1\,i+1})$$

(b) *Strong nonlinearities*. The coefficients that depend on saturation or capillary pressure, i.e., k_{rl} and S'_w are called strong nonlinearities. The nonlinearity due to gas percolation has a special character and will be discussed separately. Examples of functions k_r and P_c are again found in Chapter 2. It follows from eqn. (5.10) that the nonlinearity due to S'_w disappears if P_c is a linear function of saturation, but this is not true of k_r. Therefore k_r introduces the principal nonlinearity in eqn. (5.20).

The discussion in this section is illustrated by numerical results obtained with two test problems, which are described below.

Test Problem No. 1

The first test problem is the incompressible waterflood problem with zero capillary pressure (Buckley–Leverett problem). The k_r functions are given by Fig. 2.9 and the other data are (cf. Coats, 1968; and Todd *et al*., 1972): L = 1000 ft, $B_w = B_n = 1$, $\mu_w = \mu_n$ = 1 cp, k = 300 md, ϕ= 0•2. Non-wetting phase is produced at $x = L$ at a rate of 426•5 ft^3/day and the wetting phase is injected at $x = 0$ at the same rate. The reservoir is horizontal with a cross-sectional area of 10 000 ft^2 and constant initial saturation S_{wi} = 0•16. Since finite P_c values must be used for the SS method, the zero P_c function was approximated by a linear P_c function defined by P_c = 0•1 at S_w = 0•16 and P_c = 0 at S_w = 0•8. This small value of the capillary pressure has no influence on the solution.

Test Problem No. 2

The second problem is the compressible oil-gas problem including gas percolation, with the nonlinear functions given in Chapter 2 (Fig. 2.10). The other data are (McCreary, 1971): L = 135 ft, k = 20md, ϕ = 0•04. The densities at standard conditions are ρ_{wSTC} = 60 lb/ft^3 and ρ_{nSTc} = 0•05 lb/ft^3. Wetting phase is produced at $x = L$ at the rate of 2810 ft^3/day at standard conditions. The reservoir is a vertical column with cross-sectional area of 5 414929 ft^2, and constant initial saturation S_w = 0•99. Initial pressure is given by gravity equilibrium with p_w = 1750 psia at the top of the column.

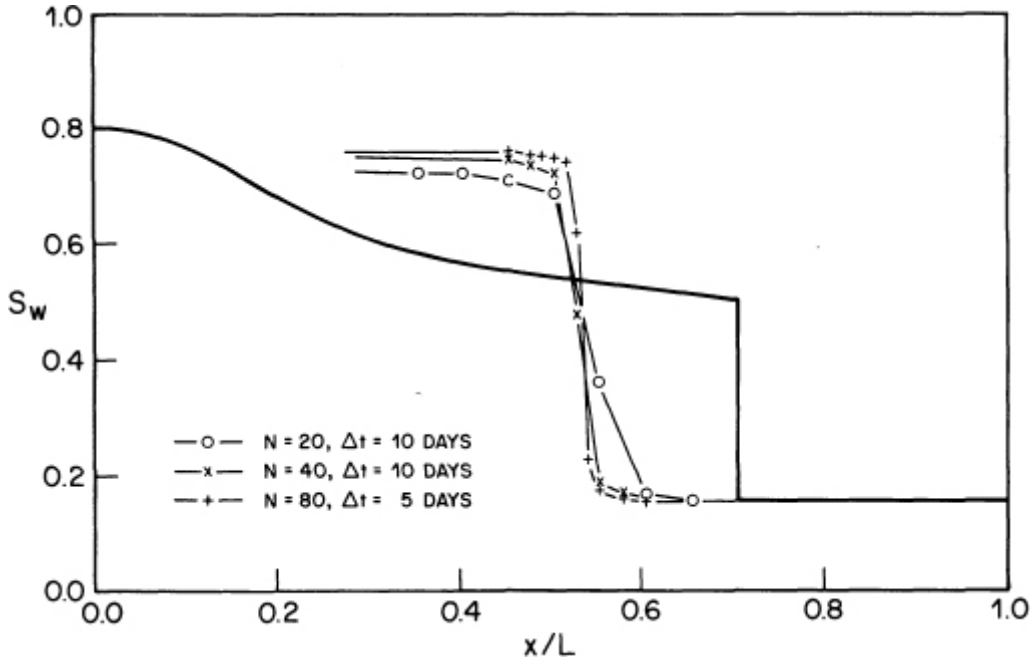


FIG. 5.1. Convergence of midpoint weighting—almost hyperbolic problem. Test problem No. 1, t = 1500 days.

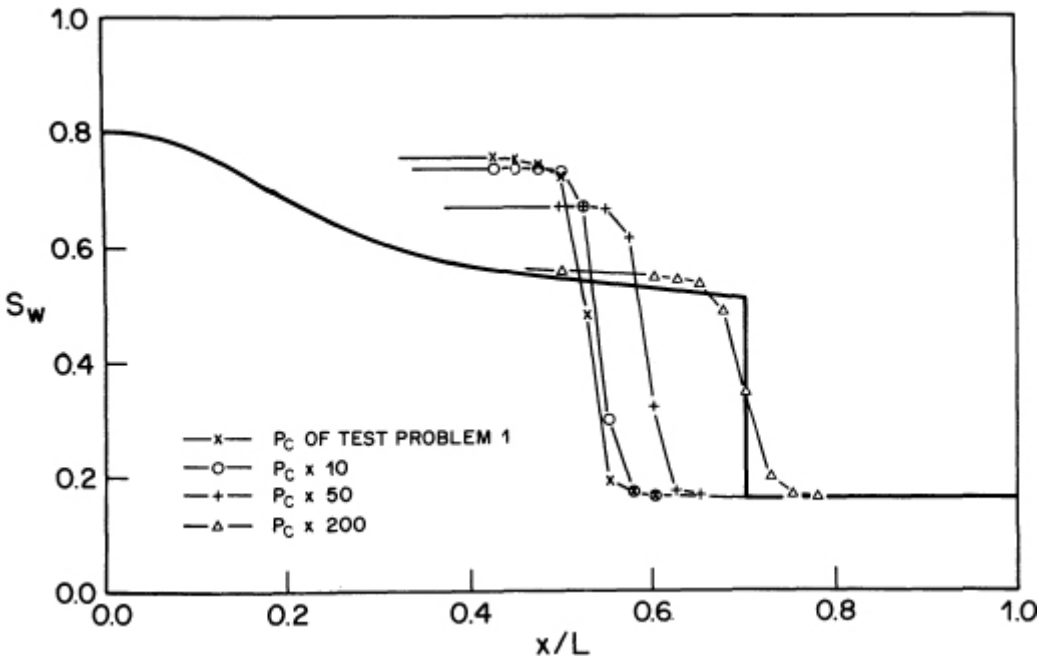


FIG. 5.2. Solution with midpoint weighting for different P_c. Test problem No. 1, $\Delta x = L/40$, $\Delta t = 10$ days, $t = 1500$ days.

5.5.1 Weighting of Transmissibilities

Weighting formulae relate the value of $k_{rli+1/2}$ *to* S_{wi} *and* S_{wi+1}. The approximation, that seems to be most appropriate from the standpoint of numerical analysis,

$$k_{\mathrm{rl}_{i+1/2}} = \tfrac{1}{2}[k_{\mathrm{rl}}(S_{\mathrm{w}_i}) + k_{\mathrm{rl}}(S_{\mathrm{w}_{i+1}})] \tag{5.76}$$

may be called 'midpoint weighting' and it is of second order. An alternative formula may be defined as

$$k_{\mathrm{rl}_{i+1/2}} = k_{\mathrm{rl}}[\tfrac{1}{2}(S_{\mathrm{w}_i} + S_{\mathrm{w}_{i+1}})] \tag{5.77}$$

Although both approximations are of second order, as shown later in this section, they produce erroneous results. This is shown in Fig. 5.1 for the numerical solution of the Buckley–Leverett problem. The small values of P_c used with the SS equations results in an almost hyperbolic problem with true solution very close to the Buckley–Leverett solution, shown in Fig. 5.1 by the solid line. However, with refinement of the grid, the numerical solution using the weighting scheme (eqn. (5.76)) actually converges to a different, unreal solution. This behaviour is a consequence of the hyperbolic nature of the equation. In the purely hyperbolic case ($P_c \equiv 0$), the differential problem is *not* properly posed (see Cardwell, 1959) and does not have a unique solution. The midpoint

weighting converges to a solution which is mathematically possible, but physically incorrect. When the magnitude of P_c' is increased, (the level of P_c is immaterial), the results approach the physically correct solution (Fig. 5.2). However, the P_c' at which this happens may be higher than the actual value for the problem being considered; moreover, it depends on the grid spacing.

For this reason, the commonly used scheme is the 'upstream weighting', defined by

$$k_{rl_{i+1/2}} = \begin{cases} k_{rl}(S_{w_i}) & \text{if flow is from } i \text{ to } i+1 \\ k_{rl}(S_{w_{i+1}}) & \text{if flow is from } i+1 \text{ to } i \end{cases} \tag{5.78}$$

The direction of flow is given by the sign of

$$\Delta\Phi_{l_{i+1/2}} = [p_{l_{i+1}} - p_{l_i} - \gamma_{l_{i+1/2}}(z_{i+1} - z_i)].$$

Flow is from i to $i + 1$ if $\Delta\Phi < 0$ and vice versa. This formula provides only a first-order approximation.

Todd *et al.* (1972) proposed an asymmetric second-order approximation that uses two upstream points:

$$k_{rl_{i+1/2}} = \begin{cases} \frac{1}{2}[3k_{rl}(S_{w_i}) - k_{rl}(S_{w_{i-1}})] & \text{for flow from } i \text{ to } i+1 \\ \frac{1}{2}[3k_{rl}(S_{w_{i+1}}) - k_{rl}(S_{w_{i+2}})] & \text{for flow from } i+1 \text{ to } i \end{cases} \tag{5.79}$$

When formula (5.79) is viewed in terms of weighting of saturations (as was done earlier for stability analysis, eqn. (5.58)), it turns out that it is equivalent to the second-order approximation proposed for convection–diffusion equations by Price *et al.* (1966). Because eqn. (5.79) involves an extrapolation process, it is essential to constrain computed values to physically admissible values, i.e., $0 < k_r < 1$.

Comparison of the two upstream formulae is shown in Fig. 5.3. Both methods converge to the correct answer. The second-order upstream formula gives a sharper displacement front than the single-point formula. A critical evaluation of upstream differencing for fluid flow problems is provided by Raithby (1976).

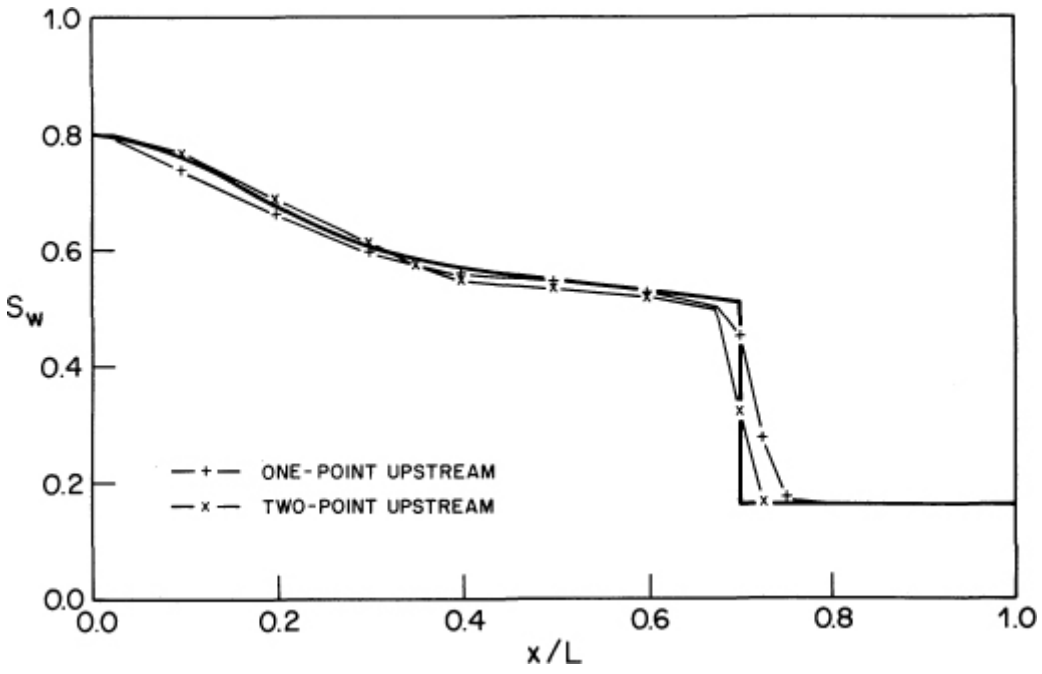


FIG. 5.3. Comparison of upstream weighting formulae. Test problem No. 1, $\Delta x = L/40$, $\Delta t = 10$ days, $t = 1500$ days (after Settari and Aziz, 1975).

5.5.1.1 *Truncation Errors and Discussion*

Truncation errors are best analysed with the hyperbolic form of the equation as we have done in Section 5.4.1. The operator to be approximated is denoted by

$$\mathrm{A}S_w = -\frac{\partial S_w}{\partial x} - C_w \frac{\partial S_w}{\partial t} = 0 \tag{5.80}$$

where

$$C_w = \frac{\phi}{u_T f'_w}$$

A difference operator approximating A at the point x_i is denoted by $\mathrm{L}(S_{wi})$ and the truncation error of L is defined as $e = \mathrm{A}S_w(x_i) - \mathrm{L}(S_{wi})$.

The derivation of the truncation errors is given in Settari and Aziz (1975). If we denote $\partial S_w/\partial x = S'$ and $\partial S_w/\partial t = S^{\cdot}$, the final results of the linearised analysis are:

$$R_m = -\frac{\Delta t}{2} S'^{\cdot} + \frac{\Delta t^2}{3} S'^{\cdot\cdot} + \frac{\Delta x^2}{6} S''' + O(\Delta^3) \tag{5.81}$$

for both midpoint formulae,

$$R_u = \left(-\frac{\Delta t}{2} + C_w \frac{\Delta x}{2} \right) S'^{\cdot} + \frac{\Delta t^2}{3} S'^{\cdot\cdot} + \frac{\Delta x \, \Delta t}{2} S''^{\cdot} + \frac{\Delta x^2}{6} S''' + O(\Delta^3) \tag{5.82}$$

for the single-point upstream weighting, and

$$R_{2u} = -\frac{\Delta t}{2} S'^{\cdot} + \frac{\Delta t^2}{3} S'^{\cdot\cdot} - \frac{\Delta x^2}{3} S''' + O(\Delta^3) \tag{5.83}$$

for the two-point upstream formula. All three expressions correspond to explicit approximation of k_{rl} in time.

The weighting problem is a good example of a problem in which the truncation error analysis alone can be completely misleading. The truncation errors of the midpoint and two-point upstream formulae differ only in the coefficient of the Δx^2 term, but their performance is quite different. The fact that upstream weighting is superior to higher-order midpoint weighting has also been observed in solving Navier–Stokes equations (Runchal and Wolfstein, 1969; Hirt, 1968).

For the solution of practical problems, the choice is between single-point and two-point upstream weighting. Note that the first-order term in eqn. (5.82) has two parts of opposite sign (since $C_w > 0$). Therefore, the first-order term will disappear when

$$\Delta t = C_{w_i} \Delta x$$

which is exactly the stability limit for upstream explicit transmissibilities (eqn. 5.60). This is a well-known phenomenon for hyperbolic equations; e.g., the centred difference equation gives an exact solution for a linear problem if $\Delta t = C_w \, \Delta x$ (Von Rosenberg, 1969). In general:

> *Truncation errors with upstream explicit transmissibility will be minimised by the use of the maximum stable time step.*

The two-point upstream weighting is more accurate, but it does not have the 'error cancellation' property.

It has an additional advantage of reducing the undesirable phenomenon called 'grid orientation effect' in multidimensional problems (Chapter 9). The computational effort is the same as for single-point weighting if k_{rl} are treated explicitly in time. In the case of semi-implicit treatment of k_{rl} (or implicit treatment of eqn. (5.80), use of this method increases the band-width of the matrix in multidimensional problems.

5.5.2 Approximation to Transmissibilities in Time

The approximation of the time level appears to be crucial for the stability of the finite-difference equations. The explicit approximation, i.e. $\mathrm{T}^{n+1} \cong \mathrm{T}(f_2^n)$ is only conditionally stable as shown in Section 5.4.1.2; and therefore it imposes a limitation on the size of time step. Stability problems become severe especially in the simulation of multidimensional flow around a single well, where high flow velocities are attained due to convergence of flow towards the sink. It was this application (coning simulation) in which the stability problem was first identified (Welge and Weber, 1964). It has been demonstrated that the stability problem is a result of the explicit treatment of transmissibilities (Blair and Weinaug, 1969), and several methods of handling this problem are now available, involving linearised (MacDonald and Coats, 1970; Letkeman and Ridings, 1970; Sonier *et al*., 1973) as well as nonlinear (Nolen and Berry, 1972; Robinson, 1971) approximations to the fully implicit transmissibilities. We will show in this section that most of these methods are closely related to Newton's method for the solution of nonlinear equations.

The treatment of production terms is an integral part of these methods. Production terms must be approximated in the same fashion as the interblock transmissibilities, in order to maintain stability when saturations change near production wells. The treatment of production terms is discussed in Section 5.7.

In the rest of this section, we will assume single-point, upstream weighting for the space approximation of k_{rl}, eqn. (5.78). When we start considering **T** and **D** at different time levels, it is also necessary to introduce new notation for the residual vector, defined previously by eqn. (5.18). We write

$$\boldsymbol{R}_{\mathrm{m}}^{k} = \mathbf{T}^{m}\boldsymbol{P}^{k} - \mathbf{D}^{m}(\boldsymbol{P}^{k} - \boldsymbol{P}^{n}) - \boldsymbol{G}^{m} \tag{5.84}$$

where the superscript m on the right side denotes the time level at which the coefficients are evaluated. In this notation, we can write eqn. (5.20) with coefficients evaluated at time level m as

$$(\mathbf{T}^{m} - \mathbf{D}^{m})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n}) = -\boldsymbol{R}_{\mathrm{m}}^{n} + \boldsymbol{Q} \tag{5.85}$$

Further differentiation of time levels will be necessary when we discuss treatment of matrix **D**, since **D** is required at time level $n + 1$.

Of the two basic solution methods (SS and IMPES), only the SS method is suitable for implicit treatment of **T** matrix. The IMPES method, by definition, assumes explicit treatment of P_c and saturations in matrix **T**. However, an implicit analog of IMPES known as the SEQ method will be discussed in Section 5.6.

For clarity, we will treat in this section only the SS method with pressures as the dependent variables. However, we will make remarks whenever the treatment differs if other variables are chosen.

5.5.2.1 *Some Basic Methods*

(*a*) *Explicit Transmissibilities*

As shown before, the approximation

$$\mathrm{T}^{n+1} \simeq \mathrm{T}(f_2^n) \tag{5.86}$$

is only conditionally stable. This is demonstrated in Fig. 5.4 where the solutions for Test problem No. 1 obtained with different time steps are shown in comparison with the exact solution. Note that in this case the frontal advance velocity is ~0•5ft/day and the stability limit from eqn. (5.62) is $\Delta t \sim 50$ days which is in agreement with the numerical results. Instead of using $\mathbf{P}^n$ in eqn. (5.86), one might extrapolate pressure and

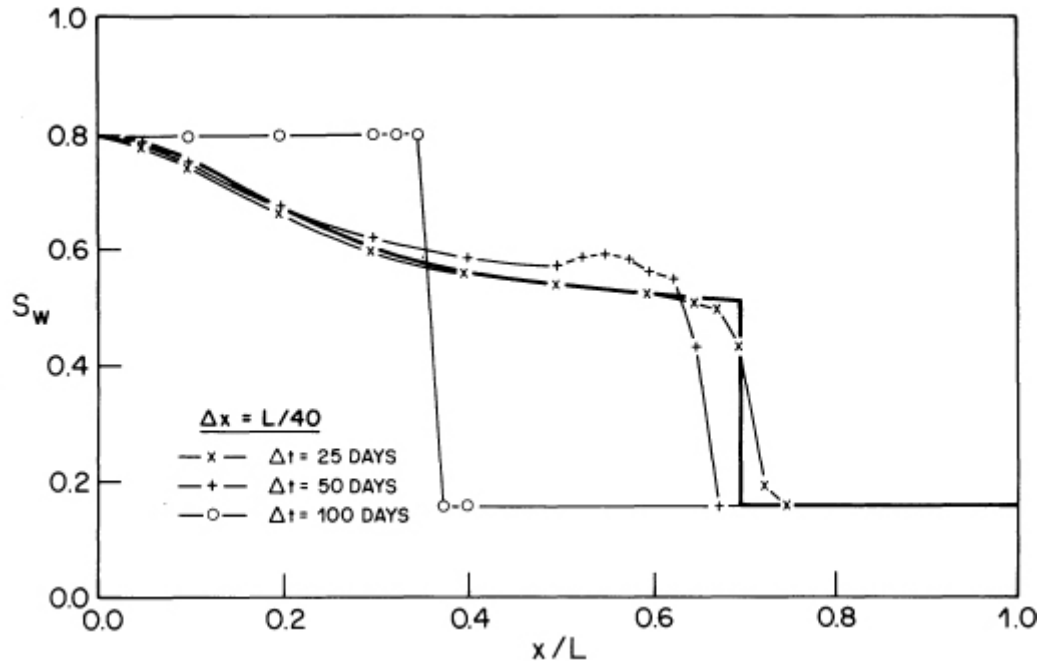


FIG. 5.4. Stability of the SS Method with explicit transmissibilities for Test problem No. 1 at t = 1500 days (from Settari and Aziz, 1975).

saturation from two previous time steps, i.e., compute

$$\boldsymbol{P}^k = \boldsymbol{P}^n + \frac{\Delta t^{n+1}}{\Delta t^n}(\boldsymbol{P}^n - \boldsymbol{P}^{n-1})$$

and use $\mathrm{T}(f_2^k)$. This provides only a slight improvement in stability. (Note that the results shown in Fig. 5.4 are also valid for the IMPES method since $P_c \cong 0$ for this case.)

(*b*) *Simple Iteration on Matrix* **T**

Such a method for solving eqn. (5.20) with $\mathbf{T}^{n+1}$ may be written as

$$[\mathbf{T}^{(\nu)} - \mathbf{D}][\boldsymbol{P}^{(\nu+1)} - \boldsymbol{P}^{(\nu)}] = -\boldsymbol{R}_{(\nu)}^{(\nu)} + \boldsymbol{Q} \qquad \nu = 0, 1, \ldots; \ \boldsymbol{P}^{(0)} = \boldsymbol{P}^{(n)} \tag{5.87}$$

where $\mathbf{T}^{(\nu)} = \mathbf{T}(f_2^{(\nu)})$. It has been found through numerical experiments that eqn. (5.87) converges for $\Delta t < \Delta t_{cr}$, where Δt_{cr} is the stability limit for the explicit approximation (eqn. 5.86). During the iterative process, the saturations oscillate with decreasing amplitude if $\Delta t < \Delta t_{cr}$, and with increasing amplitude if $\Delta t > \Delta t_{cr}$. In the latter case, the oscillations can be 'damped out' by the use of weighted average for $\mathbf{T}^{(\nu)}$, but methods of this type are impractical.

(*c*) *Linearised Implicit Transmissibilities*

The method in its original formulation (MacDonald and Coats, 1970; Letkeman and Ridings, 1970) consists of extrapolating T_l by the first-order approximation to f_2^{n+1} as follows:

$$\mathrm{T}_l^{n+1} \simeq \mathrm{T}_l(f_2^n) + \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{n+1} - P_c^n) \tag{5.88}$$

where

$$\frac{\partial \mathrm{T}_l}{\partial P_c} = GC f_1 \frac{\mathrm{d}f_2}{\mathrm{d}S_\mathrm{w}} \frac{\mathrm{d}S_\mathrm{w}}{\mathrm{d}P_c}$$

is the derivative with respect to the upstream point. These extrapolated transmissibilities are introduced into $\mathbf{T}\boldsymbol{P}$ and the nonlinear terms are linearised. For example, the nonlinear part of a typical term of $\mathbf{T}\boldsymbol{P}$, $\mathrm{T}_{l_{i+1/2}}^{n+1}(p_{l_{i+1}} - p_{l_i})^{n+1}$, is linearised by the following assumption

$$(p_{l_{i+1}} - p_{l_i})^{n+1} \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{n+1} - P_c^n) \simeq (p_{l_{i+1}} - p_{l_i})^n \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{n+1} - P_c^n) \tag{5.89}$$

We will now show that this method of linearisation can be interpreted as the first iteration of Newton's method for the equation with implicit transmissibilities. With m =

$n + 1$, eqn. (5.85) will be

$$\boldsymbol{R}_{\mathrm{n}+1}^{n+1} = \mathbf{T}^{n+1}\boldsymbol{P}^{n+1} - \mathbf{D}^{n+1}(\boldsymbol{P}^{n+1} - \boldsymbol{P}^{n}) - \boldsymbol{G}^{n+1} = \boldsymbol{Q} \tag{5.90}$$

and the classical Newton's method for it is an iterative process defined by:

$$\mathbf{DR}^{(\nu)}[\boldsymbol{P}^{(\nu+1)} - \boldsymbol{P}^{(\nu)}] = -\boldsymbol{R}_{(\nu)}^{(\nu)} + \boldsymbol{Q} \qquad \nu = 0, 1, \ldots; \ \boldsymbol{P}^{(0)} = \boldsymbol{P}^{(n)} \tag{5.91}$$

where **DR** is the Jacobi matrix of $R(P)$.

Let us now assume that **D** and γ_l are constant and examine the Jacobi matrix **DR**. By definition, elements of **DR** are partial derivatives of the vector ***R***. In the notation introduced earlier, the block element of **DR** in the ith row and jth column will consist of derivatives $\partial R_{l_i}/\partial P_{k_j}$, where $l, k = \mathrm{w}, \mathrm{n}$:

$$\begin{bmatrix} \dfrac{\partial R_{\mathrm{w}_i}}{\partial p_{\mathrm{w}_j}} & \dfrac{\partial R_{\mathrm{w}_i}}{\partial p_{\mathrm{n}_j}} \\[2ex] \dfrac{\partial R_{\mathrm{n}_i}}{\partial p_{\mathrm{w}_j}} & \dfrac{\partial R_{\mathrm{n}_i}}{\partial p_{\mathrm{n}_j}} \end{bmatrix}$$

It can be readily seen that the matrix **DR** can only have non-zero elements in the locations of the three block-diagonals of matrix **T** (note that this is not true if two-point upstream weighting is used). Under the above assumptions, derivatives of γ_l are zero and derivatives of **D*P*** give again matrix **D**.

The ith elements of the vector **TP** – ***G*** may be written as

$$-\mathrm{T}_{l_{i-1/2}}[p_{l_i} - p_{l_{i-1}} - (\gamma_l \Delta z)_{i-1/2}] + \mathrm{T}_{l_{i+1/2}}[p_{l_{i+1}} - p_{l_i} - (\gamma_l \Delta z)_{i+1/2}] \tag{5.92a}$$

or in a concise form as

$$-(\mathrm{T}\,\Delta\Phi)_{l_{i-1/2}} + (\mathrm{T}\,\Delta\Phi)_{l_{i+1/2}} \qquad l = \mathrm{w}, \mathrm{n} \tag{5.92b}$$

The three non-zero elements for a typical row of matrix **DR** may be derived by differentiating eqn. (5.92a) three times with respect to $p_{l_{i-1}}$, p_{l_i} and $p_{l_{i+1}}$. Using upstream weighting for the transmissibilities for flow from i to $i + 1$, the derivatives may be written as

$$\frac{\partial}{\partial p_{k_{i+1}}}[(\mathrm{T}\,\Delta\Phi)_{l_{i+1/2}} - (\mathrm{T}\,\Delta\Phi)_{l_{i-1/2}}] = \delta_{kl}\mathrm{T}_{l_{i+1/2}}$$

$$\frac{\partial}{\partial p_{k_i}}[(\mathrm{T}\,\Delta\Phi)_{l_{i+1/2}} - (\mathrm{T}\,\Delta\Phi)_{l_{i-1/2}}] = -\delta_{kl}\mathrm{T}_{l_{i+1/2}} + \mathrm{T}'_{l_{i+1/2}}\frac{\partial P_{c_i}}{\partial p_{k_i}} - \delta_{kl}\mathrm{T}_{l_{i-1/2}}$$

$$\frac{\partial}{\partial p_{k_{i-1}}}[(\mathrm{T}\,\Delta\Phi)_{l_{i+1/2}} - (\mathrm{T}\,\Delta\Phi)_{l_{i-1/2}}] = \delta_{kl}\mathrm{T}_{l_{i-1/2}} - \mathrm{T}'_{l_{i-1/2}}\frac{\partial P_{c_{i-1}}}{\partial p_{k_{i-1}}} \qquad k = \mathrm{n},\ \mathrm{w}$$

where we have used the definition

$$\mathrm{T}'_{l_{i+1/2}} = \Delta\Phi_{l_{i+1/2}}\frac{\partial \mathrm{T}_{l_{i+1/2}}}{\partial P_{c_i}} \tag{5.93}$$

We note that the derivatives of T_l are with respect to the upstream value of the capillary pressure. Furthermore, it is easy to see that

$$\frac{\partial P_c}{\partial p_k} = \begin{cases} 1 \text{ for } k = \mathrm{n} \\ -1 \text{ for } k = \mathrm{w} \end{cases}$$

since $P_c = p_n - p_w$. The elements of **DR** contain the terms of **T** – **D** matrix and up to four additional T'_l terms in a row.

After collecting all terms it is easy to see that the matrix **DR** may be written as:

$$\mathbf{DR} = \mathbf{T} + \mathbf{T}' - \mathbf{D} \tag{5.94}$$

where $\mathbf{T}'$ is a matrix composed of T'_l. The form of $\mathbf{T}'$ is generally dependent on the direction of flow. In a special case, when the flow is in the direction of increasing i for all grid points and for both phases, $\mathbf{T}'$ will be a lower block-triangular matrix with non-zero entries in only the main diagonal and the subdiagonal. If the diagonal block-element of matrix $\mathbf{T}'$ for the row i is denoted by $\mathrm{T}C_i$ and the subdiagonal element by $\mathrm{T}X_i$, the matrix will be

$$\mathbf{T}' = \begin{bmatrix} TC_1 & & & & & & \\ TX_2 & TC_2 & & & & & \\ & \ddots & \ddots & & & & \\ & & TX_i & TC_i & & & \\ & & & \ddots & \ddots & & \\ & & & & TX_N & TC_N \end{bmatrix}$$

where

$$\mathbf{TX_i} = \begin{bmatrix} T'_w & -T'_w \\ T'_n & -T'_n \end{bmatrix}_{i-1/2} \qquad \begin{aligned} \mathbf{TC_i} &= -\mathbf{TX_{i+1}} \qquad i = 1, \ldots, N-1 \\ \mathbf{TC_N} &= \mathbf{0} \end{aligned} \tag{5.95}$$

The derivatives $\partial T_l/\partial P_c$ and the $\Delta\Phi$ terms in eqn. (5.93k) may be evaluated at different time levels m and k and the matrix $\mathbf{T}'$ will then be denoted by $\mathbf{T}'^{k}_{m}$ in analogy with the definition of $\boldsymbol{R}$ (eqn. (5.84)).

For the classical Newton's method with tangents, both k and m are at the level of the previous iteration, i.e., $\mathbf{T}' = \mathbf{T}'^{(\nu)}_{(\nu)}$ and $\partial T_l/\partial P_c$ are tangents at $\boldsymbol{P}^{(\nu)}$. If it is now assumed that only one Newton's iteration (eqn. (5.91)) will be performed per time step, $\boldsymbol{P}^{(1)} = \boldsymbol{P}^{n+1}$; then one obtains, with respect to eqn. (5.94), the equation

$$(\mathbf{T}^n + \mathbf{T}'^n_n - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\boldsymbol{R}^n_n + \boldsymbol{Q} \tag{5.96}$$

which is the matrix formulation of the linearised method (eqn. (5.89)). Therefore, we have the result:

> *Linearised method* (*5.89*) *is the first iteration of classical Newton's method.*

Numerical results for this method are in Fig. 5.5. The method is about twice as stable as the explicit method. It should be noted again, that the one-dimensional problem discussed here is not the most severe from the point of view of stability. In 1-D, instability of explicit equations occurs, when the saturation front advances one grid point per time step. In multidimensional (especially single-well) problems, instability of explicit equations occurs for much smaller time steps and the improvement by using

the linearised method (eqn. (5.96)) is much larger than indicated by the results shown on Fig. 5.5.

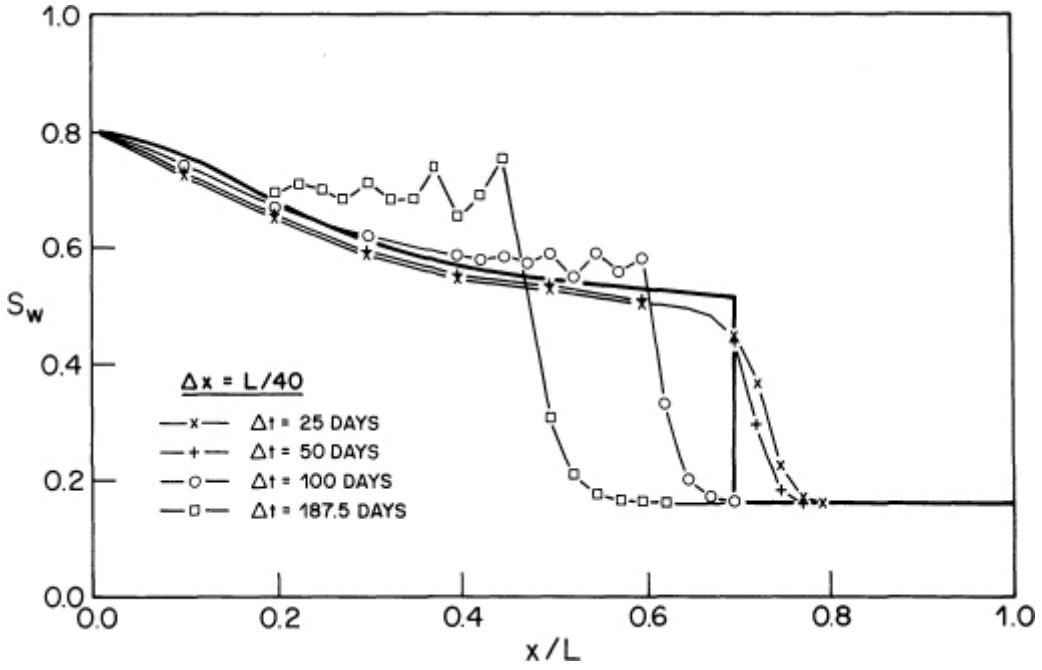


FIG. 5.5. Stability of the SS linearised implicit method for Test problem No. 1 at t = 1500 days (from Settari and Aziz, 1975).

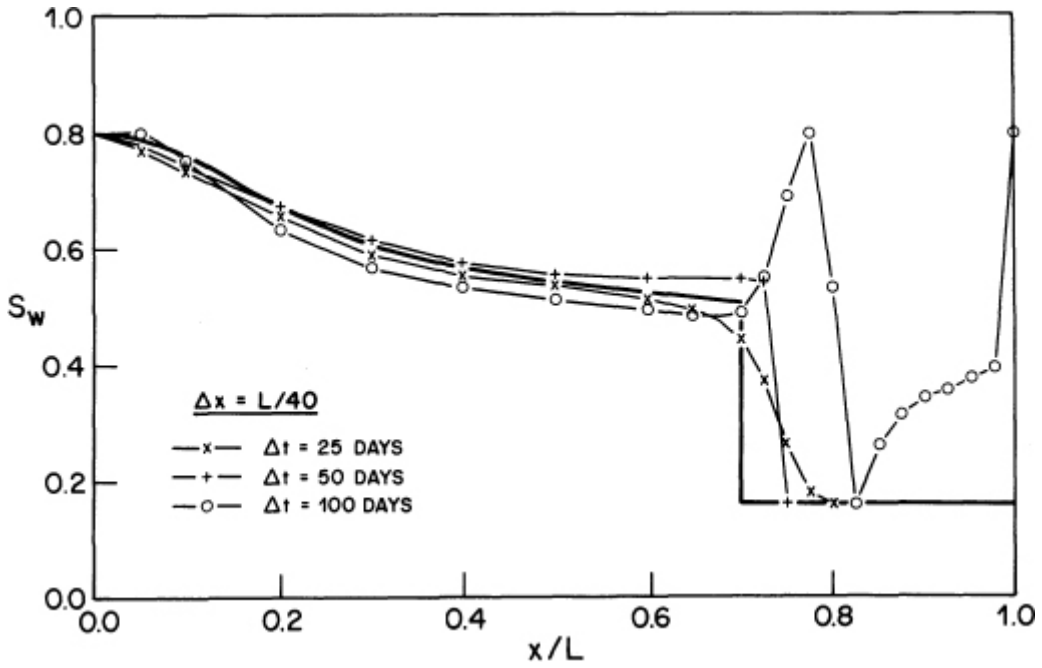


FIG. 5.6. Stability of the SS semi-implicit tangent method for Test problem No. 1 at t = 1500 days (from Settari and Aziz, 1975).

(*d*) *Semi-implicit method of Nolen and Berry* (1972)

These authors retain the nonlinearity in expressions (5.89). If we assume that the derivatives in $\mathbf{T}'$ are still evaluated at the level n, the matrix formulation of the method is

$$(\mathbf{T}^n + \mathbf{T}'^{n+1}_{\mathbf{n}} - \mathbf{D})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\boldsymbol{R}^n_{\mathbf{n}} + \boldsymbol{Q} \tag{5.97}$$

which represents a system of nonlinear equations. The nonlinearity $\mathbf{T}'^{n+1}_n(P^{n+1} - P^n)$ was solved by Newton's iteration by Nolen and Berry (1972). This is equivalent to iteration on the left side of eqn. (5.89) as follows:

$$\begin{aligned}(p_{l_{i+1}} - p_{l_i})^{(v+1)} \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{(v+1)} - p_c^n) &= (p_{l_{i+1}} - p_{l_i})^{(v)} \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{(v+1)} - P_c^{(v)}) \\ &+ [(p_{l_{i+1}} - p_{l_i})^{(v+1)} - (p_{l_{i+1}} - p_{l_i})^{(v)}] \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{(v)} - P_c^n) \\ &+ (p_{l_{i+1}} - p_{l_i})^{(v)} \frac{\partial \mathrm{T}_l}{\partial P_c}(P_c^{(v)} - P_c^n) \qquad v = 0, 1, 2, \ldots\end{aligned} \tag{5.98}$$

We observe immediately the following properties of the above method:

1. *If* $\boldsymbol{P}^{(0)} = \boldsymbol{P}^n$ *and only one iteration* (*5.98*) *is performed, the method of Nolen and Berry becomes the linearised implicit method* (*5.96*).

2. *If the functions* $k_{rl}(S_w)$ *are linear, the method of Nolen and Berry gives the solution of the fully implicit equations.*

Note that the second conclusion does not hold for the linearised method.

The treatment of the derivatives $\partial T_l/\partial P_c$ is crucial for the convergence of the iterations in eqn. (5.98). This is demonstrated by Figs. (5.6) and (5.7). The first figure shows the results with a tangent method, when $\partial \mathrm{T}_l/\partial P_C$ is a tangent at $\boldsymbol{P}^n$. In this case iterations start to diverge for Δt = 100 days. Better results are obtained, when the derivative is approximated by a secant (chord) between $\boldsymbol{P}^n$ and a reasonable estimate of $\boldsymbol{P}^{n+1}$ denoted by $\boldsymbol{P}^k$

$$\frac{\partial \mathrm{T}_l}{\partial P_c} \simeq \frac{\mathrm{T}_l(S_w^k) - \mathrm{T}_l(S_w^n)}{S_w^k - S_w^n} \frac{\mathrm{d}S_w}{\mathrm{d}P_c} \tag{5.99}$$

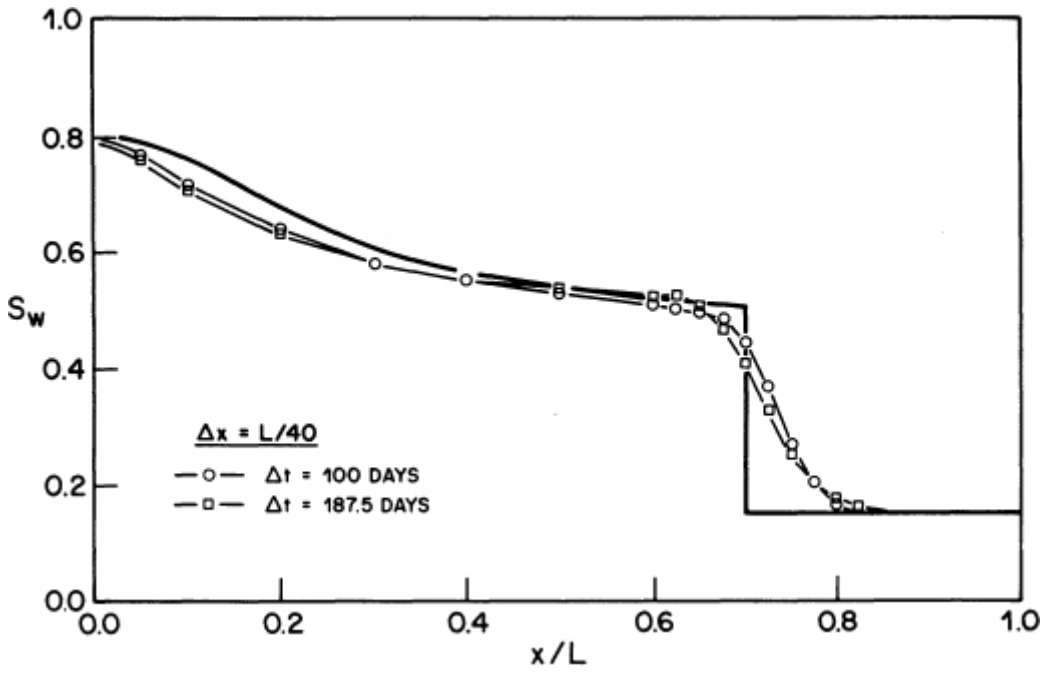


FIG. 5.7. Stability of the SS semi-implicit secant ($\delta S_W = 0{\bullet}5$) method for Test problem No. 1 at t = 1500 days (from Settari and Aziz, 1975).

Figure 5.7 shows the results obtained with a constant value $\delta S_w = S_w^k - S_w^n = 0 \cdot 5$, which are clearly superior to all other methods discussed so far.

(*e*) *Fully implicit method*

All methods discussed so far used only some approximation to the fully implicit equations

$$(\mathbf{T}^{n+1} - \mathbf{D}^{n+1})(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\boldsymbol{R}^n_{n+1} + \boldsymbol{Q} \tag{5.100}$$

These equations may also be solved by the Newton's method. Using the notation already introduced the tangent method may be written as

$$[\mathbf{T}^{(\nu)} + \mathbf{T}'^{(\nu)}_{(\nu)} - \mathbf{D}][\boldsymbol{P}^{(\nu+1)} - \boldsymbol{P}^{(\nu)}] = -\boldsymbol{R}^{(\nu)}_{(\nu)} + \boldsymbol{Q} \tag{5.101}$$

$$\nu = 0, 1, 2, \ldots; \quad \boldsymbol{P}^{(0)} = \boldsymbol{P}^n$$

Numerical results for the fully implicit transmissibilities are shown on Fig. 5.8.

The development is quite analogous when the equations are formulated in p and 5, and the form of matrix $\mathbf{T}'$ is easily deduced from $\mathbf{T}$ given in Section 5.2.3. The drivatives are taken directly with respect to S_w rather than P_c.

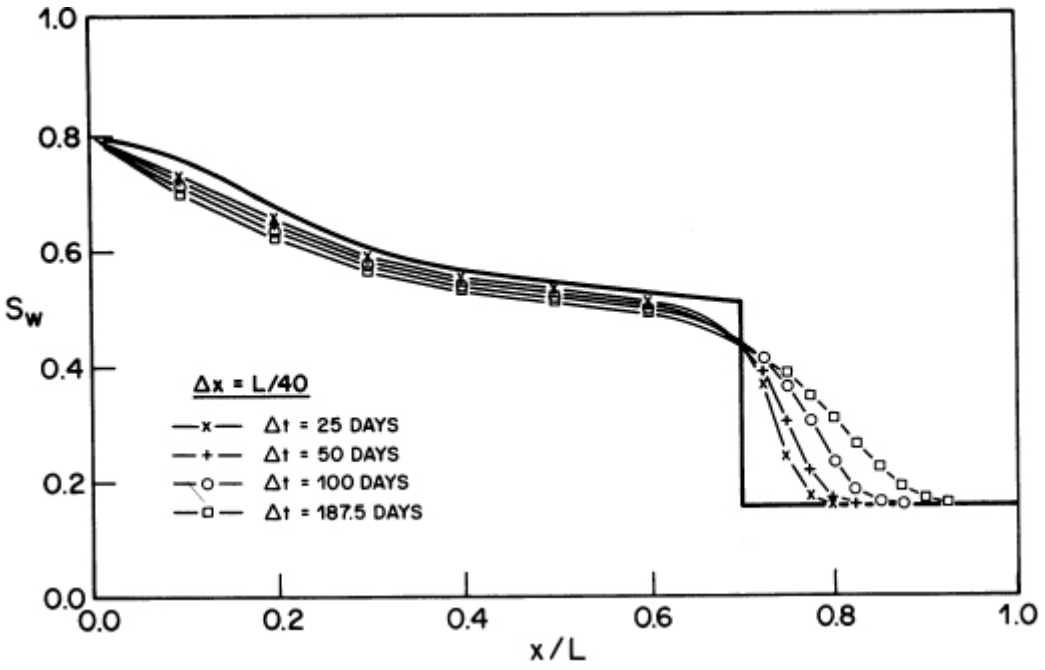


FIG. 5.8. Stability of the implicit method. Test problem No. 1 at t = 1500 days (from Settari and Aziz, 1975).

5.5.2.2 *Discussion of Basic Methods*

It is easy to see from the form of matrix **T′** that, although **T′** is not symmetric, eqn. (5.66) holds for it. From this it follows that for all methods formulated here, Theorems 3 and 4 of Section 5.4.2 hold and therefore these methods *satisfy material balance.*

Stability can be investigated in the same fashion as it was done for the case of explicit transmissibilities in Section 5.4.1.2. Such a linearised stability analysis shows that all three methods (i.e., linearised, semi-implicit and fully implicit) are unconditionally stable. A more refined, nonlinear stability analysis of the linearised method was given by Peaceman (1977). It shows that this method has a stability limitation, depending on f''_w, but this limit does not impose any significant restrictions in practice.

We also need to investigate the convergence of the iterative process for the semi-implicit and fully implicit method. Theoretical treatment of Newton's method becomes quite complicated for systems of equations (Ortega and Rheinboldt, 1970) and the conditions for convergence, existence, and uniqueness of solution are not easily established for practical problems. The essential conditions are that the functions R_i^{n+1} have continuous second derivatives and the Jacobi matrix **DR** have an inverse, and they are usually met for practical problems. Note that we always have a good starting value for the iteration, which is the result of the previous time step.

A fast rate of convergence is crucial for the practical feasibility of the fully implicit method as well as for the semi-implicit method, because one iteration needs approximately the same amount of work as does the solution of one time step for any linearised method. This comment is based on the assumption that each iteration by Newton's method is solved to the same degree of accuracy as the solution of linearised equations. While this is always the case when a direct method is used for the solution of the linearised matrix equations, the work ratio may be more favourable for Newton's method when an iterative method is used, since the equations for every Newton's iteration require to be solved only approximately in the latter case (Nolen and Berry, 1972).

5.5.2.3 *Comparison of Methods for the Implicit Treatment of Transmissibilities*

Examination of the results presented in this section shows that the stability increases with increasing 'implicitness' of the method, as we progress from explicit to fully implicit treatment. However, this improvement is gained at the expense of larger truncation errors. The linearised truncation error analysis gives the following errors (Settari and Aziz, 1975).

For the linearised and semi-implicit method

$$R = \left[-\frac{3}{2}\Delta t + C_w \frac{\Delta x}{2} \right] S'' + \frac{4\,\Delta t^2}{3} S''' + \frac{2\,\Delta x\,\Delta t}{3} S''' + \frac{\Delta x^2}{6} S'''' + O(\Delta^3) \tag{5.102}$$

For the fully implicit method

$$R = \left[\frac{\Delta t}{2} + C_w \frac{\Delta x}{2} \right] S'' + \frac{\Delta t^2}{6} S''' + \frac{\Delta x^2}{6} S'''' + O(\Delta^3) \tag{5.103}$$

Note that in comparison with the error for the explicit method (eqn. (5.82)), the implicit method has always larger errors. This was pointed out first by MacDonald and Coats (1970). The linearised method exhibits partial error cancellation, as seen from Fig. 5.5, while the errors with the implicit method increase monotonically with Δt.

It is possible to formulate other similar methods; for example, the linearised method based on the second-order Newton's scheme was investigated by Settari and Aziz (1975). All of the available results lead to the following observation:

> *As the implicitness of the method increases, stability improves, but truncation errors also increase.*

Finally, we will briefly discuss secant (chord) methods, already introduced by eqn. (5.99). If the chord is chosen reasonably, a secant method has a better rate of convergence than the corresponding tangent method (Ortega and Rheinboldt, 1970, Chapter 10). Ideally, the saturation change $\triangle S_w = S_w^k - S_w^n$ should be predicted for every point. Since this is difficult, a constant chord is usually used, which is calculated from the maximum anticipated saturation change.

Chord methods were compared with tangent methods by Settari and Aziz (1975). The improvement for the linearised method was marginal and the rate of convergence for the Newton's method was also relatively insensitive to the choice of the chord. Only the semi-implicit method of Nolen and Berry benefited from the use of chords.

For the solution of practical multidimensional problems, the linearised method is recommended. The fully implicit equations have large truncation errors and the stability gain may not be utilised because time steps are limited also by other considerations. The semi-implicit method has smaller truncation errors, but it is sensitive to the choice of chords, which makes it less suitable when saturation changes cannot be predicted (flow reversals, etc.).

5.5.3 Nonlinearity Due to P_c Function

When the function $S_w = f(P_c)$ is not linear, the elements of matrix **D** become implicit: $\mathbf{D} = \mathbf{D}^{n+1}$. We will consider for clarity again the incompressible flow with p_w, p_n as the dependent variables; in this case the ith block of matrix **D** for the SS method with p_w, p_n as dependent variables is

$$\mathbf{D}_i^{n+1} = \frac{V_{p_i}}{\Delta t}(S_w')_i^{n+1}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5.104}$$

The nonlinearity due to S_w' may also enter matrix **T',** its effect on this term is, however, small and will not be considered. In order for S_w' to satisfy eqn. (5.10), some iterative method must be used to obtain the solution. In the case that an iterative method is used to solve for implicit transmissibilities, the iterations on **D** may be subiterations of such an iterative method. In order to keep the discussion brief, we will consider the linearised method only.

(*a*) *Simple iteration*

Based on the last iteration $\boldsymbol{P}^{(v)}$, the derivative of S_w is updated as

$$S_w'^{(v)} = \frac{S_w(P_c^{(v)}) - S_w(P_c^n)}{P_c^{(v)} - P_c^n} \tag{5.105}$$

If the corresponding matrix $\mathbf{D}$ is denoted as $\mathbf{D}^{(\nu)}$, the iterative scheme for the linearised method (5.96) may be written as

$$(\mathbf{T}^n + \mathbf{T}'^n_{\mathbf{n}} - \mathbf{D}^{(\nu)})(\boldsymbol{P}^{(\nu+1)} - \boldsymbol{P}^n) = -\boldsymbol{R}^n_{\mathbf{n}} + \boldsymbol{Q} \qquad (5.106)$$

$$\nu = 0, 1, 2, \ldots; \qquad \boldsymbol{P}^{(0)} = \boldsymbol{P}^n$$

This method converges for small time steps, but its stability may be even lower than the stability of the equations with respect to explicit transmissibilities, depending on the function P_c. Also, the derivative S'_w must be continuous in order that the method converges at all. Numerically it means that the method requires at least second-order interpolation if the function P_c is given in the form of a table.

(*b*) *Newton's iteration*

In analogy with eqn. (5.70) define vector $\boldsymbol{DS}^{n,(\nu)}$ by

$$\boldsymbol{DS}^{n,(\nu)} = \frac{V_{p_i}}{\Delta t}[S_l^{(\nu)} - S_l^n] \qquad l = \mathrm{w}, \mathrm{n} \qquad (5.107)$$

Then the Newton's method, considering $\mathbf{D}^{n+1}$ to be the only nonlinearity, is

$$(\mathbf{T}^n + \mathbf{T}'^n_{\mathbf{n}} - \mathbf{D}^{(\nu)})(\boldsymbol{P}^{(\nu+1)} - \boldsymbol{P}^{(\nu)}) = -\boldsymbol{R}^n_{\mathbf{n}} - (\mathbf{T}^n + \mathbf{T}'^n_{\mathbf{n}})(\boldsymbol{P}^{(\nu)} - \boldsymbol{P}^n) \qquad (5.108)$$

$$+ \boldsymbol{DS}^{n,(\nu)} + \boldsymbol{Q} \qquad \nu = 0, 1, \ldots; \qquad \boldsymbol{P}^{(0)} = \boldsymbol{P}^n$$

where now S'_{w_i} is the tangent at $S^{(\nu)}_{w_i}$ rather than chord slope.

The method (5.108) does not always converge (Peaceman, 1967; Settari and Aziz, 1975). When the function S_{w} has very small derivatives at the ends of the interval, Newton's method will not converge (Ostrowski, 1973).

(*c*) *Modified Newton's method*

In order to ensure convergence of Newton's method it must be modified. One solution to this problem, proposed first by Peaceman (1967), is to do an 'inverse iteration', by treating S_{w} as the primary variable. This approach is discussed in detail by Settari and Aziz (1975) and only a brief discussion is presented here. After we solve the νth iteration of eqn. (5.108), we denote the capillary pressure as $P_c^{(c)} = p_n^{(\nu+1)} - p_w^{(\nu+1)}$ and calculate first new saturation $S_{\mathrm{w}}^{(\nu+1)}$ as

$$S^{(\nu+1)}_{\mathrm{w}_i} = S^{(\nu)}_{\mathrm{w}_i} + S'^{(\nu)}_{\mathrm{w}_i}(P_c^{(c)} - P_c^{(\nu)})_i \qquad i = 1, \ldots, N \qquad (5.109)$$

Then we re-define the capillary pressure as

$$P_{c_i}^{(v+1)} = f(S_{w_i}^{v+1}) \tag{5.110}$$

and adjust one of the phase pressures to satisfy $p_n - p_w = P_c$.

This method converges very rapidly and for most problems, there is no need to iterate at all.

In conclusion, we will discuss the treatment of P_c nonlinearity with different dependent variables. When p and P_c are used as variables, the treatment is quite analogous and the method defined by eqn. (5.109) and (5.110) is appropriate.

When p and S are used, then for this simple case the matrix **D** takes the form

$$\mathbf{D}_i^{n+1} = \frac{V_p}{\Delta t}\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$$

and it does not have a P_c nonlinearity. This also explains the rapid convergence of the modified iteration above, since it corresponds to use of S_w as one of the dependent variables.

5.5.4 Gas Percolation

In flow problems with solution gas, free gas is released from solution if the pressure decreases below the bubble point. Because gas viscosity is very small, its mobility is high and gas flows upwards (percolates) with relatively high velocities. Serious stability problems arise in the case of a vertical or inclined co-ordinate, as soon as the gas phase becomes mobile (i.e., $k_{rn} > 0$). Similar problems exist in other cases where gas flows due to the influence of gravity. This instability is essentially due to the explicit treatment of transmissibilities, and it is accentuated by the large difference in densities of oil and gas and the stronger nonlinearity of pressure-dependent functions for gas.

The first method for controlling this nonlinearity (Coats, 1968b) was developed before the importance of implicit treatment of coefficients was recognised. Later a similar, but simpler method was proposed (McCreary, 1971). Several investigators have shown that the use of the linearised or semi-implicit method also solves the stability problem associated with gas percolation.

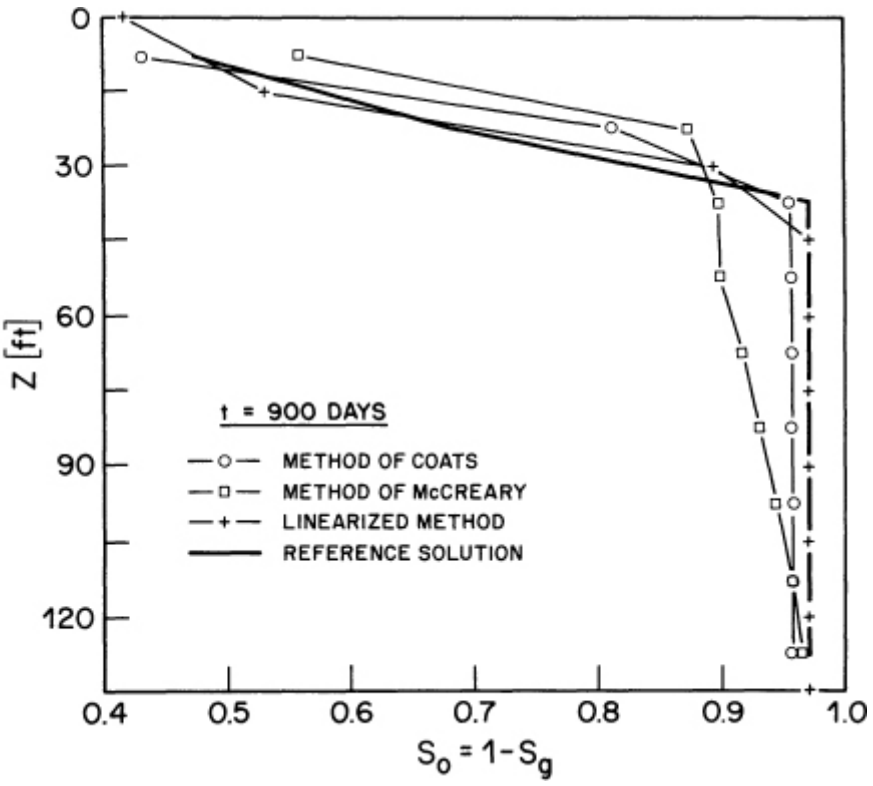


FIG. 5.9. Comparison of three methods for the gas percolation problem of McCreary, Test problem No. 2 (from Settari and Aziz, 1975).

To illustrate this, we will show a comparison of the linearised method (5.96) results with the results for methods of Coats and McCreary, reported in McCreary (1971), for the Test problem No. 2. Saturations calculated by all three methods for a selected time t = 900 days are shown in Fig. 5.9. The solid line represents the reference solution computed by McCreary using very small time steps and explicit transmissibilities. Obviously, the linearised method gives a much more accurate solution than the other two methods. This is especially true in the oil zone, where the saturation of gas is only slightly higher than the critical saturation at which gas starts to flow. The common feature of the methods of Coats and McCreary is that they restrict the mobility of gas regardless of the relationship $k_{rn} = f(S_w)$, which permits development of large gas saturations in the oil zone. This is especially true of the method of McCreary.

Stability of the linearised method for this problem was about equal to the method of Coats, while McCreary's method permitted use of about two times larger time steps. However, in view of the poor accuracy of McCreary's method, its gain in stability is not significant.

Therefore, the use of the linearised method is recommended in preference to earlier methods. However, the results of Nolen and Berry (1972) must also be kept in mind where they showed that the semi-implicit method may be superior to other methods for some difficult problems.

5.6. THE SEQUENTIAL SOLUTION METHOD (SEQ)

We have seen in Section 5.4 that both the IMPES and the explicit SS methods have a rather limited stability due to the explicit treatment of transmissibilities. This can be alleviated for the SS method by the implicit treatment of coefficients. The idea of the SEQ method is to improve the stability of the IMPES method by incorporating implicit treatment of saturations, but without solving simultaneously for pressures and saturations. Such a scheme was first formulated by MacDonald and Coats (1970), but its use was not reported until much later by Spillette *et al.* (1973), Coats *et al.* (1974) and Coats (1976). The SEQ method consists of two steps. The first step is to obtain an implicit pressure solution in exactly the same way as for the IMPES method. The second step is an implicit solution for saturations using linearised implicit transmissibilities. Therefore, the method can be looked at (and derived) in two ways; as a 'splitting-up' of the SS method with linearised implicit transmissibilities, or as an implicit approximation to the saturation (fractional flow) equation (eqn. 2.95).

5.6.1 SEQ Method for Two-phase Flow

Let us introduce the SEQ method by considering a simple case with $c_R = 0$ and $\Delta z = 0$. The SS equations (eqns. 5.11) with linearised implicit transmissibilities may be written in the following form:

$$\begin{aligned} &\mathrm{T}^n_{l_{i-1/2}}(p_{i-1} - p_i)^{n+1}_l - \mathrm{T}'_{l_{i-1/2}}(S^{n+1}_{\mathrm{w}} - S^n_{\mathrm{w}})_- \\ &\quad + \mathrm{T}^n_{l_{i+1/2}}(p_{i+1} - p_i)^{n+1}_l + \mathrm{T}'_{l_{i+1/2}}(S^{n+1}_{\mathrm{w}} - S^n_{\mathrm{w}})_+ \\ &= \frac{V_p}{\Delta t}[b^{n+1}_l \Delta_t S_l + S^n_l b'_l \Delta_t p_l] + Q^n_l + Q'_{l\mathrm{w}} \Delta_t S_{\mathrm{w}} + Q'_{lp} \Delta_t p_l \end{aligned} \tag{5.111}$$

where for flow from i to $i + 1$

$$\mathrm{T}'_{l_{i+1/2}} = (p_{i+1} - p_i)^n_l \frac{\partial \mathrm{T}_{l_{i+1/2}}}{\partial S_{\mathrm{w}_i}}$$

$Q'_{lw} \triangle_t S_w$ and $Q'_{lp} \triangle_t p_l$ are the implicit contributions to production with respect to S_{w} and p_l respectively. Subscripts + and – on $(S^{n+1}_{\mathrm{w}} - S^n_{\mathrm{w}})$ terms denote upstream values. In the first step of the SEQ method the T'_l and $Q'_{l\mathrm{w}}$ terms are neglected and the equations for the two phases are combined as for the IMPES method with $\Delta_t p_{\mathrm{w}} = \Delta_t p_{\mathrm{n}} = \Delta_t p$) to obtain a single pressure equation in $p = p_n$:

$$
\begin{aligned}
&(B^{n-1}_{\mathrm{w}_i}\mathrm{T}^n_{\mathrm{w}\,i-1/2} + B^{n+1}_{\mathrm{n}_i}\mathrm{T}^n_{\mathrm{n}\,i-1/2})(p_{i-1} - p_i)^{n+1} \\
&\quad + (B^{n+1}_{\mathrm{w}_i}\mathrm{T}^n_{\mathrm{w}\,i+1/2} + B^{n+1}_{\mathrm{n}_i}\mathrm{T}^n_{\mathrm{n}\,i+1/2})(p_{i+1} - p_i)^{n+1} \\
&= \left\{\frac{V_p}{\Delta t}[S^n_{\mathrm{w}}B^{n+1}_{\mathrm{w}}b'_{\mathrm{w}} + (1 - S^n_{\mathrm{w}})B^{n+1}_{\mathrm{n}}b'_{\mathrm{n}}]\right. \\
&\quad \left. + \mathrm{Q}'_{\mathrm{w}p}B^{n+1}_{\mathrm{w}} + \mathrm{Q}'_{\mathrm{n}p}B^{n+1}_{\mathrm{n}}\right\}(p^{n+1} - p^n)_i + B^{n+1}_{\mathrm{w}_i}[\mathrm{T}^n_{\mathrm{w}\,i-1/2}(P_{c\,i-1} - P_{c_i})^n \\
&\quad + \mathrm{T}_{\mathrm{w}\,i+1/2}(P_{c\,i+1} - P_{c_i})^n] + Q_{\mathrm{w}}B^{n+1}_{\mathrm{w}} + Q_{\mathrm{n}}B^{n+1}_{\mathrm{n}}
\end{aligned}
\tag{5.112}
$$

This is a parabolic equation and the expression inside {} on the right side represents total compressibility c_{T} for the block (note that Q'_{lp} terms contribute to c_{T} and may possibly make it negative).

After solving eqn. (5.112), we may solve the explicit (IMPES) saturation equation (e.g., eqn. (5.40)) and call the result S^*_{w}. However, it is not necessary to explicitly perform this step (see eqn. (5.115) below).

Now we can derive an implicit saturation equation by starting with eqn. (5.111) and utilising the pressure solution just obtained:

$$
\begin{aligned}
&\mathrm{T}^n_{\mathrm{w}\,i-1/2}(p_{i-1} - p_i)^{n+1} + \mathrm{T}^n_{\mathrm{w}\,i+1/2}(p_{i+1} - p_i)^{n+1} \\
&\quad = \frac{V_p}{\Delta t}[b^*_{\mathrm{w}}(S^*_{\mathrm{w}} - S^n_{\mathrm{w}}) + S^n_{\mathrm{w}}b'\,\Delta_t p] + \mathrm{T}^n_{\mathrm{w}\,i-1/2}(P_{c\,i-1} - P_{c_i})^n \\
&\quad + \mathrm{T}^n_{\mathrm{w}\,i+1/2}(P_{c\,i+1} - P_{c_i})^n + Q^n_{\mathrm{w}} + Q'_{\mathrm{w}p}(p^{n+1} - p^n)_i
\end{aligned}
\tag{5.113}
$$

where b^* is the value for b^{n+1} for the pressure step.

Substitution of eqn. (5.113) into (5.111) written for l = w provides the equation for the second step of the SEQ method:

$$
\begin{aligned}
&-\mathrm{T}'_{\mathrm{w}\,i-1/2}(\Delta_t\mathrm{S}_{\mathrm{w}})_- + \mathrm{T}'_{\mathrm{w}\,i+1/2}(\Delta_t\mathrm{S}_{\mathrm{w}})_+ - \mathrm{T}^n_{\mathrm{w}\,i-1/2}\Delta_t P_{c\,i-1} \\
&\quad + (\mathrm{T}^n_{\mathrm{w}\,i-1/2} + \mathrm{T}^n_{\mathrm{w}\,i+1/2})\Delta_t P_{c_i} - \mathrm{T}^n_{\mathrm{w}\,i+1/2}\Delta_t P_{c\,i+1} \\
&\quad = \frac{V_p}{\Delta t}[b^*_{\mathrm{w}}(S^{n+1}_{\mathrm{w}} - S^*_{\mathrm{w}})]_i + Q'_{\mathrm{ww}}\Delta_t\mathrm{S}_{\mathrm{w}_i}
\end{aligned}
\tag{5.114}
$$

This equation contains the implicit terms with respect to saturation and capillary pressure change over a time step. The $\Delta_t P_c$ terms appear here because the IMPES step

assumed $P_c^{n+1} = P_c^n$. It is important for stability to include P_c terms in this step (Spillette *et al.*, 1973). Equation (5.114) may be rearranged in terms of $\Delta_t S_w$ only:

$$-T'_{w_{i-1/2}}(\Delta_t S_w)_- + T'_{w_{i+1/2}}(\Delta_t S_w)_+ + [T^n_{w_{i-1/2}} P'_{c_{i-1}} \Delta_t S_{w_{i-1}} \tag{5.115}$$

$$-(T_{w_{i-1/2}} + T_{w_{i+1/2}})^n P'_{c_i} \Delta_t S_{w_i} + T^n_{w_{i+1/2}} P'_{c_{i+1}} \Delta_t S_{w_{i+1}}]$$

$$= \left[\frac{V_p}{\Delta t} b^*_w + Q'_{ww}\right]_i \Delta_t S_{w_i} - \left[\frac{V_p}{\Delta t} b^*_w (S^*_w - S^n_w)\right]_i$$

where the value of the last term can be obtained from eqn. (5.113) without computing S^*_w. Equation (5.115) can be written in matrix form as:

$$(\mathbf{T}'_s + \mathbf{T}'_{P_c} - \mathbf{D}_s)\,\Delta_t \boldsymbol{S}_w = \boldsymbol{R}_s \tag{5.116}$$

where $\mathbf{T}'_s$ and $\mathbf{T}'_{P_c}$ are the tridiagonal matrices of implicit flow terms due to changes in relative permeability and capillary pressure over the time step and $\mathbf{D}_s$ is the diagonal accumulation matrix. The part of the equation due to k_r changes has a hyperbolic character, and the part due to P_c changes has a parabolic character. This will become obvious later when we consider the derivation from the fractional flow equation.

Solution of eqn. (5.112) gives the new pressures and solution of eqn. (116) the new saturations in a sequential fashion. Both steps require solution of a matrix equation of the same size. Therefore, the work for one time step with the SEQ method is approximately twice the work for the IMPES method, regardless of the dimensionality of the problem. This makes the SEQ method much faster compared to the SS method in 2-D and 3-D problems (see Chapter 9).

Let us now analyse the implicit saturation calculation (5.116) in detail. First, we recall that the IMPES step itself satisfies material balance (Section 5.4.2.2). If we now sum up eqn. (5.115), all transmissibility terms on the left-hand side will cancel out. The resulting equation is:

$$0 = \sum_i \left[\frac{V_p}{\Delta t} b^*_w (S^{n+1}_w - S^*_w)\right]_i + \sum_i (Q'_{ww} \Delta_t S_w)_i$$

Since $b^{n+1}_w = b^*_w$, this can be written as

$$0 = \sum_i \frac{V_{p_i}}{\Delta t} [(b_w S_w)^{n+1} - (b_w S_w)^*]_i + \sum_i \Delta_t Q_{w_i} \tag{5.117}$$

where we have denoted by $\Delta_t Q$ the implicit contribution to the production rate Q. Equation (5.117) is a material balance equation for the implicit contributions to the wetting phase flow. Therefore, the overall time step calculation is conservative for the wetting phase.

The change of S_w from S_w^* to S_w^{n+1} also brings about changes in fluxes and accumulation of the non-wetting phase. The change in the accumulation term is $-(V_p/\triangle t) b_n^* (S_w^{n+1} - S_w^*)$, which is obtained from eqn. (5.115) by multiplying it by $a_i = (-b_n^*/b_w^*)_i$. Therefore adjusting the wetting phase solution according to eqn. (5.115) is equivalent to solving the following equation for the non-wetting phase:

$$\begin{aligned} &- a_i T'_{w_{i-1/2}} (\Delta_t S_w)_- + a_i T'_{w_{i+1/2}} (\Delta_t S_w)_+ + a_i [T^n_{w_{i-1/2}} P'_{c_{i-1}} \Delta_t S_{w_{i-1}} \\ &\quad - (T_{w_{i-1/2}} + T_{w_{i+1/2}})^n P'_{c_i} \Delta_t S_{w_i} + T^n_{w_{i+1/2}} P'_{c_{i+1}} \Delta_t S_{w_{i+1}}] \\ &= - \frac{V_p}{\Delta t} b_n^* (S_w^{n+1} - S_w^*)_i + a_i Q'_{ww_i} \Delta_t S_{w_i} = \frac{V_p}{\Delta t} b_n^* (S_n^{n+1} - S_n^*)_i + \Delta_t Q_{n_i} \end{aligned} \tag{5.118}$$

This can also be seen as follows: Since the second step of SEQ treats flow as *incompressible*, any change in the interblock flow of the wetting phase must be balanced by an opposite change in the non-wetting phase on a *reservoir volume* basis. For example, the adjustment of the flux $q_{i+½}$ in reservoir units, can be obtained from eqn. (5.115):

$$\Delta_t q_{i+1/2} = \frac{1}{b_{w_i}^*} [T'_{w_{i+1/2}} (\Delta_t S_w)_+ + T^n_{w_{i+1/2}} (P'_{c_{i+1}} \Delta_t S_{w_{i+1}} - P'_{c_i} \Delta_t S_{w_i})]$$

It has two parts: one due to relative permeability and the other due to capillary pressure change. Each change in flux is balanced independently by a corresponding term in eqn. (5.118), if this equation is divided by b_n^* to obtain units of reservoir volume. Similarly, the accumulation terms are also balanced.

The material balance for the non-wetting phase is now obtained by summing up eqns. (5.118) for each block. To do this, we have to consider the direction of flow. For flow from left to right (in the direction of increasing z), we get

$$
\begin{aligned}
&- \sum_{i=1}^{N-1} (a_{i+1} - a_i) \mathrm{T}'_{\mathrm{w}i+1/2} \Delta_t S_{\mathrm{w}i} \\
&\quad + \sum_{i=1}^{N-1} (a_{i+1} - a_i) \mathrm{T}^n_{\mathrm{w}i+1/2} (P'_{ci} \Delta_t S_{\mathrm{w}i} - P'_{ci+1} \Delta_t S_{\mathrm{w}i+1}) \\
&= \sum_{i=1}^{N} \frac{V_{pi}}{\Delta t} [(b_{\mathrm{n}} S_{\mathrm{n}})^{n+1} - (b_{\mathrm{n}} S_{\mathrm{n}})^*]_i + \sum_{i=1}^{N} \Delta_t Q_{\mathrm{n}i}
\end{aligned}
\tag{5.119}
$$

The right-hand side of this equation is the material balance expression, and therefore the left-hand side sums should be equal to zero. Unfortunately, this is guaranteed only for incompressible flow. Therefore:

> *The implicit step of the SEQ method does not generally satisfy material balance of the non-wetting phase. The error is proportional to the areal variation of* $b_{\mathrm{n}}/b_{\mathrm{w}}$.

The above shortcoming of the SEQ method was first pointed out by Coats *et al.* (1974). The material balance errors are normally negligible for oil-water systems, but can become objectionable for gas-water, and especially for three-phase systems, where the errors also depend on variations of R_{s}.

A second shortcoming of the method lies in the treatment of production terms. It is necessary for stability to include implicit production terms in the second (saturation) step. Again, since this step is incompressible, the sum of implicit changes in production rates must be zero on a reservoir volume basis:

$$
B_{\mathrm{w}} \Delta_t Q_{\mathrm{w}i} + B_{\mathrm{n}} \Delta_t Q_{\mathrm{n}i} = 0
$$

Therefore, if the rate of one phase is prescribed, it will not be maintained after the implicit step. This problem can be solved only by iteration, during which the rate used in the IMPES step is entered as the prescribed rate minus the implicit change during the last iteration. In our experience, two iterations are usually sufficient, but the computing times increase nearly proportionately with the number of iterations, and the method becomes less attractive.

5.6.2 Other Forms and Derivations

(a) The above derivation of the implicit saturation step can be repeated starting with the non-wetting phase equation (eqn. (5.111)). The resulting equation will have coefficients based on derivatives of T_n. The method will satisfy material balance of the non-wetting phase, but *not* for the wetting phase.

(b) It is easy to see that sequential formulation can be derived for other implicit treatments of transmissibilities as discussed in Section 5.5.2. For example, a fully implicit sequential method would require iterating such that $\mathbf{T}_w^n + \mathbf{T}'_w \triangle_t S_w = \mathbf{T}_w^{n+1}$. Experience with such methods has not been reported in the literature so far. It is interesting to note that without capillary pressure, the structure of the matrix $\mathbf{T}' - \mathbf{D}_s = \mathbf{T}'_s - \mathbf{D}_s$ is such that it can be transformed into a triangular matrix by re-ordering of unknowns. This is a consequence of upstream weighting. Then the equations can be solved point by point rather than simultaneously, even if the transmissibilities are fully implicit (Watts, 1972).

(c) The above remark points to the fact that the implicit saturation step is an approximation of the fractional flow equations. Indeed, SEQ can be derived directly from the equations presented in Chapter 2 (Section 2.5.2). In Section 5.4.1.2, we have given the approximation to the simple form of the saturation equation

$$-\boldsymbol{u} \,.\, \nabla f_w = \phi \frac{\partial S_w}{\partial t} + q_w - f_w q_T$$

with explicit transmissibilities as eqn. (5.58). This equation therefore represents the first step of the SEQ method and can be written as

$$Q_T(f'_{w_i} S_{w_i} - f'_{w_{i-1}} S_{w_{i-1}})^n = \frac{V_{p_i}}{\Delta t}(S^*_{w_i} - S^n_{w_i}) + Q^n_w - f^n_{w_i} Q_T$$

For the fully implicit step we can write the same equation implicitly:

$$Q_T(f'_{w_i} S_{w_i} - f'_{w_{i-1}} S_{w_{i-1}})^{n+1} = \frac{V_{p_i}}{\Delta t}(S^{n+1}_{w_i} - S^n_{w_i}) + Q^{n+1}_w - f^{n+1}_{w_i} Q_T$$

Subtracting these two equations gives

$$Q_T[f'_{w_i}(S^{n+1}_w - S^n_w)_i - f'_{w_{i-1}}(S^{n+1}_w - S^n_w)_{i-1}] = \frac{V_{p_i}}{\Delta t}(S^{n+1}_w - S^*_w)_i + \Delta_t Q_w \qquad (5.120)$$

which is eqn. (5.115) for this simplified case. Note that here the transmissibility is (since $Q_T = (T_n + T_w)_{i+½}\, \Delta\Phi_{i+½}$ from the solution of the pressure equation):

$$T'_{w_{i+1/2}} = (T_n + T_w)_{i+1/2} f'_{w_i} \Delta\Phi_{i+1/2} \tag{5.121}$$

while the transmissibility used in eqn. (5.115) is

$$T'_{w_{i+1/2}} = \frac{dT_w}{dS_w} \Delta\Phi_{i+1/2} \tag{5.122}$$

5.6.3 Numerical Results

In this section a comparison of SEQ and SS methods is presented. Some of the results are from Ko (1977) and additional results have been generated by Ko and the authors.

All of the results are obtained with one point upstream approximation, and $P_c = 0$ for the Buckley–Leverett problem. Note that it is not necessary to use a non-zero value of P_c with the SEQ method as was the case with the SS method.

We first note that the results for the case where transmissibilities are treated explicitly are identical with results obtained by the IMPES method, as the SEQ method actually reduces to IMPES (step 2 is omitted). As mentioned earlier, any SEQ method can be coded to include IMPES as an option.

All nonlinearities due to transmissibilities are handled by the linearised implicit method which corresponds to one iteration with Newton's method as discussed in Section 5.5. Figure 5.10 shows the results for the Buckley–Leverett problem solved by the tangent method and Fig. 5.11 shows the results for the chord method. For the larger time step the chord method is more stable. The difference in stability between the tangent and the chord methods is not as great as that shown in Figs. 5.6 and 5.7 for the SS method with semi-implicit treatment of transmissibilities. The higher stability of the chord method is accompanied by larger space truncation errors as indicated by the results for $\Delta t = 25$ days.

Ko (1977) has also presented results for various methods of handling the nonlinearity due to capillary pressure. He studied these methods on Test problem No. 1 with large capillary pressure. The P_c used was obtained by multiplying the capillary pressure of Fig. 2.9 by a factor of ten. Reference solution was computed using the IMPES method with a small time step. Figures 5.12 to 5.14 show the solutions at 1500 days obtained using explicit P_c, and implicit P_c using tangent and chord slope method on P'_c. In all cases,

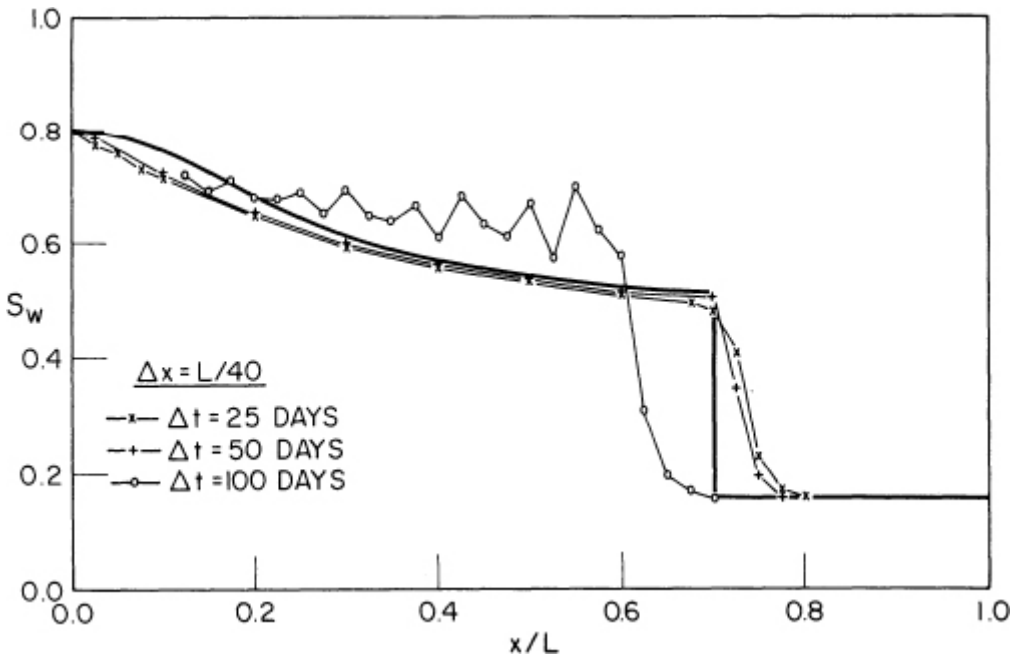


FIG. 5.10. Stability of SEQ method with linearised tangent slope transmissibilities, Test problem No. 1.

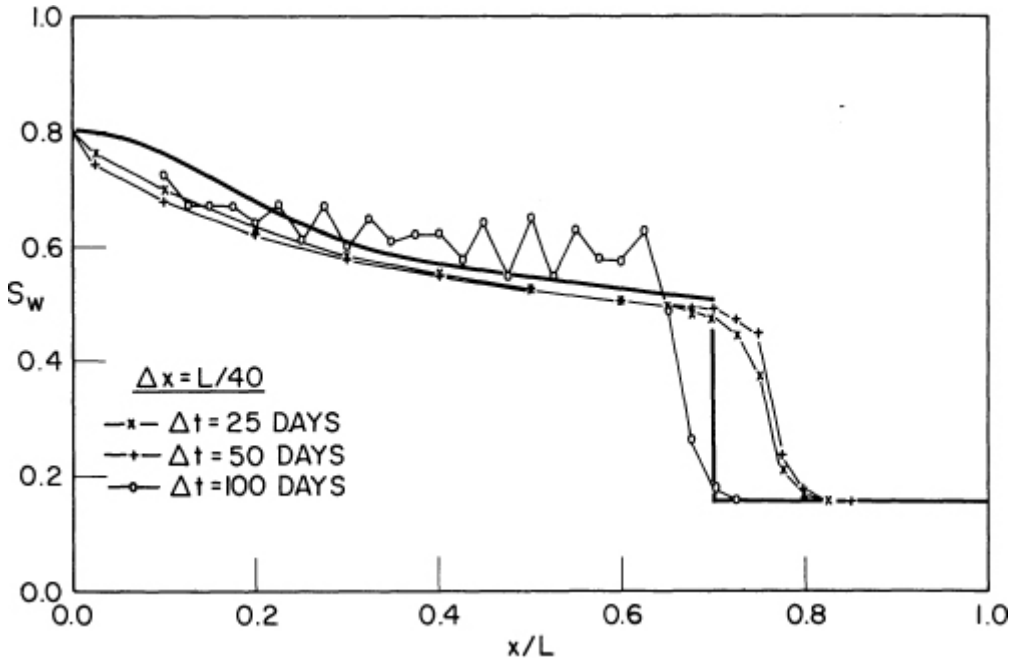


FIG. 5.11. Stability of SEQ method with linearised chord slope transmissibilities, δS_W = 0•32, Test problem No. 1.

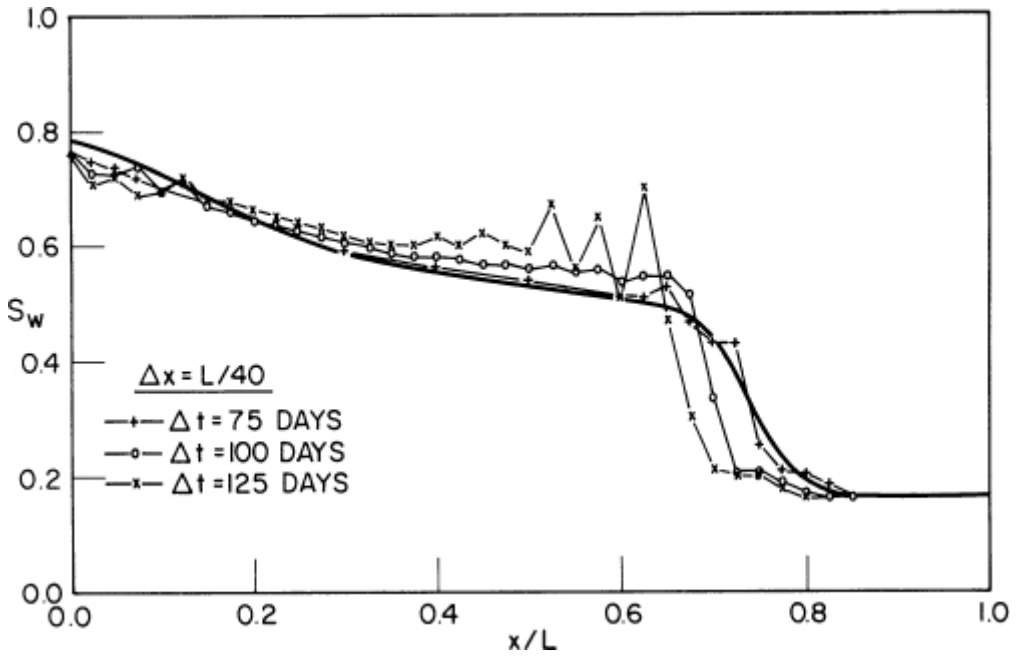


FIG. 5.12. Stability of SEQ method with explicit treatment of P_c for modified Test problem No. 1 with high P_c.

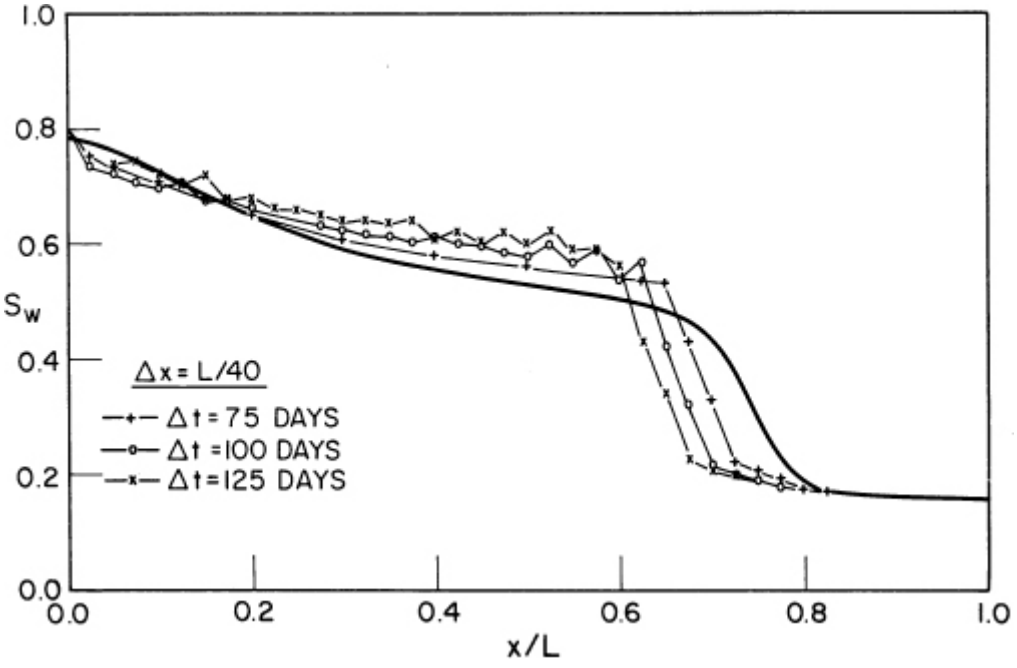


FIG. 5.13. Stability of SEQ method with tangent slope treatment of P_c for modified Test problem No. 1 with high P_c.

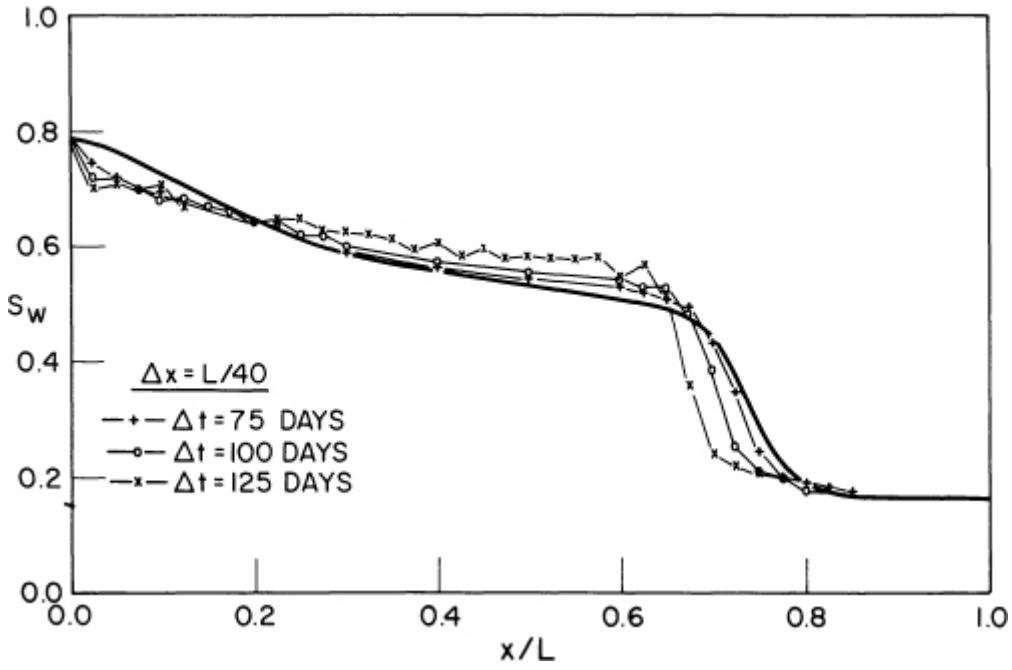


FIG. 5.14. Stability of SEQ method with chord slope treatment of P_c for modified Test problem No. 1 with high P_c.

transmissibilities were handled by the tangent slope method. The results show that:

(a) explicit treatment of P_c leads to stability problems; and
(b) the chord slope method gives better results than the tangent method.

However, when the chord method on transmissibilities is used, the tangent method of handling P_c is preferable to the chord slope method. This choice (chord on T, tangent on P_c) was also reported by Spillette *et al.* (1973), but without justification.

Two-dimensional results using SEQ method will be reported in Chapter 9 (Section 9.8).

5.6.4 SEQ Method for Three-phase Flow

Sequential approach offers even more variety in three-phase flow. Let us first consider the usual case when, in the implicit step, we solve simultaneously for two saturation changes.

Derivation of the method. As we have seen, one may arrive at different definitions of coefficients depending on how the equations are derived. While this is usually not critical for two-phase flow, it is not the case for three-phase flow. For example,

definition according to eqn. (5.122) may cause stability problems in a three-phase situation.

The implicit adjustments to interblock rates denoted by $\Delta_t q_l$ must satisfy

$$\Delta_t q_{\mathrm{w}} + \Delta_t q_{\mathrm{o}} + \Delta_t q_{\mathrm{g}} = 0 \tag{5.123}$$

independently on each block boundary. The changes $\Delta_t q_l$ are expressed as

$$\Delta_t q_l = (\mathrm{T}_l^{n+1} - \mathrm{T}_l^n)\,\Delta\Phi_l \cong \mathrm{T}'_{l_{\mathrm{w}}}\,\Delta_t S_{\mathrm{w}} + \mathrm{T}'_{l_{\mathrm{g}}}\,\Delta_t S_{\mathrm{g}} \tag{5.124}$$

where we have chosen $\Delta_t S_{\mathrm{w}}$ and $\Delta_t S_{\mathrm{g}}$ as variables. Since these two are independent, the definitions of T'_l must satisfy

$$\sum_l \mathrm{T}'_{l\mathrm{w}} = 0 \qquad \text{and} \qquad \sum_l \mathrm{T}'_{l\mathrm{g}} = 0 \tag{5.125}$$

which are obtained from eqn. (5.124) by setting $\Delta_t S_{\mathrm{g}} = 0$ and $\Delta_t S_{\mathrm{w}} = 0$, respectively. The problem now is to define two of, say $\mathrm{T}'_{l\mathrm{w}}$ such that the third one defined by eqn. (5.125) is also physically reasonable. The definitions which satisfy eqn. (5.125) are obtained from the fractional flow form of q_l:

$$q_l = q_{\mathrm{T}} \frac{\lambda_l}{\lambda_{\mathrm{T}}} = q_{\mathrm{T}} f_l$$

where q_l, q_{T} are volumetric flow rates (in units of reservoir volume). Therefore

$$\mathrm{T}'_{l_{\mathrm{w}}} = q_{\mathrm{T}} f'_{l_{\mathrm{w}}} \qquad \mathrm{T}'_{l_{\mathrm{g}}} = q_{\mathrm{T}} f'_{l_{\mathrm{g}}} \tag{5.126}$$

where

$$\begin{aligned}
f'_{\mathrm{w_w}} &= \frac{1}{\lambda_{\mathrm{T}}}\left(k\frac{k'_{\mathrm{rw}}}{\mu_{\mathrm{w}}} - \frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{T}}}\lambda'_{\mathrm{T_w}}\right) \qquad & f'_{\mathrm{w_g}} &= -\frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{T}}^2}\lambda'_{\mathrm{T_g}} \\
f'_{\mathrm{o_w}} &= \frac{1}{\lambda_{\mathrm{T}}}\left(k\frac{k'_{\mathrm{ro_w}}}{\mu_{\mathrm{o}}} - \frac{\lambda_{\mathrm{o}}}{\lambda_{\mathrm{T}}}\lambda'_{\mathrm{T_w}}\right) \qquad & f'_{\mathrm{o_g}} &= \frac{1}{\lambda_{\mathrm{T}}}\left(k\frac{k'_{\mathrm{ro_g}}}{\mu_{\mathrm{o}}} - \frac{\lambda_{\mathrm{o}}}{\lambda_{\mathrm{T}}}\lambda'_{\mathrm{T_g}}\right) \\
f'_{\mathrm{g_w}} &= -\frac{\lambda_{\mathrm{g}}}{\lambda_{\mathrm{T}}^2}\lambda'_{\mathrm{T_w}} \qquad & f'_{\mathrm{g_g}} &= \frac{1}{\lambda_{\mathrm{T}}}\left(k\frac{k'_{\mathrm{rg}}}{\mu_{\mathrm{g}}} - \frac{\lambda_{\mathrm{g}}}{\lambda_{\mathrm{T}}}\lambda'_{\mathrm{T_g}}\right)
\end{aligned} \tag{5.127}$$

and

$$\lambda'_{\mathrm{T_w}} = k\left(\frac{k'_{\mathrm{rw}}}{\mu_{\mathrm{w}}} + \frac{k'_{\mathrm{ro_w}}}{\mu_{\mathrm{o}}}\right) \qquad \lambda'_{\mathrm{T_g}} = k\left(\frac{k'_{\mathrm{ro_g}}}{\mu_{\mathrm{o}}} + \frac{k'_{\mathrm{rg}}}{\mu_{\mathrm{g}}}\right)$$

since $k_{\mathrm{rw}} = f(S_{\mathrm{w}})$ and $k_{\mathrm{rg}} = f(S_{\mathrm{g}})$. We should note that

$$k'_{\mathrm{ro}l} \equiv \frac{\partial k_{\mathrm{ro}}}{\partial S_l} \qquad l = \mathrm{w},\ \mathrm{g}$$

The derivatives in eqns. (5.127) must be evaluated at the upstream saturation (i.e., either at i or $i + 1$ for transmissibilities at i+½) for each phase. Note that eqn. (5.127) will also satisfy eqn. (5.125) for the case of countercurrent flow.

We may now observe that formula (5.121) is a special case of eqn. (5.127) and also that eqn. (5.122) is obtained from eqn. (5.127) by *neglecting the change in total mobility*. The saturation equations can be expressed in matrix form of eqn. (5.116); however, now the elements are 2 × 2 matrices.

A different method of calculating the implicit flows was used by Coats (1976). He defines $\mathrm{T}'_{l\mathrm{m}}$ using *chords* on the fractional flow equation between saturations S_l^n and an estimate S_l^k of S_l^{n+1}. For example, if S_{w} and S_{g} are the unknowns,

$$\mathrm{T}'_{l_{\mathrm{w}}} = [q_l(S_{\mathrm{w}}^k, S_{\mathrm{g}}^n) - q_l(S_{\mathrm{w}}^n, S_{\mathrm{g}}^n)]/(S_{\mathrm{w}}^k - S_{\mathrm{w}}^n) \tag{5.128}$$

where again, S_l^k and S_l^n must be the upstream values. For example, if water flows from i to $i + 1$ and gas in the opposite direction, then

$$q_{\mathrm{w}_{i+1/2}} = q_{\mathrm{w}}(S_{\mathrm{w}_i}, S_{\mathrm{g}_{i+1}}) \qquad q_{\mathrm{g}_{i+1/2}} = q_{\mathrm{g}}(S_{\mathrm{w}_i}, S_{\mathrm{g}_{i+1}})$$

If we decide to solve the water and oil equations instead, we must determine the upstream oil saturation and write eqn. (5.128) for q_{w} and q_{o} .

The advantage of eqn. (5.128) is that by iterating on these coefficients (without resolving the pressure equation) one obtains *a sequential formulation with fully implicit transmissibilities*.

Variations of the SEQ method. Variations of the basic method given above are possible along these lines:

(a) Handling of implicit transmissibilities in the saturation step. The transmissibilities can be handled by a tangent method (as indicated by eqn. (5.127) or by a chord method (eqn. (5.128)). Fully implicit treatment via updating the chords in

eqn. (5.128) increases stability but also computer work per time step. Implicit treatment of $\mathbf{\Delta\Phi}$ in the $\mathbf{T'}$ matrix leads to non-linear terms involving the product of saturation and capillary pressure derivatives.

(b) Choice of equations to be solved in the saturation step will affect material balance, which will not be satisfied for the third phase. As long as two equations are solved simultaneously, this choice has little effect on the stability of the method.

(c) The saturation equations can be uncoupled by solving first implicitly for $\Delta_t S_w$, followed by implicit updating of S_g. This involves solving three sets of equations of the same size and is thus faster than solving simultaneously for S_w and S_g, but it suffers some loss of stability as a result of the uncoupling of saturations.

Another possibility, suggested by Coats (1976a) is to solve simultaneously for pressures and implicit gas saturations, followed by an implicit water saturation update. Such a scheme eliminates the material balance errors associated with the variations of R_s, which can be very serious for problems such as gas injection in undersaturated reservoirs (Stright *et al.,* 1977) or steamflooding (Coats, 1976a). As reported by Coats, simultaneous p and S_g solution is more stable than simultaneous S_w and S_g solution for the majority of the black-oil simulation problems, in which the major source of instability is gas–oil and oil–water interactions. Only in the case of gas-water interactions is the $S_w - S_g$ method superior.

5.6.5 Discussion

We have presented basic forms of the SEQ method for two- and three-phase flow and have discussed several possible variations. In general, sequential approach requires less computer effort than the simultaneous solution, but it produces equations which are not conservative for all phases and are not as stable as SS equations. In our experience, SEQ method is best suited for problems of 'intermediate' difficulty which cannot be solved with explicit transmissibilities and/or P_c, but do not require the fully implicit treatment. The evaluation of various forms of SEQ is still incomplete at this writing.

5.7 TREATMENT OF PRODUCTION TERMS

We have seen, in Section 3.4 of Chapter 3, that for single-phase flow, boundary conditions can be conveniently represented as source/sink terms in the finite-difference equations. This concept is also directly applicable to multiphase flow. The actual boundary conditions are multidimensional, and several aspects of this treatment can only be experienced in a 2-D setting. For this reason, detailed treatment of boundary

conditions is postponed to Chapter 9 (Section 9.4). Here, we will only discuss the one-dimensional aspects of the case when the production terms represent flow across the face of a block (Fig. 5.15). The case when a production term represents a well inside a relatively large block requires the use of methods described in Chapter 7 (Section 7.7), to account for the high level of approximation involved. However, once this is done, the equations for the production terms can be treated in the same way as described here.

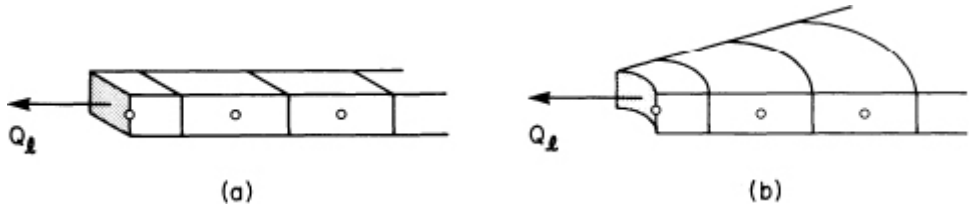


FIG. 5.15. Flow from boundaries.

5.7.1 Differential Form of Boundary Conditions

The flow of all phases at the outlet of a porous medium is assumed to satisfy Darcy's Law:

$$q_l = -\lambda_l\left(\frac{\partial p_l}{\partial x} - \gamma_l\frac{\partial z}{\partial x}\right) \qquad l = \text{o, w, g} \tag{5.129}$$

where q_l is the flow rate per unit area in reservoir volume units.

Therefore, if the pressures and saturations at the boundary are *known*, these equations can be used to define q_l. The problem of specifying boundary condition is to distribute the rate without knowing the solution.

Boundary effects. Suppose that we have fixed the boundary conditions in two-phase oil-water flow by specifying q_w and q_o. It is easy to see that in the presence of capillary pressure, the fractional flow condition

$$\frac{q_w}{q_o} = \frac{\lambda_w\left(\dfrac{\partial p_w}{\partial x} - \gamma_w\dfrac{\partial z}{\partial x}\right)}{\lambda_o\left(\dfrac{\partial p_o}{\partial x} - \gamma_o\dfrac{\partial z}{\partial x}\right)}$$

could be satisfied by more than one combination of $S_w\ \partial p_w/\partial x$ and $\partial p_o/\partial x$. This shows that the solution of multiphase flow equations is not sufficiently determined by the specification of q_w and q_o. The additional required condition is obtained by considering the physics of the flow at the boundary. Physically, simultaneous flow of two phases at the outlet can only take place if the capillary pressure decreases to the value of capillary pressure P_{co} outside the porous media, or to the closest possible value. This value may be the P_c in the wellbore or in a fracture etc. and it is usually close to zero. This phenomenon is called the *outlet* (*or boundary*) *effect* (Collins, 1961; Section 6.10). The saturation at the outlet end must reach the value S_{wo} corresponding to P_{co}, before the second phase can flow out. Figure 5.16 shows this for two types of P_c curves, assuming $P_{co} = 0$. If we start with the

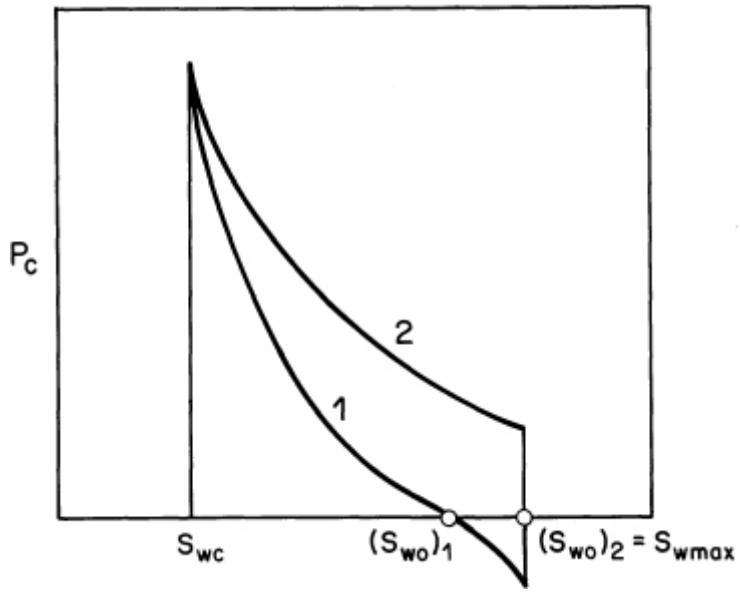


FIG. 5.16. Definition of S_{wo} for two different P_c curves.

conditions where only oil is produced, water saturation will increase at the outlet, but water will not flow out until S_w reaches S_{wo}.

After water breakthrough, *the saturation at the boundary remains constant.* A detailed investigation of saturation and pressure changes near the boundary is found in Collins (1961) for linear flow and in Settari and Aziz (1974a) for radial flow. The appropriate equations are also derived by Sonier *et al.* (1973). This analysis shows that *the saturation has a non-zero gradient at the outlet*, which becomes infinite if k_{ro} at S_{wo} is zero (curve 2 in Fig. 5.16). Figure 5.17 shows the result of 1-D calculation of saturations near the well for different conditions. These results are expressed in

dimensionless co-ordinates S_w and $\xi = (r/r_w)^c$ where $c = -q_o/2\pi k$ and $q_o < 0$ for production. The following conclusions may be drawn from these results:

1. The zone influenced by the outlet effect decreases with increasing q_o and q_w/q_o.

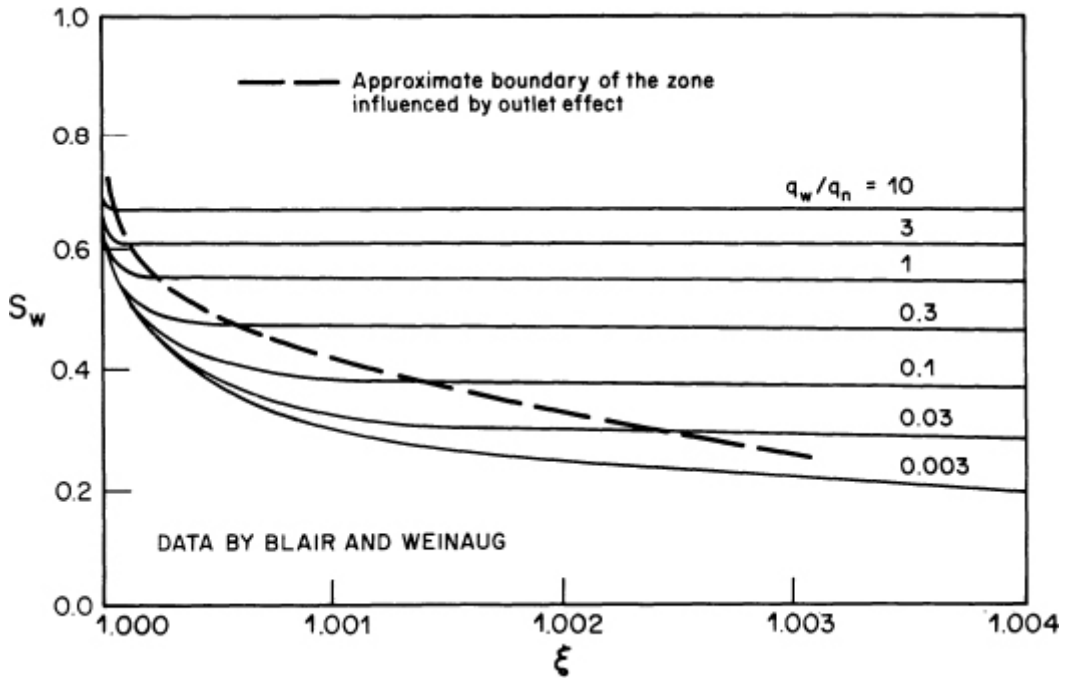


FIG. 5.17. Saturation at the outlet in a 1-D radial flow (from Settari and Aziz, 1974a).

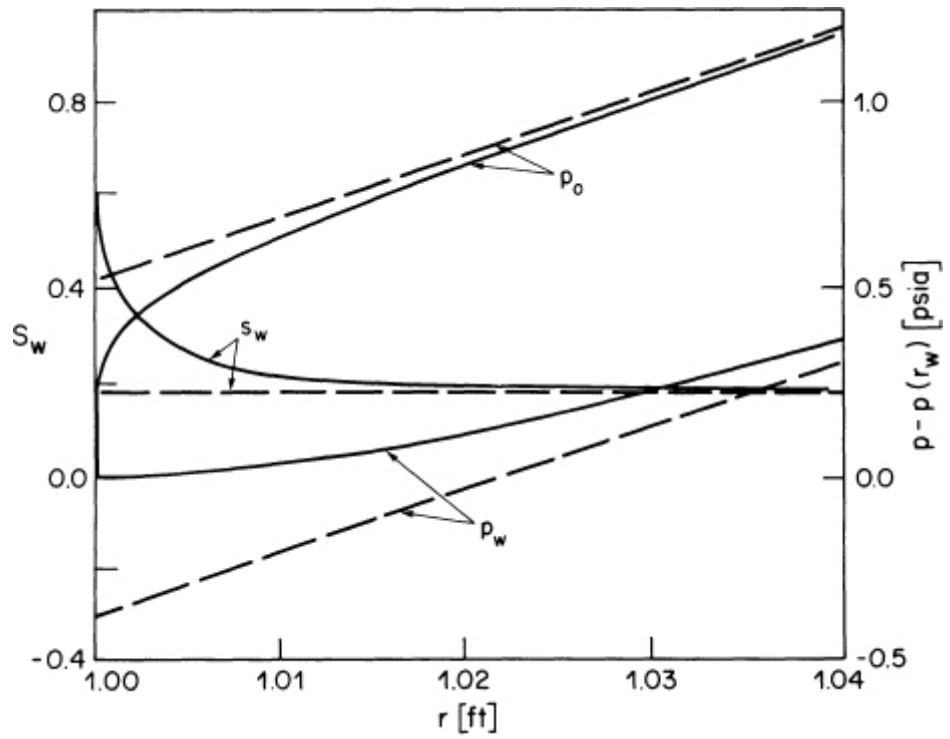


FIG. 5.18. Calculated pressure and saturation distribution at the well with and without the outlet effect (from Settari and Aziz, 1974a).

2. The extent of the zone is usually of the order of inches or smaller and therefore can be neglected on a reservoir scale. However, the zone of influence can become large when the flow is dominated by capillary forces (imbibition flow).

Similarly, a physical condition at the inlet boundary requires that the *saturation gradient be zero for simultaneous injection of two phases.*

In reservoir simulation, the outlet effect is as a rule neglected and zero saturation gradient is also assumed for the production boundary. The assumption $\partial S_w/\partial x = 0$ makes the solution unique, but different from the true solution. This is shown in Fig. 5.18, where the dashed lines represent the solution obtained by using $\partial S_w/\partial x = 0$. Note that this solution yields *two* pressures at the boundary and the true pressure lies between these values.

The above simplification of the boundary effects is adequate for most reservoir problems. Possible exceptions are tight reservoirs with very high capillary pressures and fractured reservoirs, in which capillary flow is important for the transfer of mass between the fissures and the matrix.

Prescribed conditions. The actual conditions imposed on the production boundary are one of the following:

(a) Flow rate of one phase (usually oil)
(b) Liquid rate (oil + water)
(c) Total rate (oil + water + gas)
(d) Constant pressure at the boundary.

The first three conditions are in fact constraints rather than boundary conditions; this is better seen in two dimensions in Chapter 9 (Section 9.4.1). The above rates can be imposed either in terms of volume at reservoir condition/time, or STC volume/time (equivalent to a mass rate). Since the rates Q_l in difference equations are always in STC volume/time, the former must be converted using B_l values and R_s for the current conditions at the boundary.

5.7.2 Discretisation of Boundary Conditions

5.7.2.1 *Rate Conditions*

Discretisation of eqn. (5.129) at the boundary $i = 1$ gives

$$Q_l = -A\left(k\frac{k_{rl}}{B_l\mu_l}\right)_1 \frac{1}{\Delta x_{1+1/2}}[p_{l_2} - p_{l_1} - \gamma_{l_{1+1/2}}(z_2 - z_1)] \\ = -\mathrm{T}Q_{l_1}\Delta\Phi_{l_{1+1/2}} \tag{5.130}$$

where the pressure difference was approximated between grid points 1 and 2.

This equation has the same form as the terms for interblock flow rates *except* that the transmissibility $\mathrm{T}Q_{li}$ is different from $\mathrm{T}_{li+1/2}$ because it is evaluated *at* the point i rather than according to some upstream weighting formula.

We can now consider various possible ways of specifying production rates:

(a) Specified oil rate $Q = Q_o$ in STC units. Since

$$Q_o = -\mathrm{T}Q_o\Delta\Phi_o$$

we have

$$Q_w = \frac{\mathrm{T}Q_w\Delta\Phi_w}{\mathrm{T}Q_o\Delta\Phi_o}Q_o \tag{5.131}$$

$$Q_g = \frac{\mathrm{T}Q_g\Delta\Phi_g}{\mathrm{T}Q_o\Delta\Phi_o}Q_o$$

(b) Specified liquid rate Q_L in STC units. Since

$$Q_L = TQ_o \Delta\Phi_o + TQ_w \Delta\Phi_w \tag{5.132}$$

$$Q_l = \frac{TQ_l \Delta\Phi_l}{TQ_o \Delta\Phi_o + TQ_w \Delta\Phi_w} Q_L \qquad l = o, w, g$$

(c) Specified total rate in reservoir units (voidage rate). The voidage rate is

$$Q_{VT} = \sum_l B_l TQ_l \Delta\Phi_l$$

and the individual (STC) rates will be

$$Q_l = \frac{TQ_l \Delta\Phi_l}{\sum_l B_l TQ_l \Delta\Phi_l} Q_{VT} \tag{5.133}$$

It is easy to see how to modify these formulae in case that the rates are specified in RC as opposed to STC units and vice versa.

All three formulae show that the rates Q_l which are required in the finite difference equations, should be distributed according to *transmissibilities and potential difference.* In order that equations such as eqn. (5.133) may be incorporated in the matrix equation, the potential differences must be *evaluated explicitly, i.e.,* $\mathbf{\Delta\Phi} = \mathbf{\Delta\Phi}^n$ or some estimate $\mathbf{\Delta\Phi}^k$ of $\mathbf{\Delta\Phi}^{n+1}$. *However, the treatment of TQ should be consistent with the treatment of interblock transmissibilities in the solution technique used. These points will be discussed below.*

The rate calculation according to eqns. (5.131) to (5.133) is sometimes called 'allocation of production according to potentials'. This formulation is correct, but it can cause stability problems, especially when one tries to iterate on $\mathbf{\Delta\Phi}$ to obtain $\mathbf{\Delta\Phi}^{n+1}$ in the formulae.

A simplified procedure can be obtained by assuming

$$\Delta\Phi_o = \Delta\Phi_w = \Delta\Phi_g \tag{5.134}$$

Such an assumption is justified if the capillary forces are small and $\gamma_l dz/dx = 0$ or negligible. This method, called 'allocation according to transmissibilities' gives the following counterparts to eqn. (5.131) to (5.133):

$$Q_w = \frac{TQ_w}{TQ_o} Q_o \qquad Q_g = \frac{TQ_g}{TQ_o} Q_o \tag{5.135}$$

for oil production specified,

$$Q_l = \frac{TQ_l}{TQ_o + TQ_w} Q_L \qquad l = o, w, g \tag{5.136}$$

for total liquid rate specified, and

$$Q_l = \frac{TQ_l}{\sum_l B_l TQ_l} Q_{VT} \qquad l = o, w, g \tag{5.137}$$

for total rate in reservoir units.

Let us now briefly consider the implementation of the 'outlet effect'. Suppose that oil production is specified in an oil–water system and $S_{w1} < S_{wo}$ initially (Fig. 5.19a). Then according to the discussion in Section 5.7.1, $Q_w = 0$. When the saturation of the first block reaches the value of S_{wo}, water breakthrough occurs (Fig. 5.19b). From this time on, the saturation of this block *remains constant* and this is the condition which determines the water production rate. The condition of constant saturation can be easily imposed. If the equations are written in terms of p_o and S_w, the water-phase equation for the production block is simply replaced by the equation

$$S_{w_1} = S_{wo} \tag{5.138a}$$

In case that p_o and p_w are used, the water-phase equation is replaced by

$$p_{o_1} - p_{w_1} = P_c(S_{wo}) \tag{5.138b}$$

The water production rate, Q_w, to satisfy the above condition is then determined by substituting the solution in the water equation in a manner similar to computing saturations in the IMPES method.

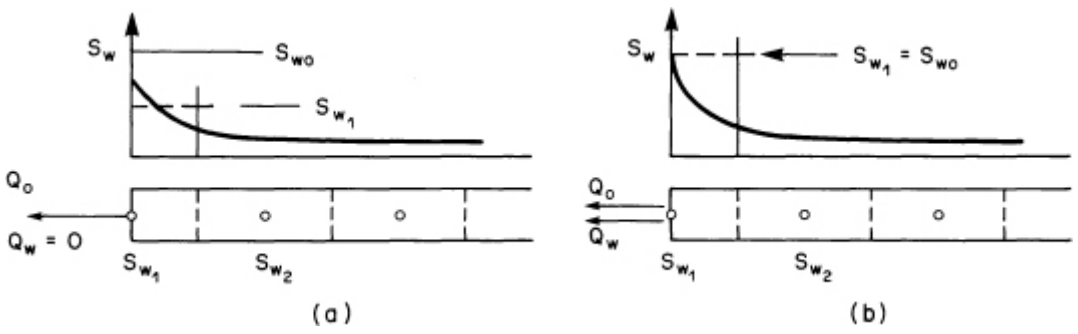


FIG. 5.19. Saturation profile before (a) and after (b) breakthrough.

Because S_{w1} is an average saturation of the block, the actual saturation on the boundary will be in our case higher than S_{w1} as shown in Fig. 5.19a. Therefore this method will predict longer than actual breakthrough time. Improvement (apart from decreasing the block size) is possible by calculating the saturation S_{w1} as an average of the analytical solution within the boundary block, based on the q_w/q_o computed from S_{w2} (assumed to be beyond the zone influenced by the outlet effect).

5.7.2.2 *Pressure Conditions*

In dealing with pressure conditions, it is illustrative to consider both point-distributed and block-centred grid. For simplicity, we will omit the gravity terms.

In a point-distributed grid, Fig. 5.20a, the *pressure of only one of the phases can be specified*, and the production rates of the other phases are defined in relation to the rate of this phase. If we choose oil pressure, then the oil equation is replaced by

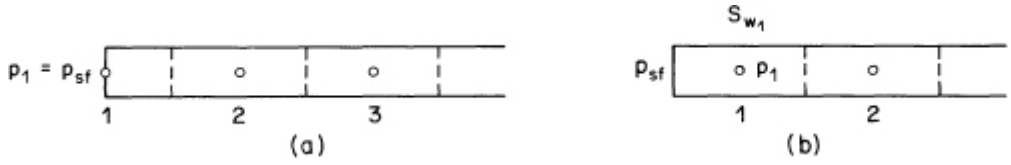


FIG. 5.20. Specification of pressure at boundaries. (a), Point-distributed grid; (b), block-centred grid.

$$p_{o_1} = p_{sf} \text{ (sandface pressure)} \tag{5.139}$$

and the resulting oil rate can be calculated from the equation for this point:

$$Q_o = T_{1+1/2}(p_{o_2} - p_{sf}) - \frac{V_p}{\Delta t}\Delta_t(S_o b_o)_1$$

Then the water and gas rates are determined from formulae (5.135):

$$Q_l = \frac{\mathrm{T}Q_l}{\mathrm{T}Q_o} Q_o \qquad l = \mathrm{w},\ \mathrm{g} \tag{5.140}$$

The rates Q_w and Q_g are then used in the appropriate difference equations, to solve for S_{wl} and S_{gl}.

If we want to consider the outlet effect, the approach is quite similar to what was described earlier for the case of specified rate. Before breakthrough, the conditions will be $p_{o1} = p_{sf}$, $Q_{\underline{w}} = 0$. After break-through, $p_{o1} = p_{sf}$, and $p_{w1} = p_{sf} - P_c(S_{wo})$, i.e., *both* pressures are fixed. The rates are then obtained by substituting the solution in the appropriate difference equations. The gas equation is handled in the same way.

Let us now consider the block-centred grid (Fig. 5.20b). The rates of all phases are given by

$$Q_l = \mathrm{T}_{l1\,2}(p_{l_1} - p_{l_{sf}}) \qquad l = \mathrm{o},\ \mathrm{w},\ \mathrm{g} \tag{5.141}$$

where the coefficient $T_{l1\,2}$ represents the transmissibility between the first block centre and the boundary. In this case consideration of the outlet effect oased on the saturation S_{w1} is not recommended, unless some kind of integration as described in the previous section is performed. In most cases we can assume that $p_{lsf} = p_{sf}$ for all phases. The expression (5.141) is easily incorporated in the matrix equation, by adding $T_{l1\,2}$ to the diagonal term and $T_{l1\,2}\,p_{sf}$ to the right-hand side vector. The rates are then obtained by substituting the solution p_{l1} back into eqn. (5.141).

The eqn. (5.141) can also be used for a well located at an interior node *i*. In this case it is written as

$$Q_l = WI_l(p_{l_i} - p_{wf}) \tag{5.142}$$

when p_{wf} is the flowing well pressure and WI_l is the productivity index for the phase l which can be calculated from the single-phase productivity index for the well and $(k_{rl}/\mu_l B_l)$ at the well.

5.7.2.3 *Time Approximation of Boundary Conditions*

Generally, the treatment of boundary conditions should be consistent with or more implicit than the treatment of saturations and transmissibilities. For example, for an IMPES method, the rate conditions (5.135) will be explicit:

$$Q_l = Q_l^{n+1} = \frac{\mathrm{T}Q_l^n}{\mathrm{T}Q_\mathrm{o}^n} Q_\mathrm{o} \qquad l = \mathrm{w},\ \mathrm{g}$$

while the pressure condition (5.141) will have an implicit pressure term:

$$Q_l = Q_l^{n+1} = \mathrm{T}_{l_{1,2}}(p_{l_1}^{n+1} - p_\mathrm{sf}) = \mathrm{T}_{l_{1,2}}(p_{l_1}^n - p_\mathrm{sf}) + \mathrm{T}_{l_{1,2}} \Delta_t p_{l_1}$$

which can be written as

$$Q_l^{n+1} = Q_l^n + Q'_{l_p} \Delta_t p_{l_1} \tag{5.143}$$

For an SS method with linearised implicit transmissibilities, the rates must also be treated implicitly for the rate condition,

$$Q_l = \left[\left(\frac{\mathrm{T}Q_l}{\mathrm{T}Q_\mathrm{o}} \right)^n + \left(\frac{\mathrm{T}Q_l}{\mathrm{T}Q_\mathrm{o}} \right)'_\mathrm{w} \Delta_t S_\mathrm{w} + \left(\frac{\mathrm{T}Q_l}{\mathrm{T}Q_\mathrm{o}} \right)'_\mathrm{g} \Delta_t S_\mathrm{g} \right] Q_\mathrm{o}$$

or

$$Q_l^{n+1} = Q_l^n + Q'_{l\mathrm{w}} \Delta_t S_\mathrm{w} + Q'_{l\mathrm{g}} \Delta_t S_\mathrm{g} \tag{5.144}$$

The derivatives in expressions of this type can be evaluated in different ways. We can assume for example, that $\mathrm{T}Q_\mathrm{o}$ does not change over the time step; then

$$\left(\frac{\mathrm{T}Q_l}{\mathrm{T}Q_\mathrm{o}} \right)'_\mathrm{w} \cong \frac{(\mathrm{T}Q_l)'_\mathrm{w}}{\mathrm{T}Q_\mathrm{o}^n} \tag{5.145}$$

Such approximations can be used when the rate of one phase is specified. When the total rate is specified, the derivatives such as

$$\left(\frac{\mathrm{T}Q_l}{\sum_l \mathrm{T}Q_l} \right)'_\mathrm{w} = \left(\frac{\mathrm{T}Q_l}{\mathrm{T}Q_\mathrm{T}} \right)'_\mathrm{w}$$

cannot be simplified in the manner of eqn. (5.145) because the total rate condition at $n + 1$ level would not necessarily be satisfied. In this case, one can use the formulae (5.127) listed in Section 5.6.4:

$$\left(\frac{\mathrm{T}Q_\mathrm{w}}{\mathrm{T}Q_\mathrm{T}} \right)'_\mathrm{w} = \frac{1}{\mathrm{T}Q_\mathrm{T}} \left[(\mathrm{T}Q_\mathrm{w})'_\mathrm{w} - \frac{\mathrm{T}Q_\mathrm{w}}{\mathrm{T}Q_\mathrm{T}} (\mathrm{T}Q_\mathrm{T})'_\mathrm{w} \right], \text{ etc.}$$

which have the property that

$$\sum_l Q'_{lw}\,\Delta_t S_w = \sum_l Q'_{lg}\,\Delta_t S_g = 0$$

The pressure conditions can be treated implicitly by either method since all rates are variable.

The most general expression is

$$Q_l^{n+1} = Q_l^n + Q'_{lw}\,\Delta_t S_w + Q'_{lg}\,\Delta_t S_g + Q'_{lp}\,\Delta_t p \tag{5.146}$$

where some or all of the implicit terms may be zero.

EXERCISES

Exercise 5.1

Show that for $P_c \to 0$ the SS method becomes singular.

Solution Outline

A typical block element of matrix **D** is

$$\mathbf{D_i} = \mathbf{D_{pi}} + \frac{V_{pi}}{\Delta t} S'_{wi} \begin{bmatrix} b_w & -b_w \\ -b_n & b_n \end{bmatrix} = \mathbf{D_{pi}} + \mathbf{D_{si}}$$

where $\mathbf{D_{pi}}$ contains the compressibility terms. Dividing both equations for block i by S'_{wi} gives

$$(\hat{\mathbf{T}} - \hat{\mathbf{D}}_p - \hat{\mathbf{D}}_s)(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = -\hat{\boldsymbol{R}}^n \tag{A}$$

where elements of $\hat{\mathbf{D}}_s$ are $\hat{\mathbf{D}}_{si} = \mathbf{D}_{si}/S'_{wi}$. When $\max \| P'_c \| \to 0$, $\hat{\mathbf{T}}$, $\hat{\mathbf{D}}_p \to 0$, $\boldsymbol{R}^n \to 0$ and (A) in the limit reduces to a singular problem

$$-\hat{\mathbf{D}}_s(\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) = \boldsymbol{0} \tag{B}$$

Exercise 5.2

Derive the alternate form of the gas equation for the SS method, starting with

(*a*) *Difference eqns.* (5.25) *and* (5.26)
(*b*) *Differential equations* (5.2a)
(*c*) *Discuss the nature of the gas equation derived in part* (*b*) *for the case of no free gas. Consider cases of constant and variable bubble point reservoirs.*

Solution Outline

(a) Assume that $R_{si+\frac{1}{2}}$ in eqn. (5.26) is expressed as

$$R_{s\,i+1/2} = \tfrac{1}{2}(R_{s\,i+1} + R_{s\,i})^n = R^n_{s\,i} + \tfrac{1}{2}(R_{s\,i+1} - R_{s\,i})^n \tag{A}$$

Expand the terms in the gas equation as

$$\Delta[R_s T_o(\Delta p_o - \gamma_o \Delta z)] = R^n_s \Delta T_o(\Delta p_o - \gamma_o \Delta z) + \Delta R^n_s T_o(\Delta p_o - \gamma_o \Delta z)$$

$$\Delta_t(R_s \phi b_o S_o) = R^n_s \Delta_t(\phi b_o S_o) + (\phi b_o S_o)^{n+1} \Delta_t R_s$$

and subtract the oil equation multiplied by

$$\Delta T_g(\Delta p_g - \gamma_g \Delta z) + \Delta R^n_s T_o(\Delta p_o - \gamma_o \Delta z) \tag{B}$$

$$= \frac{V_i}{\Delta t}[\Delta_t(\phi b_g S_g) + (\phi b_o S_o)^{n+1} \Delta_t R_s]_i + Q_{g_i}$$

where

$$\Delta R^n_s T_o(\Delta p_o - \gamma_o \Delta z) \equiv \tfrac{1}{2}[(R_{s_{i+1}} - R_{s_i})(T_o(\Delta p_o - \gamma_o \Delta z))_{i+1/2}$$

$$+ (R_{s_i} - R_{s_{i-1}})(T_o(\Delta p_o - \gamma_o \Delta z))_{i-1/2}]$$

(b) Expand the terms in eqn. (5.2a) as

$$\frac{\partial}{\partial x}\left[R_s \lambda_o\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial z}{\partial x}\right)\right] = R_s \frac{\partial}{\partial x}\left[\lambda_o\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial z}{\partial x}\right)\right] + \frac{\partial R_s}{\partial x}\lambda_o\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial z}{\partial x}\right)$$

$$\times \frac{\partial}{\partial t}(\phi R_s b_o S_o) = R_s \frac{\partial}{\partial t}(\phi b_o S_o) + \phi b_o S_o \frac{\partial R_s}{\partial t}$$

This gives the gas equation in the form

$$\frac{\partial}{\partial x}\left[\lambda_g\left(\frac{\partial p_g}{\partial x} - \gamma_g \frac{\partial z}{\partial x}\right)\right] + \frac{\partial R_s}{\partial x}\lambda_o\left(\frac{\partial p_o}{\partial x} - \lambda_o \frac{\partial z}{\partial x}\right) \tag{C}$$

$$= \frac{\partial}{\partial t}(\phi b_g S_g) + \phi b_o S_o \frac{\partial R_s}{\partial t} + q_g$$

Show that, if the second term on the left side of (C) is approximated by discretising

$$\frac{1}{2}\left[\frac{\partial R_s}{\partial x}\lambda_o\left(\frac{\partial p_o}{\partial x}-\gamma_o\frac{\partial z}{\partial x}\right)\right]_{i+1/2}+\frac{1}{2}\left[\frac{\partial R_s}{\partial x}\lambda_o\left(\frac{\partial P_o}{\partial x}-\gamma_o\frac{\partial z}{\partial x}\right)\right]_{i-1/2}$$

the finite difference equation is identical with (B).

Exercise 5.3

Formulate the SS difference equations for:

(a) *Two-phase flow in terms of* $\bar{p} = p_n + p_w$ *and* $P_c = p_n - p_w$ (*Douglas et al.*, 1959). *Discuss the form of matrices* **T** *and* **D**.
(b) *Three-phase flow in terms of* p_o, S_w *and* S_g, *and* p_o, P_{cow} *and* P_{cog}.

Exercise 5.4

Formulate the two-phase equations in terms of pressure and the mass of the wetting phase expressed as:

(a) *moles per unit volume*
(b) *total density.*

Discuss the advantages and disadvantages of these formulations.

Exercise 5.5

Formulate the IMPES method for three-phase flow in terms of p_o, S_w, S_g *by*

(a) *starting with the SS equations*
(b) *discretising the equations derived in Exercise 2.3.*

Solution Outline

(a) The equation is obtained from eqn. (5.37) by substituting $\Delta P_{cow} = P'_{cow}\,\Delta S_w$ etc. In practice, P_c is calculated at every time step and the term ΔP_c^n can be evaluated directly.
(b) Starting with equation (D) of Exercise 2.3, discretise it as

$$\Delta[T_o(\Delta p_o^{n+1} - \gamma_o \Delta z)] + (b_o/b_w)^{n+1}\Delta[T_w(\Delta p_o^{n+1} - \gamma_w \Delta z)] \quad \text{(A)}$$
$$+ (b_o/b_g)^{n+1}\Delta[T_g(\Delta p_o^{n+1} - \gamma_g \Delta z)]$$
$$= \frac{V}{\Delta t}\{S_o^n[(\phi b_o)' + (b_o^2/b_g)^{n+1}\phi^{n+1}R_s'] + S_w^n(b_o/b_w)^{n+1}(\phi b_w)' + S_g^n(b_o/b_g)^{n+1}(\phi b_g)'\}\Delta_t p_o$$
$$+ \text{explicit } P_c \text{ terms}$$

After solving (A), substitute in the water equation to get $\Delta_t S_w$ and then in the oil equation to get $\Delta_t S_g$.

Exercise 5.6

Derive the stability limit of IMPES with respect to P_c in three-phase flow (*Coats,* 1968).

Solution Outline

Neglecting solution gas, the equivalent of eqn. (5.43a) and (5.43b) is

$$T_w \Delta^2 e_2^{n+1} = \frac{V_p}{\Delta t}\Delta_t e_1 + P'_{cow} T_w \Delta^2 e_1^n \quad \text{(A)}$$

$$T_o \Delta^2 e_2^{n+1} = -\frac{V_p}{\Delta t}[\Delta_t e_1 + \Delta_t e_3]$$

$$T_g \Delta^2 e_2^{n+1} = \frac{V_p}{\Delta t}\Delta_t e_3 - P'_{cog} T_g \Delta^2 e_3^n$$

where e_1, e_2, e_3 are errors in S_w, p_o and S_g, respectively. Substitute the formal solution for e_l which satisfies

$$T_l \Delta^2 e_m = -T_l \gamma_m e_m \qquad l = o, w, g \qquad m = 1, 2, 3$$

where

$$\gamma_m = 4\sin^2\frac{\alpha_m}{2}$$

and eliminate $\Delta_t e_l$ terms to get two equations:

$$\frac{V_p}{\Delta t} e_1^{n+1} = \left[\frac{V_p}{\Delta t} + \frac{P'_{cow} T_w}{T_T}(T_o + T_g)\gamma_1\right] e_1^n + \frac{P'_{cog} T_g}{T_T}\gamma_3 e_3^n \tag{B}$$

$$\frac{V_p}{\Delta t}\left(e_1^{n+1} + e_3^{n+1}\right) = \left(\frac{V_p}{\Delta t} + \frac{P'_{cow} T_w}{T_T}\gamma_3\right) e_1^n + \left(\frac{V_p}{\Delta t} - \frac{P'_{cog} T_g}{T_T}\gamma_3\right) e_3^n$$

Expressing this in matrix form as $\boldsymbol{e}^{n+1} = \boldsymbol{B}\boldsymbol{e}^{\mathbf{n}}$ gives

$$\mathbf{B} = \frac{\Delta t}{V_p}\begin{bmatrix} \frac{V_p}{\Delta t} + \gamma_1 P'_{cow} T_w\left(1 - \frac{T_w}{T_T}\right) & \gamma_3 P'_{cog}\frac{T_w T_g}{T_T} \\ -\gamma_1 P'_{cow}\frac{T_w T_g}{T_T} & \frac{V_p}{\Delta t} - \gamma_3 P'_{cog}\frac{T_g}{T_T}(T_o + T_w) \end{bmatrix} \tag{C}$$

The eigenvalues μ_i of $\mathbf{B} = \{b_{ij}\}$ are

$$\mu_{1,2} = \tfrac{1}{2}[b_{11} + b_{22} \pm \sqrt{(b_{11} + b_{22})^2 - 4(b_{11}b_{22} - b_{12}b_{21})}]$$

The maximum value of $|\mu|$ is obtained from the root

$$\mu = 1 - \frac{\Delta t}{2V_p T_T}[\gamma_3 P'_{cog} T_g(T_o + T_w) - \gamma_1 P'_{cow} T_w(T_o + T_g) + \sqrt{X}]$$

where

$$X = [\gamma_1 P'_{cow} T_w(T_o + T_g) + \gamma_3 P'_{cog} T_g(T_o + T_w)]^2 - 4\gamma_1\gamma_3 P'_{cow} P'_{cog} T_w^2 T_g^2$$

The restriction $|\mu| < 1$ gives

$$\Delta t \le \frac{4V_p T_T}{\gamma_3 P'_{cog} T_g(T_o + T_w) - \gamma_1 P'_{cow} T_w(T_o + T_g) + \sqrt{X}} \tag{D}$$

Exercise 5.7

Show that the SS method is unconditionally stable with respect to the primary variables. For simplicity, assume incompressible, two-phase flow.

Solution Outline

Using the linearised stability analysis, the equations are locally linearised by assuming that $\Delta(T\,\Delta p) = T\,\Delta^2 p$, etc. Ignoring gravity forces,

$$\mathrm{T_w}\,\Delta^2 p_{\mathrm{w}}^{n+1} = \frac{V_p}{\Delta t} S'_{\mathrm{w}}(\Delta_t p_n - \Delta_t p_{\mathrm{w}}) + q_{\mathrm{w}} \tag{A}$$

$$\mathrm{T_n}\,\Delta^2 p_{\mathrm{n}}^{n+1} = -\frac{\mathrm{V}_p}{\Delta t} S'_{\mathrm{w}}(\Delta_t p_{\mathrm{n}} - \Delta_t p_{\mathrm{w}}) + q_{\mathrm{n}}$$

The errors satisfy

$$\begin{aligned} \mathrm{T_w}\,\Delta^2 e_1^{n+1} &= C(\Delta_t e_2 - \Delta_t e_1) \\ \mathrm{T_n}\,\Delta^2 e_2^{n+1} &= -C(\Delta_t e_2 - \Delta_t e_1) \end{aligned} \tag{B}$$

where $C = -V_p S'_{\mathrm{w}}/\Delta t > 0$. Since $\Delta^2 e_l = -\gamma_l e_l$

$$\boldsymbol{e}^{n+1} = \mathbf{B}\boldsymbol{e}^n$$

where

$$\boldsymbol{e} = [e_1, e_2]^{\mathrm{T}} \tag{C}$$

$$B = \frac{C}{C(\mathrm{T_w}\gamma_1 + \mathrm{T_n}\gamma_2) + \mathrm{T_w T_n}\gamma_1\gamma_2}\begin{bmatrix} \mathrm{T_n}\gamma_2 & -\mathrm{T_n}\gamma_2 \\ -\mathrm{T_w}\gamma_1 & \mathrm{T_w}\gamma_1 \end{bmatrix}$$

The maximum eigenvalue of **B** is

$$\mu = \frac{C(\mathrm{T_n}\gamma_2 + \mathrm{T_w}\gamma_1)}{C(\mathrm{T_n}\gamma_2 + \mathrm{T_w}\gamma_1) + \mathrm{T_w T_n}\gamma_1\gamma_2} < 1 \tag{D}$$

Exercise 5.8

Show that the difference equation (*eqn.* (5.58)) *is equivalent to eqn.* (5.57).

Solution Outline

Starting with eqn. (5.57), for a grid point without a source

$$-(\mathrm{T_w} + \mathrm{T_n})_{i+1/2}(p_{i+1} - p_i) = u_{\mathrm{T}\,i+1/2} = u_{\mathrm{T}\,i-1/2} = u_{\mathrm{T}} \tag{A}$$

Using this, the wetting phase equation is

$$-\left(\frac{\mathrm{T_w}}{\mathrm{T_w} + \mathrm{T_n}}\right)_{i+1/2} u_{\mathrm{T}} + \left(\frac{\mathrm{T_w}}{\mathrm{T_w} + \mathrm{T_n}}\right)_{i-1/2} u_{\mathrm{T}} = \frac{V_{p_i}}{\Delta t}\Delta_t S_{\mathrm{w}_i} \tag{B}$$

Using upstream weighting on T_l, this is

$$-u_{\mathrm{T}}(f_{\mathrm{w}_i} - f_{\mathrm{w}_{i-1}}) = \frac{V_{p_i}}{\Delta t}\Delta_t S_{\mathrm{w}_i} \tag{C}$$

which is equivalent to eqn. (5.58).

Exercise 5.9

Derive a set of sufficient conditions for the existence of a unique solution for the three-phase SS equations. (Hint*: Find the condition under which all rows of* (– **T** + **D**) *are diagonally dominant; use the formulation of Exercise* 5.2).

Solution Outline

Denote

$$\begin{aligned} RT_{i+1/2} &= \tfrac{1}{2}(R_{s_{i+1}} - R_{s_i})T_{o_{i+1/2}} \\ a &\equiv \sum RT = -RT_{i+1/2} + RT_{i-1/2} \\ b &= |-RT_{i+1/2}| + |RT_{i-1/2}| \geq |a| \end{aligned} \tag{A}$$

The diagonal blocks of – **T** and **D** are

$$-\mathbf{T} = \begin{bmatrix} \sum T_w & 0 & 0 \\ 0 & \sum T_o & 0 \\ 0 & \sum RT & \sum T_g \end{bmatrix}$$

$$\mathbf{D} = \frac{V_p}{\Delta t}\left\{ \begin{bmatrix} -b_w^{n+1}S'_w & b_w^{n+1}S'_w & 0 \\ b_o^{n+1}S'_w & -b_o^{n+1}(S'_w - S'_g) & -b_o^{n+1}S'_g \\ 0 & -b_g^{n+1}S'_g & b_g^{n+1}S'_g \end{bmatrix} + \begin{bmatrix} S_w^n b'_w & 0 & 0 \\ 0 & S_o^n b'_o & 0 \\ 0 & R'_s(S_o b_o)^n & S_g^n b'_g \end{bmatrix} \right\}$$

Assuming $S'_g > 0$, $S'_w < 0, R'_s > 0$, only the element in the third row and second column needs to be investigated. This gives

$$\frac{V_p}{\Delta t}(b_g S'_g + S_g b'_g) \geq \left|\frac{V_p}{\Delta t}(S_o b_o R'_s - b_g S'_g) + a\right| + b = |A| + b \tag{B}$$

Since $S_g b'_g \geq 0$, this term may be ignored on the left side. Consider first the case when $A > 0$. Then

$$b_g S'_g < S_o b_o R'_s + a\frac{\Delta t}{V_p} \tag{C}$$

Substitution of (C) in (B) leads to

$$\Delta t(b - a) < V_p S_o b_o R'_s \tag{D}$$

The case of $A < 0$ leads to the same condition. Note that (D) is sufficient, but not necessary for the existence of solution.

CHAPTER 6

SOLUTION OF BLOCK TRIDIAGONAL EQUATIONS

6.1 INTRODUCTION

In this chapter we will consider the solution of matrix equations resulting from the discretisation of one-dimensional, two- and three-phase flow equations. The particular problem of interest here is the one for which simultaneous solution is required for more than one unknown at each grid point. The simultaneous solution approach for two-phase flow results in two unknowns per grid point and for three-phase flow three unknowns per grid point. The most general three-phase flow equations may be written as:

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d} \tag{6.1}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{b}_1 & & & \\ \mathbf{c}_2 & \mathbf{a}_2 & \mathbf{b}_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \mathbf{c}_N & \mathbf{a}_N \end{bmatrix} \tag{6.2}$$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{u}_N \end{bmatrix} \quad \boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{d}_N \end{bmatrix} \tag{6.3a, b}$$

$$\boldsymbol{u}_i = \begin{bmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \end{bmatrix} \quad \boldsymbol{d}_i = \begin{bmatrix} d_i^1 \\ d_i^2 \\ d_i^3 \end{bmatrix} \tag{6.4a, b}$$

$$\mathbf{a}_i = \begin{bmatrix} a_i^{1,1} & a_i^{1,2} & a_i^{1,3} \\ a_i^{2,1} & a_i^{2,2} & a_i^{2,3} \\ a_i^{3,1} & a_i^{3,2} & a_i^{3,3} \end{bmatrix} \tag{6.5}$$

and $\mathbf{c}_i$ and $\mathbf{b}_i$ are defined in a similar fashion.

We note that eqn. (6.1) may be obtained from eqn. (4.2) by replacing scalars quantities with appropriate matrices or vectors. The equations for a general grid point *i* take the form

$$\mathbf{c}_i \boldsymbol{u}_{i-1} + \mathbf{a}_i \boldsymbol{u}_i + \mathbf{b}_i \boldsymbol{u}_{i+1} = \boldsymbol{d}_i \tag{6.6}$$

For cyclic or periodic boundary conditions we obtain a matrix of the form of eqn. (4.6) with scalar elements replaced by 3 × 3 matrices. This analogy between single-phase and multiphase problems shows that at least some of the methods presented in Chapter 4 may be extended for the solution of the problem being considered here. In many practical problems matrices $\mathbf{c}_i$, $\mathbf{a}_i$, and $\mathbf{b}_i$ are not full and considerable computer time can be saved by taking advantage of the zeros in these matrices.

6.2 METHODS OF SOLUTION

A detailed discussion (with computational algorithms) of all the available methods will not serve our objectives. Instead we will present a general discussion of several methods and then present some key programs in Appendix B.

6.2.1 Extension of Thomas' Algorithm

This is an extension of the Thomas algorithm for eqn. (6.1). The equation is of the form referred to as quasi-tridiagonal by Schechter (1960) and block-tridiagonal by Varga (1962). It is also discussed by Richtmyer and Morton (1967). We write

$$\mathbf{A} = \mathbf{WQ} \tag{6.7}$$

where **W** and **Q** are defined by

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & & & & \\ \mathbf{c}_2 & \mathbf{w}_2 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \mathbf{c}_{N-1} & \mathbf{w}_{N-1} \\ & & & & \mathbf{c}_N \quad \mathbf{w}_N \end{bmatrix} \tag{6.8}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{q}_1 & & & \\ & \mathbf{I}_2 & \mathbf{q}_2 & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \mathbf{I}_{N-1} \quad \mathbf{q}_{N-1} \\ & & & & \mathbf{I}_N \end{bmatrix} \tag{6.9}$$

The relationships for finding $\mathbf{w_i}$ and $\mathbf{q_i}$ are obtained by equating the block elements of **WQ** to the corresponding elements of **A**. The resulting equations are

$$\mathbf{w}_1 = \mathbf{a}_1, \mathbf{q}_1 = \mathbf{w}_1^{-1}\mathbf{b}_1 \tag{6.10}$$

$$\left.\begin{aligned} \mathbf{q}_{i-1} &= \mathbf{w}_{i-1}^{-1}\mathbf{b}_{i-1} \\ \mathbf{w}_i &= \mathbf{a}_i - \mathbf{c}_i\mathbf{q}_{i-1} \end{aligned}\right\} \quad i = 2, \ldots, N \tag{6.11) (6.12}$$

We note that $\mathbf{w_i}$ must be non-singular and their inverse is required.

Once $\mathbf{w_i}$ and $\mathbf{q_i}$ are known, the solution may be obtained by first solving the forward elimination

$$\mathbf{W}g = d \tag{6.13}$$

This is accomplished by

$$\begin{aligned} g_1 &= \mathbf{w}_1^{-1}d_1 \\ g_i &= \mathbf{w}_i^{-1}(d_i - \mathbf{c}_i g_{i-1}), \qquad i = 2, \ldots, N \end{aligned} \tag{6.14}$$

The solution vectors u_i may be computed from

$$\mathbf{Q}u = g \tag{6.15}$$

by backward elimination. This is accomplished with

$$\begin{aligned} \boldsymbol{u}_N &= \boldsymbol{g}_N \\ \boldsymbol{u}_i &= \boldsymbol{g}_i - \mathbf{q}_i \boldsymbol{u}_{i+1}, \qquad i = N-1, \ldots, 1 \end{aligned} \tag{6.16}$$

Note again that in this algorithm w_i^{-1} are required and these matrices must not be singular. A slightly different factorisation of **A** is presented by Schechter (1960). Many versions of this algorithm are possible. Douglas *et al.* (1959) have presented an algorithm for two-phase problems (bitridiagonal) and Von Rosenberg (1969) has presented algorithms for two- and three-phase (tridiagonal) problems. The main difference of the algorithms by Douglas *et al.* and Von Rosenberg compared to this algorithm is that the matrix operations are expanded in terms of matrix elements. Consequently, it is easier to take advantage of systematically appearing zeros in this formulation. In the process of factorisation we have chosen the diagonal elements of **Q** to be the identity matrices **I**. This is only one of many possibilities open to us. For example, for the case of 2 × 2 sub-matrices, Weinstein *et al.* (1970) show the preferred form in which the identity matrices are replaced by matrices of the form

$$\begin{bmatrix} 1 & \varepsilon_i \\ 0 & 1 \end{bmatrix}$$

This choice leads to $\mathbf{w}_i$ matrices which are lower triangular. This form is claimed to have lower round-off error.

Fully tested programs for bi-tridiagonal and tri-tridiagonal problems are presented in Appendix B.

6.2.2 Use of Methods for Band Matrices

The matrix **A** is a band matrix and any of the methods developed for band matrices may be used. However, these methods will not be as efficient for eqn. (6.1) as the methods discussed in the previous section, because band algorithm treats all zeros within the band as non-zeros.

We will present a full discussion of band matrices in Chapter 8.

CHAPTER 7

FLOW OF A SINGLE FLUID IN TWO DIMENSIONS

7.1 INTRODUCTION

In this chapter we will extend the ideas presented in Chapter 3 to the solution of various reservoir simulation problems involving the flow of a single fluid in two space dimensions. Some new problems and methods that are only important in a two-dimensional (2-D) setting will be introduced here. All of these ideas will be extended to the solution of multiphase problems in Chapter 9 and to three-dimensional problems in Chapter 11.

Although we have already introduced many techniques for handling practical reservoir simulation problems, so far these techniques have been applied only to highly idealised situations in one space dimension. This is the first chapter where we can start discussing details of some practical models for gas reservoirs, and oil reservoirs above the bubble point. Numerical techniques that are closely related to the partial differential equations for flow in two dimensions will also be presented here. Some of the techniques that can be applied to the matrix equations resulting from the finite-difference approximations of the partial differential equations, will be discussed in the next chapter.

7.2 CLASSIFICATION OF 2-D PROBLEMS

All real reservoirs are, of course, three-dimensional; it is, however, possible in many practical situations to assume that flow in one of the three co-ordinate directions is negligible compared to flow in the other two directions. There are three classes of problems that can be handled in this manner. These problems and associated mathematical models are described below.

7.2.1 Areal Problems (x–y)

In thin reservoirs of large areal extent it is often possible to assume that the pressure gradients and hence flow in the z-direction is negligible compared to flow in the other two directions. Small changes in vertical thickness may be accommodated by allowing

the block thickness Δz and elevation h to be functions of x and y. A typical block for the areal model is shown in Fig. 7.1.

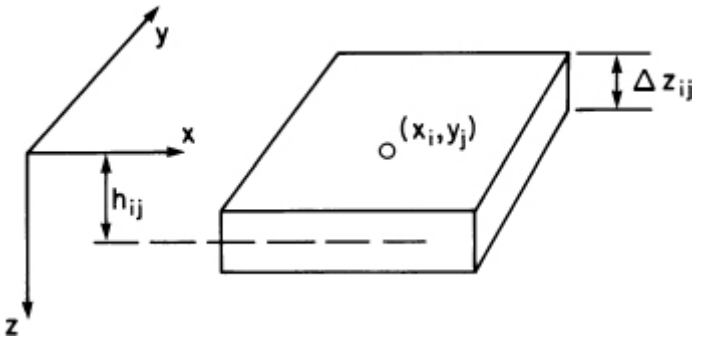


FIG. 7.1. A typical block for the areal model

For a case of this type the general flow equation may be written as (see eqns. 2.35, 2.42, 2.43, 2.44, 2.48, 2.51 and 2.53):

$$\frac{\partial}{\partial x}\left[\Delta z \lambda X\left(\frac{\partial p}{\partial x} - \gamma \frac{\partial h}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\Delta z \lambda Y\left(\frac{\partial p}{\partial y} - \gamma \frac{\partial h}{\partial y}\right)\right] = \Delta z \beta \frac{\partial p}{\partial t} + \Delta z q \tag{7.1}$$

where

$$p(x, y) = \frac{1}{\Delta z}\int_0^{\Delta z} p(x, y, z)\,\mathrm{d}z$$

The coefficients λX, λY and β depend on the partial differential equation being solved. For example, if we consider eqn. (2.42), then,

$$\lambda X = \frac{k_x}{\mu B}$$

$$\lambda Y = \frac{k_y}{\mu B}$$

and

$$\beta = \left(\phi \frac{c_f}{B^\circ} + \phi^\circ \frac{c_R}{B}\right)$$

where the reservoir properties must be defined as averages over the thickness (see the definition of pressure used above). The symbol γ used in the gravitational terms is defined by

$$\rho \frac{g}{g_c} = \gamma$$

and the gravitational terms are identically zero if x and y co-ordinates are in the horizontal plane. Obviously the coefficients λX, λY and β can be functions of x, y, t and p while γ is a function of p. Hence, in general, the problem is nonlinear, even though the nonlinearity in this case is rather weak.

7.2.2 Cross-sectional Problems (x–z)

If instead of neglecting flow in the vertical direction, flow in one of the two horizontal directions is neglected, the resulting model is called cross-sectional. This type of model can be used for cases where the flow is predominantly in the vertical direction and one of two horizontal directions. If we choose to write the equations in x–z co-ordinates then small variations in the y direction may be accounted for by using a variable thickness Δy for each block. A typical block for this type of a model is shown in Fig. 7.2.

A typical equation for cross-sectional problems is obtained by replacing y by z, λY by λZ and Δz by Δy in eqn. (7.1):

$$\frac{\partial}{\partial x}\left[\Delta y \lambda X\left(\frac{\partial p}{\partial x} - \gamma \frac{\partial h}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[\Delta y \lambda Z\left(\frac{\partial p}{\partial z} - \gamma \frac{\partial h}{\partial z}\right)\right] = \Delta y \beta \frac{\partial p}{\partial t} + \Delta y q \tag{7.2}$$

where

$$p(x, z) = \frac{1}{\Delta y}\int_0^{\Delta y} p(x, y, z)\, dy$$

$$\lambda X = \frac{k_x}{\mu B}$$

$$\lambda Z = \frac{k_z}{\mu B}$$

and

$$\beta = \left(\phi \frac{c_f}{B^\circ} + \phi^\circ \frac{c_R}{B} \right)$$

If the x-co-ordinate is truly horizontal and if Δy; is constant, the equation simplifies to

$$\frac{\partial}{\partial x}\left[\lambda X \frac{\partial p}{\partial x}\right] + \frac{\partial}{\partial z}\left[\lambda Z\left(\frac{\partial p}{\partial z} - \gamma\right)\right] = \beta \frac{\partial p}{\partial t} + q \tag{7.3}$$

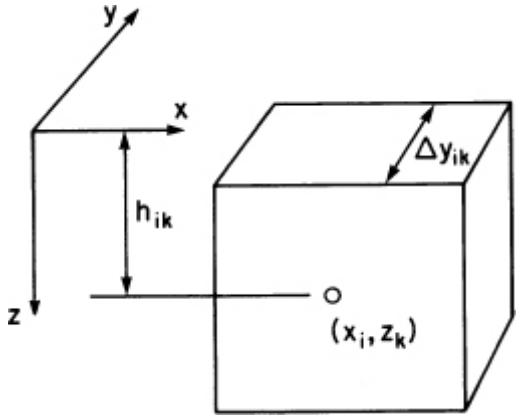


FIG. 7.2. A typical block for the cross-sectional model.

7.2.3 Single-Well Problems (r–z)

One of the important applications of the fluid flow equations to petroleum engineering problems is in the field of well testing. Oil and gas wells are tested to determine their productivity. Most well tests involve the production of a well at a sequence of constant flow rates and the observation of bottom hole pressure during the test period. A comparison of test data with theory allows one to predict the basic reservoir data and hence the productivity of the well. This rather specialised subject is discussed by Matthews and Russell (1967), in a manual published by the Energy Resources Conservation Board of Alberta (ERCB, 1975), and by Earlougher (1977).

Most of the well test theory is based on the assumption of single-phase, one-dimensional radial flow with some additional assumptions which linearise the partial differential equation (see Chapter 2). For cases where the flow is two-dimensional or when the equations cannot be linearised, it is not practical or possible to get analytical solutions. This is where a computer model in r-z co-ordinates can be used to analyse well test data. A typical block in r-z co-ordinates is shown in Fig. 7.3.

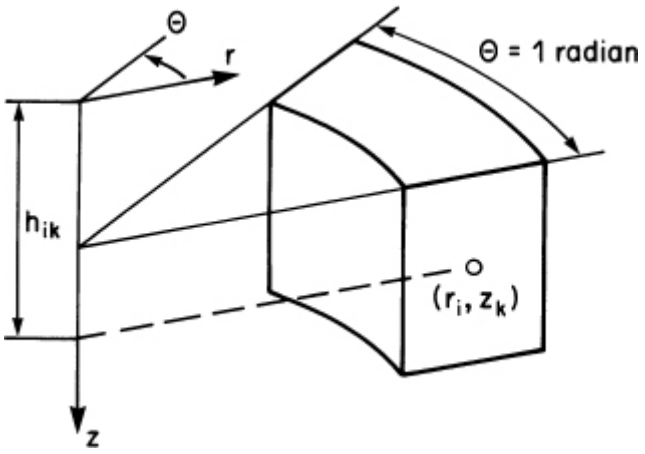


FIG. 7.3. A typical block for a well model in *r-z* co-ordinates.

A typical equation in *r–z* co-ordinates can be readily obtained from eqn. (7.2):

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\lambda R\left(\frac{\partial p}{\partial r}-\gamma\frac{\partial h}{\partial r}\right)\right]+\frac{\partial}{\partial z}\left[\lambda Z\left(\frac{\partial p}{\partial z}-\gamma\frac{\partial h}{\partial z}\right)\right]=\beta\frac{\partial p}{\partial t}+q \tag{7.4}$$

where

$$p(r,z)=\frac{1}{2\pi}\int_0^{2\pi}p(r,z,\theta)\,\mathrm{d}\theta$$

$$\lambda R=\frac{k_r}{\mu B}$$

$$\lambda Z=\frac{k_z}{\mu B}$$

and

$$\beta=\left(\phi\frac{c_\mathrm{f}}{B^\circ}+\phi^\circ\frac{c_\mathrm{R}}{B}\right)$$

(See Section 7.10.2 for a derivation of the co-ordinate transformation.) The flow equation in the form of eqn. (7.4) is written for an angle of one radian, which must be accounted for in the definition of *q*.

7.2.4 Comments on Two-Dimensional Models

A review of the equations presented above reveals that a single model could be developed to handle all of the three cases. Also the form of the equations presented is quite general, and with appropriate definitions of coefficients these equations represent various fluid flow problems discussed in Chapter 2. In the next section we will show how terms in eqns. (7.1) to (7.4) can be discretised.

All 2-D problems are simplifications of the real situation. This fact must be kept in mind while results of simulation runs are being interpreted. The 3-D character is only partially accounted for by varying the thickness in the third dimension. Strictly speaking, two-dimensional flow equations with variable thickness in the third dimension are not correct, but they do provide a good approximation as long as the variation of thickness is not large.

7.3 DISCRETISATION OF FLOW EQUATIONS

In this section we will derive difference approximations that represent various two-dimensional single-phase flow situations. The condition for the stability of the explicit method will also be discussed.

7.3.1 Difference Approximations

Let us first consider the discretisation of eqn. (7.1) in detail over the grid shown in Fig. 7.4. A typical block from Fig. 7.4 is shown in Fig. 7.5. The notation used is a direct extension of that used in Chapter 3 (Section 3.5). The flow term in the x-direction is approximated by an application of the operator given by eqn. (3.117). After multiplying through by $\Delta x_i \Delta y_j$ the finite-difference approximation may be written as

$$\begin{aligned} \Delta x_i \Delta y_j \frac{\partial}{\partial x}\left[\Delta z \lambda X\left(\frac{\partial p}{\partial x} - \gamma \frac{\partial h}{\partial x}\right)\right] &\simeq \Delta_x \mathrm{T}X \Delta_x (p - \gamma h) \\ &\equiv \mathrm{T}X_{(i+1/2),j}[p_{i+1} - p_i - \gamma_{i+1/2}(h_{i+1} - h_i)]_j \\ &\quad + \mathrm{T}X_{(i-1/2),j}[p_{i-1} - p_i - \gamma_{i-1/2}(h_{i-1} - h_i)]_j \end{aligned} \tag{7.5}$$

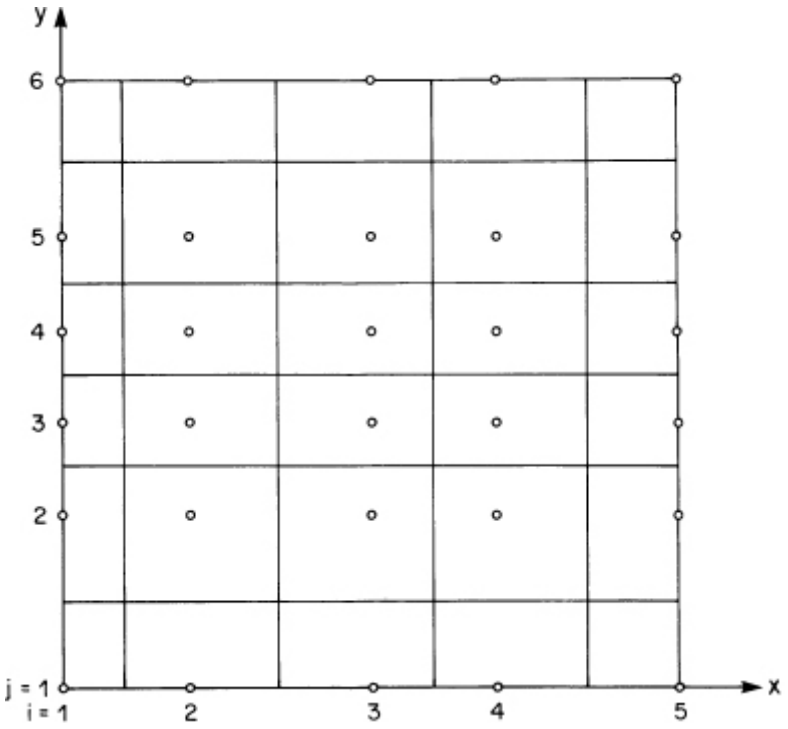


FIG. 7.4. Typical grid for 2-D (x–y) models.

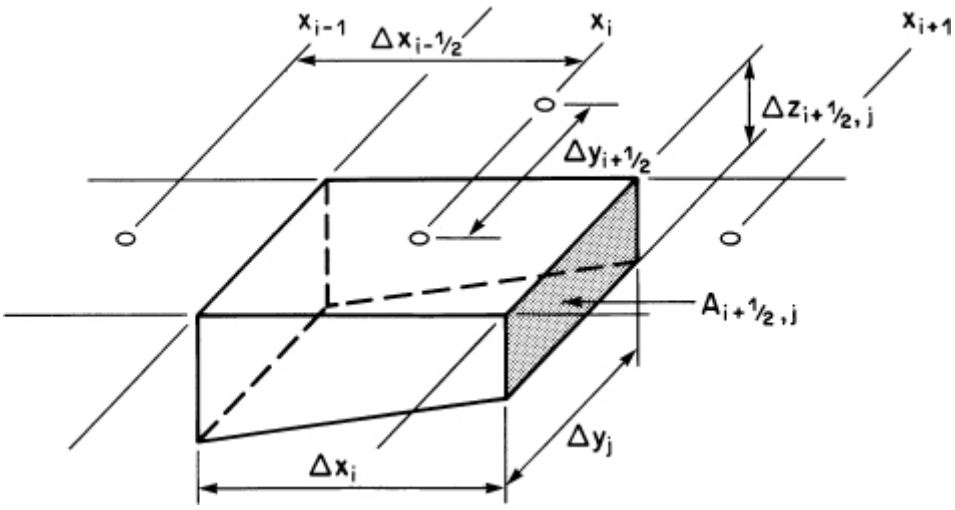


FIG. 7.5. Typical block from the grid of Fig. 7.4.

The transmissibility terms are defined by

$$TX_{(i+1/2),j} = \lambda X_{(i+1/2),j} \frac{\Delta y_j \Delta z_{(i+1/2),j}}{\Delta x_{i+1/2}} \tag{7.6a}$$

$$= \lambda X_{(i+1/2),j} \frac{A_{(i+1/2),j}}{\Delta x_{i+1/2}}$$

$$\mathrm{T}X_{(i-1/2),j} = \lambda X_{(i-1/2),j} \frac{A_{(i-1/2),j}}{\Delta x_{i-1/2}} \tag{7.6b}$$

The $\Delta x_{i+\frac{1}{2}}$ and $\Delta x_{i-\frac{1}{2}}$ terms are defined by

$$\begin{aligned} \Delta x_{i+1/2} &= x_{i+1} - x_i \\ \Delta x_{i-1/2} &= x_i - x_{i-1} \end{aligned} \tag{7.7}$$

and

$$\Delta x_i = \tfrac{1}{2}[\Delta x_{i+1/2} + \Delta x_{i-1/2}] \tag{7.8}$$

Using a similar notation we can write the approximation of y-derivative as

$$\begin{aligned} \Delta x_i \Delta y_j \frac{\partial}{\partial y}\left[\Delta z \lambda Y\left(\frac{\partial p}{\partial y} - \gamma \frac{\partial h}{\partial y}\right)\right] &\simeq \Delta_y \mathrm{T}Y \Delta_y (p - \gamma h) \\ &\equiv \mathrm{T}Y_{i,j+1/2}[p_{j+1} - p_j - \gamma_{j+1/2}(h_{j+1} - h_j)]_i \\ &\quad + \mathrm{T}Y_{i,j-1/2}[p_{j-1} - p_j - \gamma_{j-1/2}(h_{j-1} - h_j)]_i \end{aligned} \tag{7.9}$$

where

$$\begin{aligned} \mathrm{T}Y_{i,j+1/2} &= \lambda Y_{i,j+1/2} \frac{\Delta x_i \Delta z_{i,j+1/2}}{\Delta y_{j+1/2}} \\ &= \lambda Y_{i,j+1/2} \frac{A_{i,j+1/2}}{\Delta y_{j+1/2}} \end{aligned} \tag{7.10}$$

$$\mathrm{T}Y_{i,j-1/2} = \lambda Y_{i,j-1/2} \frac{A_{i,j-1/2}}{\Delta y_{j-1/2}} \tag{7.11}$$

The finite-difference approximation for eqn. (7.1) at point i, j may now be written in a compact form as

$$[\Delta \mathrm{T} \Delta(p - \gamma h)]_{ij} = \frac{V_{ij}\beta_{ij}}{\Delta t} \Delta_t p_{ij} + Q_{ij} \tag{7.12}$$

where

$$[\Delta \mathrm{T} \Delta(p - \gamma h)]_{ij} \equiv [\Delta_x \mathrm{T}X \Delta_x (p - \gamma h) + \Delta_y \mathrm{T}Y \Delta_y (p - \gamma h)]_{ij} \tag{7.13}$$

$$\Delta_t p_{ij} = p_{ij}^{n+1} - p_{ij}^{n} \tag{7.14}$$

$$V_{ij} = \Delta z_{ij} \Delta x_i \Delta y_j \tag{7.15}$$

$$Q_{ij} = q_{ij} V_{ij} \tag{7.16}$$

In writing eqn. (7.12) we have not specified any time levels. For the forward-difference (explicit) method the left side is evaluated at the n level of time and for the backward-difference approximation (implicit) it is evaluated at the $n + 1$ level of time. On the right side of eqn. (7.12) the terms β_{ij} must be evaluated by a scheme which conserves material (see eqns. 3.173 and 3.174). One such approximation is

$$\beta_{ij} = \left(\frac{\phi^n c_f}{B^\circ} + \frac{\phi^\circ c_R}{B^{n+1}} \right)_{ij} \tag{7.17}$$

We note that one of the terms in eqn. (7.17) must be evaluated at the $n + 1$ level even for the explicit method.

For the Crank–Nicolson method the left side of eqn. (7.12) is evaluated at the n + ½ level by the following average:

$$[\Delta T \, \Delta(p - \gamma h)]_{ij}^{n+1/2} = \tfrac{1}{2}[\Delta T \, \Delta(p - \gamma h)]_{ij}^{n} + \tfrac{1}{2}[\Delta T \, \Delta(p - \gamma h)]_{ij}^{n+1} \tag{7.18}$$

It should be clear from the above discussion that the approximations for the cross-sectional model (eqn. 7.2) may be written in exactly the same manner as those for the areal model (7.1).

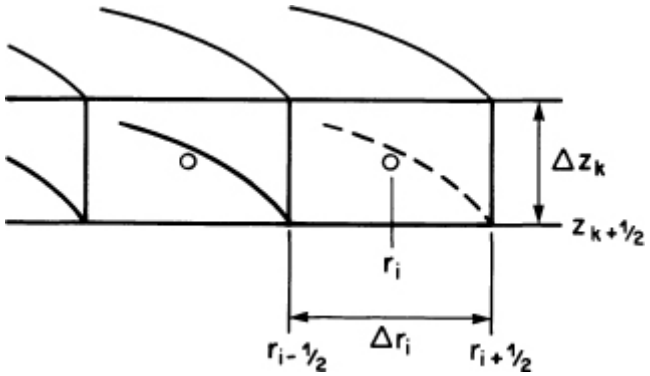


FIG. 7.6. Block boundaries in the radial direction.

As discussed at length in Chapter 3 (Section 3.6), problems involving the cylindrical co-ordinates must be handled slightly differently. The block boundaries for calculating

the block volume are chosen by (see Fig. 7.6):

$$r_{i+1/2} = \left[\frac{r_{i+1}^2 - r_i^2}{\ln\left(\frac{r_{i+1}^2}{r_i^2}\right)}\right]^{1/2} \tag{7.19}$$

$$z_{k+1/2} = \tfrac{1}{2}(z_k + z_{k+1}) \tag{7.20}$$

The finite-difference approximation for eqn. (7.4) may be written in the form of eqn. (7.12) with the following definition for the operator on the left side:

$$[\Delta \mathrm{T}\,\Delta(p - \gamma h)]_{ik} \equiv [\Delta_r \mathrm{T}R\,\Delta_r(p - \gamma h) + \Delta_z \mathrm{T}Z\,\Delta_z(p - \gamma h)]_{ik} \tag{7.21}$$

where

$$\mathrm{T}R_{(i+1/2),k} = 4\pi\left(\frac{r_{i+1/2}^2\,\Delta z_k}{r_{i+1}^2 - r_i^2}\right)\lambda R_{(i+1/2),k} \tag{7.22}$$

$$\mathrm{T}Z_{i,k+1/2} = \pi\left(\frac{r_{i+1/2}^2 - r_{i-1/2}^2}{\Delta z_{k+1/2}}\right)\lambda Z_{i,k+1/2} \tag{7.23}$$

Other transmissibilities ($\mathrm{T}R_{(i-\frac{1}{2}),\mathrm{k}}$ and $\mathrm{T}Z_{i,k-\frac{1}{2}}$) are defined in a similar fashion. The volume of block (i, k) is given by

$$V_{ik} = \pi\,\Delta z_k(r_{i+1/2}^2 - r_{i-1/2}^2) \tag{7.24}$$

and Q_{ik} is the production for $\theta = 2\pi$.

We see that with appropriate definitions of transmissibilities, block boundaries, and block volumes the finite-difference approximations are of identical form for the areal, cross-sectional and radial models. For this reason most of the discussion to follow will be related to the areal model. We will, however, point out important differences whenever appropriate.

7.3.2 Stability of Difference Schemes

The explicit approximation for the areal model may be obtained by expanding eqn. (7.12) and collecting terms:

$$\mathrm{T}X_{(i-1/2),j}p^n_{i-1,j} + \mathrm{T}Y_{i,j-1/2}p^n_{i,j-1/2} \tag{7.25}$$
$$-\left(\mathrm{T}X_{(i-1/2),j} + \mathrm{T}Y_{i,j-1/2} + \mathrm{T}X_{(i+1/2),j} + \mathrm{T}Y_{i,j+1/2} - \frac{V_{ij}\beta_{ij}}{\Delta t}\right)p^n_{ij}$$
$$+\mathrm{T}X_{(i+1/2),j}p^n_{(i+1/2),j} + \mathrm{T}Y_{i,j+1/2}p_{i,j+1/2} - d_{ij} = \frac{V_{ij}\beta_{ij}}{\Delta t}p^{n+1}_{ij}$$

where d_{ij} contains all sources and gravity terms. In the above equation p^{n+1}_{ij} on the right side is the only unknown provided β_{ij} can be evaluated at the n level of time. Handling of the nonlinearity introduced by β has been discussed in Chapter 3 (Section 3.7.2.1). So we see that there is nothing difficult about computing p^{n+1}_{ij} from the values of p^n_{ij} for all values of i and j.

However, as expected, the explicit method is only conditionally stable. The stability condition may be readily derived by using the concept of *positive type* difference equations (see Chapter 3, Section 3.7.2.1). The resulting condition corresponding to the condition (3.179) for one-dimensional problems is:

$$\max_{i,j,n}\left[\frac{\Delta t}{V_{ij}\beta_{ij}}(\mathrm{T}X_{(i-1/2),j} + \mathrm{T}Y_{i,j-1/2} + \mathrm{T}X_{(i+1/2),j} + \mathrm{T}Y_{i,j+1/2})\right] \leq 1 \tag{7.26}$$

The above condition is more restrictive than the corresponding condition for 1-D problems. As shown in Chapter 3, the stability limit (7.26) makes the explicit method impractical for most cases. A similar condition can be written for the radial and cross-sectional models.

The implicit and Crank–Nicolson methods are, of course, unconditionally stable for the linearised version of the flow equation.

7.4 BOUNDARY CONDITIONS

The initial state of the reservoir (initial conditions) and the interaction of the reservoir with its surrounding must be known before it can be modelled. In many important situations a detailed knowledge of these conditions is lacking and good engineering judgement is required to obtain reasonable estimates.

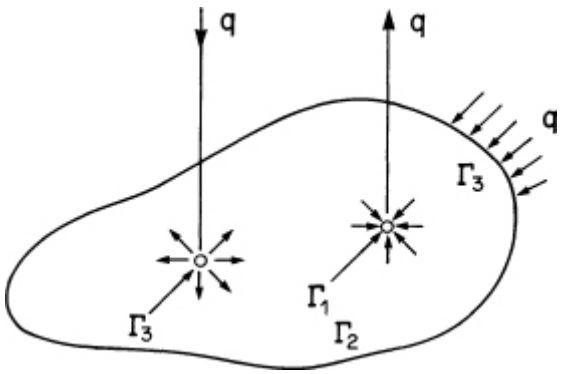


FIG. 7.7. Boundaries for areal systems.

The boundary conditions are required at wells and at the exterior boundary of the reservoir. These boundary conditions may be of one of three types: (1) flow rate specified at the boundary, (2) no flow at the boundary, and (3) pressure specified at the boundary. Except for the study of single-well systems in cylindrical co-ordinates, the wells must be treated as Dirac line or point sources. This approach is necessary because the well radius is usually small compared to the size of a block in areal and cross-sectional studies.

The specification of pressure at a boundary can be handled in a straightforward fashion as discussed in Chapter 3. In this section we will discuss the boundaries across which either zero or some finite flow rate is specified. We will also show that the 'flow' and 'no-flow' boundary conditions may be treated in the same fashion by replacing the actual boundary conditions by homogeneous Neumann (no-flow) boundary conditions and by introducing the flow into or out of the system through source or sink terms. Figures 7.7 and 7.8 show typical boundary conditions for areal and single-well models. We will now consider the 'flow' and 'no-flow' boundaries in detail.

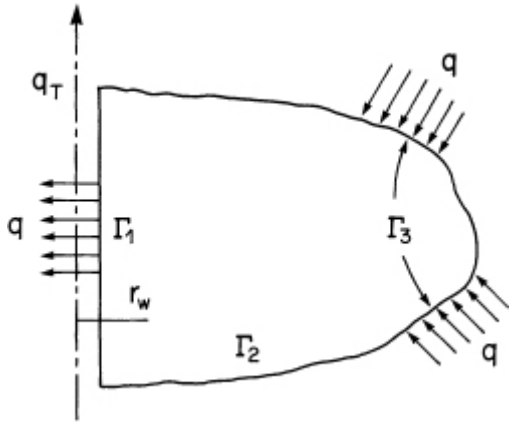


FIG. 7.8. Boundaries for *r–z* systems.

7.4.1 No-flow or Closed Boundaries

When there is no flow across a boundary (Γ_2 in Figs. 7.7 and 7.8) the component of the velocity vector normal to the boundary surface must be zero. The appropriate velocity component is obtained by taking the dot (vector) product of Darcy velocity with the normal vector $\boldsymbol{n}$:

$$\frac{k}{\mu}(\nabla p - \gamma \nabla h) \,.\, \boldsymbol{n} = 0 \qquad \text{on} \quad \Gamma_2 \tag{7.27}$$

In reservoir simulation the boundaries are usually approximated by block boundaries which are parallel to one of the co-ordinate directions. Hence in areal (x–y) models

$$\frac{k_x}{\mu}\left(\frac{\partial p}{\partial x} - \gamma \frac{\partial h}{\partial x}\right) = 0 \tag{7.28}$$

for all boundaries which are normal to the x-direction and

$$\frac{k_y}{\mu}\left(\frac{\partial p}{\partial y} - \gamma \frac{\partial h}{\partial y}\right) = 0 \tag{7.29}$$

for all boundaries which are normal to the y-direction.

7.4.2 Flow Boundaries

When some flow rate across a boundary is specified (Γ_1 and Γ_3 in Figs. 7.7 and 7.8) the normal component of the velocity vector at the boundary must equal this flow rate:

$$\frac{k}{\mu}(\nabla p - \gamma \nabla h) \,.\, \boldsymbol{n} = q(\Gamma) \tag{7.30}$$

and the total flow rate across a boundary is the integrated value of q in eqn. (7.30) over the boundary. For example, over the boundary Γ_3 of Fig. 7.8, the total flow rate is

$$q_{\mathrm{T}} = \int_{\Gamma_3} q(\Gamma)\, \mathrm{d}\Gamma \tag{7.31}$$

If the actual boundary spans more than one grid blocks and if the total flow rate q_{T} is specified, then this flow rate must be distributed, in an appropriate fashion, over the

boundary blocks. This problem will be discussed in the next section where we discuss the finite-difference or discretised form of the boundary conditions.

7.4.3 Discretisation of Boundary Conditions

As an example, let us consider a point (i, k) on the vertical boundary of a cross-sectional grid as shown in Fig. 7.9. If the flow across the boundary is zero then $\partial p/\partial x = 0$ may be approximated by using the second order expression

$$\frac{p_{i+1} - p_{i-1}}{2(x_{i+1} - x_i)} = 0 \tag{7.32}$$

In this approach complete symmetry of properties is assumed about x_i and the method is known as the 'reflection technique'. It preserves the symmetry and the order of approximation of the finite-difference equations.

Flow into or out of the system can be accounted for by two different approaches. One is to assume that the block boundary (1–4) is insulated and there is a source/sink term of appropriate strength in the differential equation. Another approach is to assume the source/sink term to be zero and force the pressure gradient at the block boundary, $\partial p/\partial x|_{xi}$, to take on a value that causes the correct amount of fluid to flow out of this boundary. These two seemingly different approaches turn out to be identical at the finite-difference level as shown below.

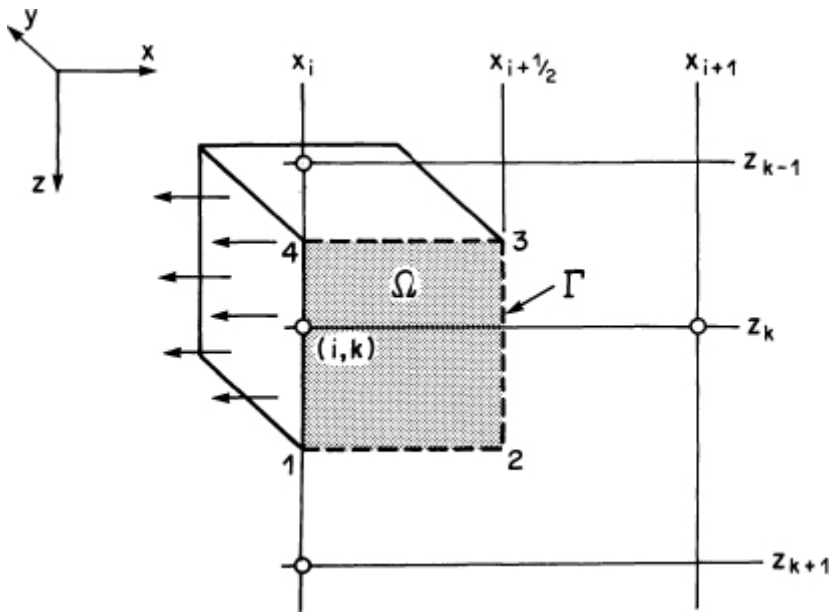


FIG. 7.9. Typical boundary grid block in a cross-section model.

Let us consider the following simplified form of eqn. (2.35) or eqn. (7.3):

$$\frac{\partial}{\partial x}\left[\lambda X\frac{\partial p}{\partial x}\right]+\frac{\partial}{\partial z}\left[\lambda Z\frac{\partial p}{\partial z}\right]=\frac{\partial}{\partial t}\left(\frac{\phi}{B}\right)+q \tag{7.33}$$

We will now apply the integral method of obtaining difference approximations discussed briefly in Chapter 3 (Section 3.2.1.2) to the above equation.

Integration over the volume of block (i, k) yields

$$\Delta y\iint_{\Omega}\left[\frac{\partial}{\partial x}\left(\lambda X\frac{\partial p}{\partial x}\right)+\frac{\partial}{\partial z}\left(\lambda Z\frac{\partial p}{\partial z}\right)\right]\mathrm{d}x\,\mathrm{d}z \tag{7.34}$$
$$=\Delta y\iint_{\Omega}\frac{\partial}{\partial t}\left(\frac{\phi}{B}\right)\mathrm{d}x\,\mathrm{d}z+\Delta y\iint_{\Omega}q(x,z)\,\mathrm{d}x\,\mathrm{d}z$$

The left side of eqn. (7.34) may be converted to a line integral using Green's theorem (e.g., Morse and Feshbach, 1953, pp. 34 and 803):

$$\Delta y\int_{\Gamma}\left(\lambda Z\frac{\partial p}{\partial z}\mathrm{d}x-\lambda X\frac{\partial p}{\partial x}\mathrm{d}z\right)=\Delta y\iint_{\Omega}\frac{\partial}{\partial t}\left(\frac{\phi}{B}\right)\mathrm{d}x\,\mathrm{d}z+\Delta y\iint_{\Omega}q\,\mathrm{d}x\,\mathrm{d}z \tag{7.35}$$

The line integral over Γ can be divided into four parts consisting of the four sides of the cross-section of the block in the x, z plane. The derivatives $\partial p/\partial z$ and $\partial p/\partial x$ may be assumed constant along each of these parts and they may be approximated by finite differences. The following approximations are obtained if we follow this approach:

$$\int_{1}^{2}\lambda Z\frac{\partial p}{\partial z}\mathrm{d}x=\lambda Z_{k+1/2}\left(\frac{x_{i+1/2}-x_i}{z_{k+1}-z_k}\right)(p_{k+1}-p_k) \tag{7.36}$$

$$-\int_{2}^{3}\lambda X\frac{\partial p}{\partial x}\mathrm{d}z=\lambda X_{i+1/2}\left(\frac{z_{k+1/2}-z_{k-1/2}}{x_{i+1}-x_i}\right)(p_{i+1}-p_i) \tag{7.37}$$

$$\int_{3}^{4}\lambda Z\frac{\partial p}{\partial z}\mathrm{d}x=\lambda Z_{k-1/2}\left(\frac{x_{i+1/2}-x_i}{z_k-z_{k-1}}\right)(p_{k-1}-p_k) \tag{7.38}$$

$$\left.\int_4^1 \lambda X \frac{\partial p}{\partial x}\,\mathrm{d}z\right\} \begin{cases} = 0 & \text{if} \quad \dfrac{\partial p}{\partial x} = 0 \\ = \displaystyle\int_4^1 q_x(z)\,\mathrm{d}z, & \text{if} \quad \dfrac{\partial p}{\partial x} \neq 0 \end{cases} \tag{7.39a) (7.39b}$$

where $\Delta y q_x(z)$ is the flow rate per unit length of the boundary (1–4) of block *i, k*:

$$q_x(z) = -\lambda X \left.\frac{\partial p}{\partial x}\right|_{x_i} \tag{7.40}$$

If the flow rate is accounted for by the specification of eqn. (7.39b) then the last term in eqn. (7.35) must be zero and the total flow rate across the boundary from eqn. (7.39) may be written as

$$\bar{Q}_{ik} = \Delta y \int_4^1 q_x(z)\,\mathrm{d}z \tag{7.41}$$

however, if eqn. (7.39a) is used then the total flow rate from the block is

$$Q_{ik} = \Delta y \iint_\Omega q(x, z)\,\mathrm{d}x\,\mathrm{d}z \tag{7.42}$$

The above two equations provide alternative interpretations of the singular 'source' boundary conditions. $\bar{Q}_{ik}$ and Q_{ik} must be equal, since they represent the same lumped production from block (i,k). At the finite-difference level where $q(x, z)$ and $q_x(z)$ must be assumed constant within a block in eqns. (7.41) and (7.42), there is no difference between the two interpretations of the boundary conditions. This provides the required justification for the use of eqn. (7.39a) in all reservoir simulation work. The first term on the right side of eqn. (7.34) may be approximated by

$$\Delta y \iint_\Omega \frac{\partial}{\partial t}\left(\frac{\phi}{B}\right)\mathrm{d}x\,\mathrm{d}z \simeq (x_{i+1/2} - x_i)(z_{k+1/2} - z_{k-1/2})\,\Delta y \frac{\partial}{\partial t}\left(\frac{\phi}{B}\right) \tag{7.43}$$

which is identical to the corresponding term from eqn. (7.33) multiplied by the correct block volume.

Let us now write the standard finite-difference approximation for eqn. (7.33) using the notation used in Section 7.3.

$$[\Delta \mathrm{T}\,\Delta p]_{ik} = \frac{V_{ik}}{\Delta t}\beta_{ik}\,\Delta_t p_{ik} + Q_{ik} \tag{7.44}$$

where

$$\begin{aligned}[\Delta \mathrm{T}\,\Delta p]_{ik} &= \Delta_x \mathrm{T}X\,\Delta_x p + \Delta_z \mathrm{T}Z\,\Delta_z p \\ &= \mathrm{T}X_{(i+1/2),k}[p_{i+1} - p_i]_k + \mathrm{T}X_{(i-1/2),k}[p_{i-1} - p_i]_k \\ &\quad + \mathrm{T}Z_{i,k+1/2}[p_{k+1} - p_k]_i + \mathrm{T}Z_{i,k-1/2}[p_{k-1} - p_k]_i\end{aligned} \tag{7.45}$$

$$\mathrm{T}X_{(i+1/2),k} = \lambda X_{(i+1/2),k}\frac{\Delta y(z_{k+1/2} - z_{k-1/2})}{x_{i+1} - x_i} \tag{7.46}$$

$$\mathrm{T}X_{(i-1/2),k} = \lambda X_{(i-1/2),k}\frac{\Delta y(z_{k+1/2} - z_{k-1/2})}{x_i - x_{i-1}} \tag{7.47}$$

$$\mathrm{T}Z_{i,k+1/2} = \lambda Z_{i,k+1/2}\frac{\Delta y(x_{i+1/2} - x_{i-1/2})}{(z_{k+1} - z_k)} \tag{7.48}$$

$$\mathrm{T}Z_{i,k-1/2} = \lambda Z_{i,k-1/2}\frac{\Delta y(x_{i+1/2} - x_{i-1/2})}{(z_k - z_{k-1})} \tag{7.49}$$

The above definitions apply to an arbitrary full block inside the reservoir. By comparison with eqn. (7.36) to (7.39) we see that the general form (7.44) could also be used for the boundary block (i, k) provided we set

$$\mathrm{T}X_{i-1/2} = 0 \qquad \text{and} \qquad x_{i-1/2} = x_i \tag{7.50}$$

Modifications of the general expression (7.44) for other boundaries are easily obtained by following the above procedure.

The only unresolved problem with respect to boundary conditions is the problem of assigning flow rates Q_{ik} to each block when the total flow rate Q_T from several blocks is specified. This problem is encountered in cross-sectional and single-well models with the well penetrating several blocks. The simplest solution is to allocate the flow rate according to the transmissibility of the block:

$$Q_k = \frac{\mathrm{T}_k}{\sum \mathrm{T}_k} Q_\mathrm{T} \tag{7.51}$$

where T_k is a suitably defined transmissibility. This problem becomes more complex for multiphase flow and will be considered again in Chapter 9 in greater detail.

7.5 INITIAL CONDITIONS

For virgin reservoirs it is reasonable to assume that the pressure gradient in the reservoir is due to the hydrostatic head of the fluid:

$$\frac{dp}{dh} = \gamma$$

The simulation of a single phase reservoir is, of course, possible from any given initial pressure distribution.

7.6 TREATMENT OF NONLINEARITIES

The methods discussed in Chapter 3 (Section 3.7.2) for the handling of nonlinear terms may be applied directly to two-dimensional problems. No further discussion of this topic is necessary at this time.

7.7 TREATMENT OF INDIVIDUAL WELLS

The pressure at a grid point predicted during reservoir simulation is the average pressure of the block surrounding the grid point. If a well is located in a grid block then the pressure at the well cannot be assumed to be the block pressure. This is particularly true when the size of grid block containing the well is large and/or the flow rate from the well is large. Accurate prediction of the well pressure is possible for single-well problems by the use of r–z co-ordinates as discussed in Section 7.2.3. In areal and cross-sectional models special techniques are required for computing the well pressure from the block pressure predicted by the model.

A reasonable approach to the solution of this problem is to assume that the flow around the well is one dimensional in the radial direction (in cylindrical co-ordinates). Analytical solutions for single-phase one-dimensional flow problems and their applications are discussed in detail by Matthews and Russell (1967) and in the ERCB (1975) Manual. We will discuss here some practical procedures for calculating well pressure p_{wf} from block pressure p_{ij} in an areal model. Extensions of these ideas to other systems is straightforward.

The concept of drainage radius is particularly convenient for solving this problem. This concept, originally introduced by Aronofsky and Jenkins (1954), is discussed in detail in the ERCB (1975) Manual. The drainage radius is defined by

$$\frac{p_{av} - p_{wf}}{p_i q_D} = \ln \frac{r_d}{r_w} \tag{7.52}$$

where p_{av} is the average reservoir pressure.

It is that radius which forces the solution of the above 'steady-state type' equation to match the transient solution. For a finite closed reservoir with external radius r_e, we have

$$\ln \frac{r_d}{r_w} = \frac{1}{2}(\ln t_D + 0{\cdot}809) \qquad \text{if} \quad t_D \leq \frac{1}{4}\left(\frac{r_e}{r_w}\right)^2 \tag{7.53}$$

and

$$r_d = 0{\cdot}472 r_e \qquad \text{if} \quad t_D > \frac{1}{4}\left(\frac{r_e}{r_w}\right)^2 \tag{7.54}$$

where

$$t_D = \frac{kt}{\phi \mu c r_w^2} \tag{7.55}$$

$$q_D = \frac{p_{STC} Z T Q_{STC} \mu}{2\pi T_{STC} p_{av} k \, \Delta z p_i} \tag{7.56}$$

for gas flow, and

$$q_D = \frac{B Q_{STC} \mu}{2\pi k \, \Delta z p_i} \tag{7.57}$$

for liquid flow.

The conversion of the above quantities to field units is discussed in detail in the ERCB (1975) Manual. The equivalent radius of the block shown in Fig. 7.10 may be computed from

$$r_e = \sqrt{\Delta x \, \Delta y / \pi} \tag{7.58}$$

The above procedure makes use of the analytical solution for constant production rate from a cylindrical closed reservoir with one-dimensional flow. It provides a reasonable approximation for square blocks or for rectangular blocks where $\Delta x \simeq \Delta y$. For blocks of other shapes analytical solutions are available and may be used for predicting well pressure (see ERCB, 1975). It is also possible to modify eqn. (7.52) to include skin, and inertial-turbulent (high-velocity) effects not accounted for by Darcy's Law:

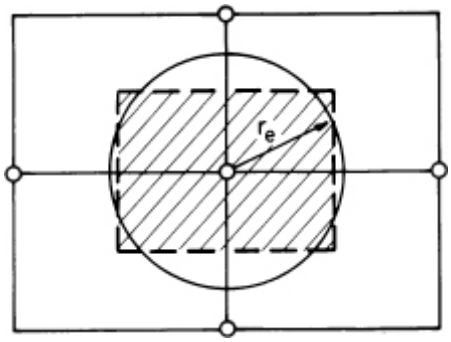


FIG. 7.10. Definition of r_e for eqns. (7.60), (7.63) and (7.64).

$$\frac{p_{av} - p_{wf}}{p_i q_D} = \ln \frac{r_d}{r_w} + S + DQ_{STC} \tag{7.59}$$

In the above equation S is the skin, and D is the inertial-turbulent flow factor which is important only for gas flow.

The above formulae indicate that the drainage radius becomes constant for a sufficiently large time. This corresponds to the pseudo steady-state conditions when pressure declines at a constant rate in all parts of the block and eqn. (7.52) can then be rewritten as:

$$p_{wf} = p_{av} - \frac{Q}{2\pi\lambda\,\Delta z}\left[\ln \frac{r_e}{r_w} - \frac{3}{4}\right] \tag{7.60}$$

where, since the block is assumed to be closed, the average pressure can be obtained by material balance:

$$p_{av} = p_i - \frac{Qt}{\pi r_e^2\,\Delta z \phi c_{av}}$$

Equation (7.60) provides the well pressure in terms of the block average pressure p_{av} and rate Q in reservoir volume units. The relationship between p_{av} and p_{ij} is considered

towards the end of this section. For typical reservoir conditions, the time to reach steady state is of the order of hours to a few days, while the typical time steps used in simulation are several days to a few months. Since the simulated rate must be constant at least over one time step, the pseudo steady-state assumption is usually justified. The exceptions are simulations of short-time behaviour as for example in pulse testing.

Other less rigorous methods based directly on steady-state solutions are also available. Van Poollen *et al.* (1968) obtained the equations for either pseudo steady state for a closed reservoir, or by assuming steady state for incompressible flow with influx at the external boundary. Since the first case leads to eqn. (7.60) derived above, we will consider the incompressible flow case.

The average pressure between r_w and r_e is

$$p_{av} = \frac{\int_{r_w}^{r_e} pr\,dr}{\int_{r_w}^{r_e} r\,dr} = \frac{2}{(r_e^2 - r_w^2)} \int_{r_w}^{r_e} pr\,dr \tag{7.61}$$

while the steady-state pressure distribution is given by

$$p = p_{wf} + \frac{Q}{2\pi\lambda\,\Delta z} \ln \frac{r}{r_w} \tag{7.62}$$

Substituting eqn. (7.62) in eqn. (7.61) and integrating, we obtain

$$p_{wf} = p_{av} - \frac{Q}{2\pi\lambda\,\Delta z(r_e^2 - r_w^2)} \left[r_e^2 \ln \frac{r_e}{r_w} - \frac{(r_e^2 - r_w^2)}{2} \right] \tag{7.63}$$

For $r_e \gg r_w$ this formula simplifies to

$$p_{wf} = p_{av} - \frac{Q}{2\pi\lambda\,\Delta z} \left[\ln \frac{r_e}{r_w} - \frac{1}{2} \right] \tag{7.64}$$

In steady-state flow the boundary pressure is given by eqn. (7.62):

$$p_e = p_{wf} + \frac{Q}{2\pi\lambda\,\Delta z} \ln \frac{r_e}{r_w} \tag{7.65}$$

Equations (7.60), (7.63) and (7.64) are displayed in Fig. 7.10. This figure can be used to estimate quickly p_{wf} by calculating first

$$\frac{Q \ln (r_e/r_w)}{2\pi\lambda \Delta z}$$

(which is equal to $p_e - p_{wf}$ for eqns. (7.63) and (7.64)), and then finding p_{wf} from

$$p_{wf} = p_{av} - \alpha\left(\frac{Q \ln (r_e/r_w)}{2\pi\lambda \Delta z}\right)$$

where α is obtained from Fig. 7.11. This figure also shows the errors in using the approximate formula (7.64), which should not be used if $r_e/r_w < \sim 3$.

The two formulae (7.60) and (7.64), although derived from the two extreme assumptions, differ only by the constant subtracted from the logarithmic term. In a real situation, production of a well will result both in pressure decline and flow across the block boundaries, and the actual pressure will correspond to a general formula

$$p_{wf} = p_{av} - \frac{Q}{2\pi\lambda \Delta z}\left[\ln \frac{r_e}{r_w} - c\right] \tag{7.66}$$

where $\frac{1}{2} \leq c \leq \frac{3}{4}$. Kumar (1977) shows how c is influenced by the strength of influx at the boundary. The unsteady state can also be represented by letting $c = f(t)$

So far three relationships between average pressure p_{av} and well pressure p_{wf} have been presented. It is natural to ask if p_{av} is equal to the block pressure p_{ij}. This is in fact the common procedure used in most simulators. However, Peaceman (1977a) has shown, using numerical and analytical solutions, that $p_{av} \neq p_{ij}$ under steady-state flow conditions. For a square grid, the appropriate relationship between p_{wf} and p_{ij} based on Peaceman's work may be expressed in the form of eqn. (7.66) as:

$$p_{wf} = p_{ij} - \frac{Q}{2\pi\lambda \Delta z}\left[\ln \frac{r_e}{r_w} - 1{\cdot}037\right] \tag{7.67}$$

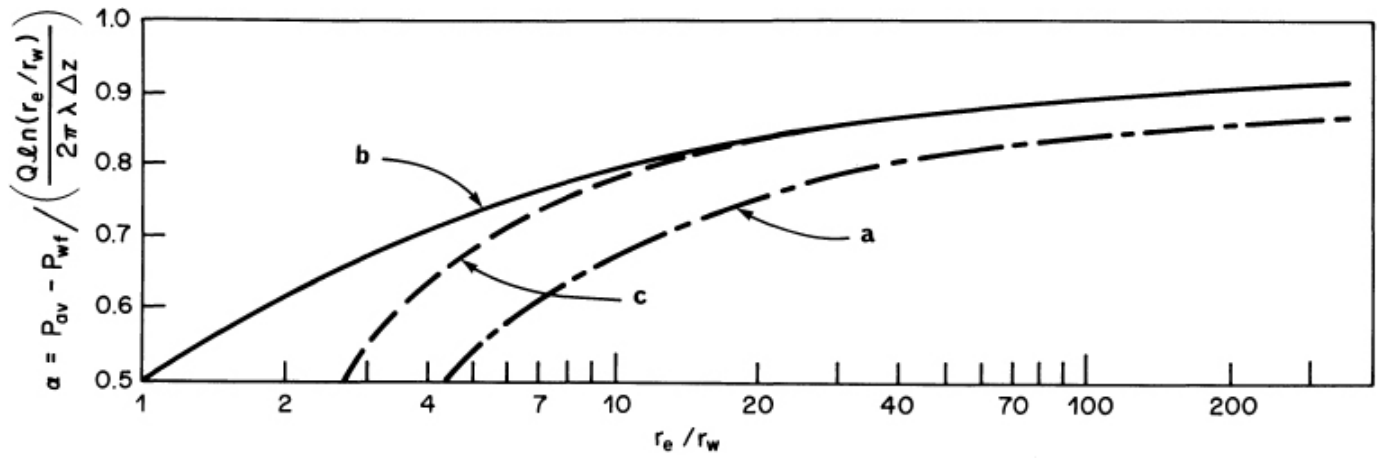


FIG. 7.11. Various methods of calculating well pressure from average pressure, (a) eqn. (7.60); (b) eqn. (7.63); (c) eqn. (7.64).

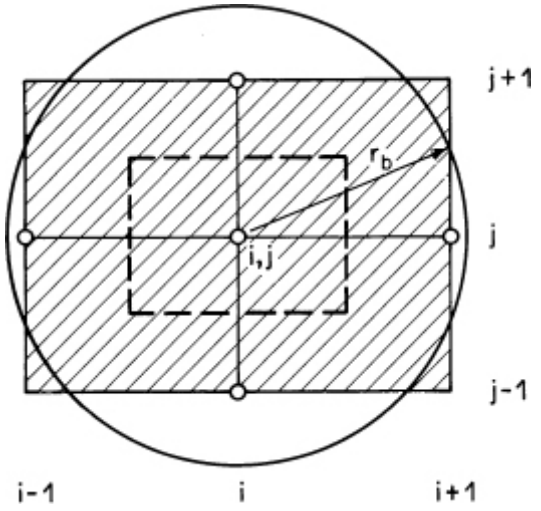


FIG. 7.12. Definition of r_b for eqn. (7.71).

This problem requires further investigation to reveal what is the best procedure particularly when the steady-state flow assumption near the well is not valid.

Closely related ideas developed for the treatment of aquifers will be discussed in Chapter 9.

Another formula for p_{wf} can be derived from the finite-difference transmissibilities for the block (i, j). We can define an average pressure difference between the point (ij) and its neighbours by

$$Q = \overline{\Delta p}\,\Sigma T = (\bar{p} - p_{ij})\Sigma T \tag{7.68}$$
$$= [TX_{i+1/2}(p_{i+1} - p_i) + TX_{i-1/2}(p_{i-1} - p_i)]_j$$
$$+ [TY_{j-1/2}(p_{j-1} - p_j) + TY_{j+1/2}(p_{j+1} - p_j)]_i$$

where

$$\Sigma T = TX_{(i+1/2),j} + TX_{(i-1/2),j} + TY_{i,j+1/2} + TY_{i,j-1/2} \tag{7.69}$$

Now assume that $\bar{p}$ is the pressure at the radius r_b of a circle representing the area shown on Fig. 7.12. Note that this radius r_b, given by

$$r_b = \sqrt{(\Delta x_{i+1/2} + \Delta x_{i-1/2})(\Delta y_{j+1/2} + \Delta y_{j-1/2})/\pi} \tag{7.70}$$

is different from the radius r_e shown on Fig. 7.10. We can now use the radial flow formula to express

$$Q = (\bar{p} - p_{wf})\frac{2\pi\lambda\,\Delta z}{\ln\dfrac{r_b}{r_w}} = (\bar{p} - p_{wf})WI\lambda \tag{7.71}$$

where WI is a productivity coefficient of the well relative to r_b. This coefficient may be modified to account for partial perforations, well stimulation or skin effect. Equations (7.71) and (7.68) can be solved to obtain

$$p_{ij} - p_{wf} = Q\frac{1 - \dfrac{WI}{\Sigma T}\lambda}{WI\lambda} = Q/TW \tag{7.72}$$

This equation differs from eqn. (7.63) or (7.64) in that it takes into account not only the permeability around the well (through WI) but also the permeability variations between the grid points around the well. However, the approximation is valid only if $WI < \sum T/\lambda$.

7.8 EQUATIONS IN MATRIX FORM

The discretised equation for any of the two-dimensional implicit models discussed in the previous sections of this chapter may be written in the form:

$$c_{ij}p^{n+1}_{i-1,j} + ax_{ij}p^{n+1}_{ij} + b_{ij}p^{n+1}_{i+1,j} + g_{ij}p^{n+1}_{i,j-1} + ay_{ij}p^{n+1}_{ij} + f_{ij}p^{n+1}_{i,j+1} + \varphi_{ij}p^{n+1}_{ij} = d_{ij} \tag{7.73}$$

where the coefficients c_{ij}, ax_{ij}, b_{ij}, g_{ij}, ay_{ij}, f_{ij}, φ_{ij} and d_{ij} depend upon the type of model and the method of discretisation. For example, the areal (x–y) model with the backward difference approximation for the time derivative (eqn. (7.12)) yields:

$$c_{ij} = -\mathrm{T}X_{(i-1/2),j} \tag{7.74}$$

$$b_{ij} = -\mathrm{T}X_{(i+1/2),j} \tag{7.75}$$

$$g_{ij} = -\mathrm{T}Y_{i,j-1/2} \tag{7.76}$$

$$f_{ij} = -\mathrm{T}Y_{i,j+1/2} \tag{7.77}$$

$$ax_{ij} = \mathrm{T}X_{(i-1/2),j} + \mathrm{T}X_{(i+1/2),j} = -(c_{ij} + b_{ij}) \tag{7.78}$$

$$ay_{ij} = \mathrm{T}Y_{i,j-1/2} + \mathrm{T}Y_{i,j+1/2} = -(g_{ij} + f_{ij}) \tag{7.79}$$

$$\varphi_{ij} = \frac{V_{ij}\beta_{ij}}{\Delta t} \tag{7.80}$$

$$d_{ij} = -Q_{ij} + \varphi_{ij}p^{n}_{ij} \tag{7.81}$$

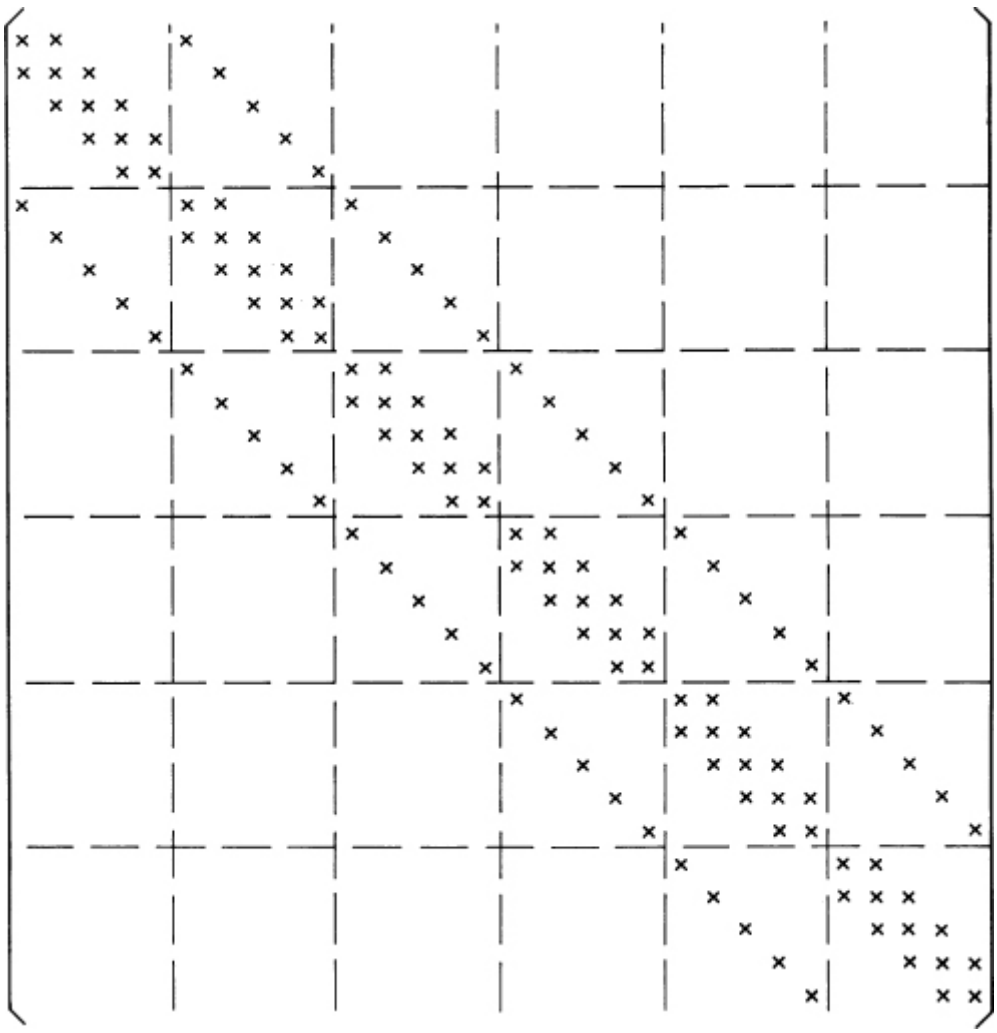


FIG. 7.13. The form of matrix in eqn. (7.84).

We will also find it convenient to define

$$a_{ij} = ax_{ij} + ay_{ij} + \varphi_{ij} \tag{7.82}$$

These equations may be written in matrix form by selecting some ordering of the unknowns. The natural ordering is to order the unknowns by x-grid lines starting with the line for $j = 1$. Then the unknown vector is

$$(\boldsymbol{P}^{n+1})^{\mathrm{T}} = [p_{1,1}, p_{2,1}, p_{3,1}, p_{4,1}, p_{5,1}, p_{1,2}, \ldots, p_{5,6}]^{n+1} \tag{7.83}$$

and the matrix equation may be written as

$$\mathbf{A}\boldsymbol{P}^{n+1} = \boldsymbol{d} \tag{7.84}$$

The form of the coefficient matrix is shown in Fig. 7.13. Equations for cross-sectional and radial models are of the same form with appropriate definitions for coefficients.

In the next chapter we will consider direct solution and iterative methods for eqn. (7.84). In the following section of this chapter we present some methods for obtaining approximations to the solution vector $\boldsymbol{P}$. These methods are closely related to the form of the partial differential equation and for this reason they are presented in this chapter.

7.9 SPECIAL METHODS FOR 2-D PROBLEMS

The methods to be discussed here are of two types. One type of methods referred to as *alternating direction explicit* (ADE) methods involve no matrix computations and reduce the problem to a form similar to the form of explicit difference equations. The ADE methods are, however, unconditionally stable (for linear problems) in contrast to the classical explicit method. The second type of methods referred to as *alternating direction implicit* (ADI) methods involve the solution of tridiagonal matrix equations and they are related to the classical implicit and Crank–Nicolson methods.

The ADE and ADI methods produce a set of difference equations that are much easier to solve than the matrix equation presented in the previous section. This simplification is, of course, obtained at the expense of some reduction in accuracy and stability. The practical application of these methods is restricted to single-phase flow problems and relatively simple multiphase problems.

7.9.1 Alternating Direction Explicit (ADE) Methods

During the early stages of the development of digital computers, machines were relatively slow with very limited core storage. This provided considerable motivation for the development of stable explicit difference approximations for partial differential equations of practical interest. The book by Saul'jev (1964) provides a comprehensive treatment of various forms of the ADE method. Further discussion of these methods for two-dimensional problems is provided by Larkin (1964) and Barakat and Clark (1966), and the application of these methods to gas reservoir simulation is discussed by Quon *et al.* (1966) and Carter (1966). The method has some obvious advantages over the standard explicit method.

The starting point for the development of ADE methods can be either the backward difference implicit formula or the Crank–Nicolson implicit formula. Let u and v be two approximations to the solution of the difference equation (eqn. (7.12)) with the left-hand side evaluated at the unknown level of time. For the sake of keeping our discussion simple, we will assume that the x and y co-ordinates are in the horizontal plane, and hence the gravity terms disappear. We can write the difference approximation (7.12) as

$$\varphi_{ij}\Delta_t p_{ij}^{n+1} = TX_{(i-1/2),j}(p_{i-1,j} - p_{i,j})^{n+l_1} + TX_{(i+1/2),j}(p_{i+1,j} - p_{i,j})^{n+l_2} \quad (7.85)$$
$$+ TY_{i,j-1/2}(p_{i,j-1} - p_{i,j})^{n+l_3} + TY_{i,j+1/2}(p_{i,j+1} - p_{i,j})^{n+l_4}$$

where for the standard backward difference approximation

$$l_1 = l_2 = l_3 = l_4 = 1$$

and

$$\Delta_t p_{ij}^{n+1} = p_{ij}^{n+1} - p_{ij}^{n}$$

For standard ordering of unknowns eqn. (7.85) becomes explicit with only one unknown per equation if $l_1 = l_3 = 1$ and $l_2 = l_4 = 0$. It is easy to see that instead of starting computations in the lower left-hand corner (Fig. 7.4) we could start at one of the other three corners with different choices of l values. The approximations to the difference equations obtained in this manner may be written as:

Approximation 1 ($l_1 = l_3 = 1, l_2 = l_4 = 0$)

$$\varphi_{ij}\Delta_t v_{ij}^{n+1} = TX_{(i-1/2),j}(v_{i-1,j} - v_{ij})^{n+1} + TX_{(i+1/2),j}(v_{i+1,j} - v_{ij})^{n} \quad (7.86)$$
$$+ TY_{i,j-1/2}(v_{i,j-1} - v_{ij})^{n+1} + TY_{i,j+1/2}(v_{i,j+1} - v_{ij})^{n}$$

Approximation 2 ($l_1 = l_3 = 0, l_2 = l_4 = 1$)

$$\varphi_{ij}\Delta_t u_{ij}^{n+1} = TX_{(i-1/2),j}(u_{i-1,j} - u_{ij})^{n} + TX_{(i+1/2),j}(u_{i+1,j} - u_{ij})^{n+1} \quad (7.87)$$
$$+ TY_{i,j-1/2}(u_{i,j-1} - u_{ij})^{n} + TY_{i,j+1/2}(u_{i,j+1} - u_{ij})^{n+1}$$

Each of the above two approximations introduces errors. However, these errors are partially compensated by the application of eqns. (7.86) and (7.87) for alternate time steps. Since eqns. (7.86) and (7.87) must be applied by re-ordering the unknowns (changing the direction of the sweep) the method is known as an *alternating direction explicit* method.

The use of both equations for each time step and then averaging the results has also been considered by Larkin (1964):

$$w_{ij}^{n+1} = \frac{u_{ij}^{n+1} + v_{ij}^{n+1}}{2}$$

where w_{ij}^{n+1} is a new approximation for p_{ij}^{n+1} and with this scheme w_{ij}^{n} should be used instead of v_{ij}^{n} and u_{ij}^{n} in eqns. (7.86) and (7.87).

Two other approximations are obtained by selecting the following values of l values in eqn. (7.85):

Approximation 3

$$l_1 = l_4 = 0$$
$$l_2 = l_3 = 1$$

Approximation 4

$$l_2 = l_3 = 0$$
$$l_1 = l_4 = 1$$

Clearly one can use various combinations of the above four approximations. It is also possible to develop similar schemes by starting with the Crank–Nicolson method instead of eqn. (7.85). Methods of this type find little use in modern reservoir simulation studies where fast computers with large cores are available. For this reason we will not present a detailed discussion.

7.9.2 Alternating Direction Implicit (ADI) and Related Methods

The solution of two-dimensional reservoir simulation problems may be reduced to the solution of a series of one-dimensional reservoir simulation problems. Since we already know how to solve tridiagonal matrix equations, no new matrix techniques are required for the solution of reservoir simulation problems being considered in this chapter. Methods of this type are also known in the literature by names such as *splitting-up methods* (Marchuk, 1975), *fractional step methods* (Yanenko, 1971) and *locally one-dimensional methods* (Mitchell, 1969).

The ADI method was first introduced by Peaceman and Rachford (1955) and Douglas (1955) and further developed by these and many other investigators. The ADI methods to be discussed here are sometimes called ‘non-iterative ADI’ methods in order to distinguish them from ADI methods for the solution of matrix equations by iteration (see Chapter 8). We will show in this section how these methods can be applied to the solution of single-phase flow problems in two dimensions.

Peaceman–Rachford (1955) *method.* Let us consider the following backward difference approximation for the areal problem

$$\Delta_x \mathrm{T} X \Delta_x p_{ij}^{n+1} + \Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n+1} = \varphi_{ij}(p_{ij}^{n+1} - p_{ij}^{n}) + Q_{ij} \tag{7.88}$$

If we want to reduce this problem to the solution of a system of one-dimensional problems, then only one of the terms on the left side of eqn. (7.88) may be evaluated at the unknown level of time. This yields a two-step process. The first step of this procedure is

$$\Delta_x \mathrm{T} X \Delta_x p_{ij}^{*} + \Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n} = V_{ij}\beta_{ij}\frac{p_{ij}^{*} - p_{ij}^{n}}{\frac{1}{2}\Delta t} + Q_{ij} \tag{7.89}$$

where p_{ij}^{*} {~?~IM: is an intermediate solution which may be looked upon as an approximation of $p_{ij}^{n+1/2}$.

Equation (7.89) may be solved for all grid points by considering unknowns for only one value of j (or one x-grid line) at a time and by using the Thomas algorithm discussed in Chapter 4. As shown by Douglas (1961) the use of eqn. (7.89) alone results in a method of limited stability. In order to obtain an unconditionally stable method, the second step is introduced which involves the implicit approximation of the second term on the left side of eqn. (7.88):

$$\Delta_x \mathrm{T} X \Delta_x p_{ij}^{*} + \Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n+1} = V_{ij}\beta_{ij}\frac{p_{ij}^{n+1} - p_{ij}^{*}}{\frac{1}{2}\Delta t} + Q_{ij} \tag{7.90}$$

Equation (7.90) may be solved for all grid points by considering unknowns for only one value of i (or one y-grid line) at a time. Unknowns along each grid line have a matrix equation with a tridiagonal coefficient matrix.

The above two-step procedure where eqns. (7.89) and (7.90) are used alternately is unconditionally stable. In order to investigate the accuracy of this procedure it is necessary to eliminate the intermediate solution from these two equations. For the linear diffusion equation on a rectangle the resulting equation is locally second order correct both in time and space and may be viewed as a perturbation of the Crank–Nicolson method (Douglas, 1961). This type of analysis shows that the nonlinear coefficients and source terms should be evaluated at the $n + ½$ level of time for this ADI method.

Briggs and Dixon (1968) have discussed the application of this method to reservoir simulation problems. They find that this method produces oscillatory results for large time steps. Such behaviour is characteristic of the Crank–Nicolson (CN) procedure and since the ADI method is a perturbation of the CN method such behaviour is observed for it also.

Douglas (1962) *method.* Since the Peaceman and Rachford (1955) method turns out to be a perturbation of the Crank–Nicolson approximation, it should be possible to develop an alternating direction method directly from the Crank–Nicolson approximation instead of the backward difference approximation.

The intermediate solution may be obtained from

$$\tfrac{1}{2}\Delta_x \mathrm{T} X \Delta_x (p_{ij}^* + p_{ij}^n) + \Delta_y \mathrm{T} Y \Delta_y p_{ij}^n = \varphi_{ij}(p_{ij}^* - p_{ij}^n) + Q_{ij} \tag{7.91}$$

and the final solution at the new level of time from

$$\tfrac{1}{2}\Delta_x \mathrm{T} X \Delta_x (p_{ij}^* + p_{ij}^n) + \tfrac{1}{2}\Delta_y \mathrm{T} Y \Delta_y (p_{ij}^{n+1} + p_{ij}^n) = \varphi_{ij}(p_{ij}^{n+1} - p_{ij}^n) + Q_{ij} \tag{7.92}$$

This method is a perturbation of the Crank–Nicolson method and it is equivalent to the Peaceman–Rachford method on a rectangle. From a computational point of view, it is more convenient to replace eqn. (7.92) by an equation that is obtained by subtracting the above two equations:

$$\tfrac{1}{2}\Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n+1} - \tfrac{1}{2}\Delta_y \mathrm{T} Y \Delta_y p_{ij}^n = \varphi_{ij}(p_{ij}^{n+1} - p_{ij}^*) \tag{7.93}$$

For two-dimensional problems this method has no real advantage over the Peaceman–Rachford method. Consideration of three-dimensional problems will reveal the real advantage of the Douglas method.

Douglas and Rachford (1956) *method*. The above two methods are both perturbations of the Crank–Nicolson method. A perturbation of the backward difference approximation is provided by the following scheme:

$$\Delta_x \mathrm{T} X \Delta_x p_{ij}^* + \Delta_y \mathrm{T} Y \Delta_y p_{ij}^n = \varphi_{ij}(p_{ij}^* - p_{ij}^n) + Q_{ij} \tag{7.94}$$

$$\Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n+1} - \Delta_y \mathrm{T} Y \Delta_y p_{ij}^n = \varphi_{ij}(p_{ij}^{n+1} - p_{ij}^*) \tag{7.95}$$

Time-dependent nonlinear coefficients and source terms, if any, should be evaluated at the $n + 1$ level of time for this method.

Splitting-up or locally one-dimensional (*LOD*) *methods.* Let us define

$$F_{ij}^n = (\Delta \mathrm{T} \Delta p_{ij}^n - Q_{ij})/(V_{ij}\beta_{ij}) \tag{7.96}$$

Then the solution for the areal problem being discussed in this section may be obtained from the following scheme:

$$\Delta_x \mathrm{T} X \Delta_x \xi^{n+1/2} = 2\varphi_{ij}(F_{ij}^n - \xi^{n+1/2}) \tag{7.97}$$

$$\Delta_y \mathrm{T} Y \Delta_y \xi^{n+1} = 2\varphi_{ij}(\xi^{n+1} - \xi^{n+1/2}) \tag{7.98}$$

$$p_{ij}^{n+1} = p_{ij}^{n} + \Delta t \xi^{n+1} \tag{7.99}$$

This method is also a perturbation of the Crank–Nicolson procedure (Marchuk, 1975, p. 147). Other variations of this approach are discussed by Marchuk (1975) and Mitchell (1969). Clearly the application of eqn. (7.97) must be by x-grid lines and (7.98) by y-grid lines. This shows the similarity of this procedure to the ADI method.

Predictor–corrector methods. Let us first predict the solution at the $n + ½$ level of time using the following two step procedure:

$$\Delta_x \mathrm{T} X \Delta_x p_{ij}^{*} = 2\varphi_{ij}(p_{ij}^{*} - p_{ij}^{n}) + Q_{ij}^{n+1/2} \tag{7.100}$$

$$\Delta_y \mathrm{T} Y \Delta_y p_{ij}^{n+1/2} = 2\varphi_{ij}(p_{ij}^{n+1/2} - p_{ij}^{*}) \tag{7.101}$$

The solution at the $n + 1$ level of time is obtained from

$$p_{ij}^{n+1} = p_{ij}^{n} + \frac{1}{2\varphi_{ij}}[\Delta \mathrm{T} \Delta p_{ij}^{n+1/2} - Q_{ij}^{n+1/2}] \tag{7.102}$$

This approach is discussed in detail by Marchuk (1975, p. 162). It also turns out to be a perturbation of the Crank–Nicolson method.

7.9.3 Comparison of Methods

Coats and Terhune (1966) provide a comparison of the ADE and ADI methods for the standard diffusion equation

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + q = \frac{\partial U}{\partial t}$$

on a rectangle. These authors point out that ADE fails to preserve no-flow conditions at exterior boundaries for the block-centered grid and this causes material balance errors. Coats and Terhune also found that the ADI method was considerably more accurate than the ADE method and it required only about 60% more work.

A more recent comparison for the same equation was provided by Sheffield (1970). He defines a '*Line Saul'jev*' method which is equivalent to two steps using eqn. (7.85) with $\Delta t/2$. In the first step, p_{ij}^{n+1} on the left side is replaced by $p_{ij}^{n+1/2}$ and the

parameters are $l_1 = l_2 = l_3 = ½$, $l_4 = 0$. In the second step, p_{ij}^{n+1} is calculated and $l_1 = l_2 = l_4 = 1$, $l_3 = 0$. Thus, both steps are solved by lines in the x-direction, but in alternating order for j. Sheffield compared this method with the Saul'jev method (alternating use of eqns. (7.86) and (7.87)) and ADI. ADI was superior to Line Saul'jev for a problem with square grid, but the Line Saul'jev was far better for a problem with elongated grid. The 'Point Saul'jev' method was least accurate in both cases. This result is typical of ADI methods. This method does not perform well for problems with anisotropic coefficients (produced either by elongated grid or large contrast in k_x and k_y). The same behaviour is also shared by iterative ADI methods which converge very slowly for such problems (see Chapter 8, Fig. 8.19).

There has been no interest in ADE or ADI methods in recent literature.

7.10 METHODS OF GRID CONSTRUCTION

In this section, we will discuss various methods of constructing two-dimensional grids. We will restrict ourselves to grids based on rectangles (or topologically equivalent grids), which correspond to the five-point difference approximation. Other types of grids (triangular, hexagonal, etc.) have found very little use in reservoir simulation so far.

7.10.1 Irregular Grid in 2-D

The motivation for the use of irregular grid is provided by our desire to obtain good definition by use of small increments in places where the pressure gradients are expected to be large and use large increments elsewhere. A natural approach is to construct irregular grids in both directions and superimpose them.

This will result in a grid with increments Δx_i, Δy_j, as it has been used throughout this chapter. Figure 7.14 gives an example of an areal grid, refined around wells. Another typical example is a radial r–z grid with variable increments in the radial direction. This kind of irregular grid is used routinely in reservoir simulation. The description of the grid is simple and the programming effort is about the same as for a regular grid.

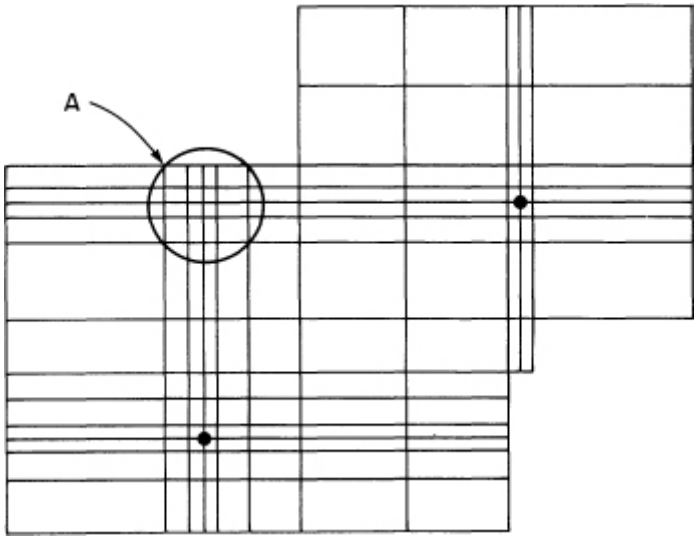


FIG. 7.14. Typical irregular grid for an areal model.

However, it is easy to see that such 'standard' grid is not optimal as far as the number of grid points necessary to cover the domain is concerned. For example, the grid in Fig. 7.14 is also refined in the circled area 'A', where a coarse grid would be sufficient. In order to obtain more 'optimal' grid, one has to use composite (or locally refined) grids, such as shown on Fig. 7.15. Such grid refinement schemes were studied by Ciment (1971) and Ciment and Sweet (1973). Unfortunately, the resulting difference approximations are more complex, and their form is not the same for all grid points. For this reason, local grid refinement has not been widely used in reservoir simulation.

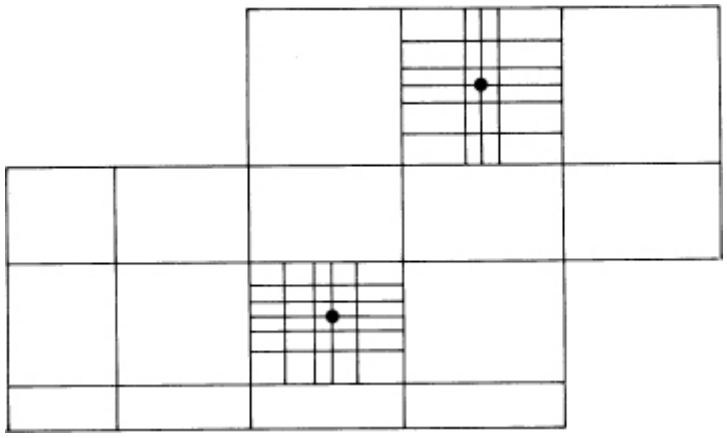


FIG. 7.15. Local refinement of grid.

We remark in this connection, that other grid systems, such as triangular, are more suitable for local grid refinement than a rectangular grid. For additional information on grid refinement see Kafka (1968), Osher (1970), Browning *et al.* (1973) and Girault (1974).

In conclusion, we note that use of irregular grid decreases the accuracy of the approximation to $O(\Delta x) + O(\Delta y)$, unless special techniques are employed.

7.10.2 Use of a Curvilinear Grid

Although the majority of reservoir problems can be simulated with rectangular grids, there are many instances in which the shape of the reservoir lends itself more naturally to a curvilinear co-ordinate system. For example, a cross-section with a considerable dip can be better described by the grid shown on Fig. 7.16b rather than the usual grid shown on Fig. 7.16a.

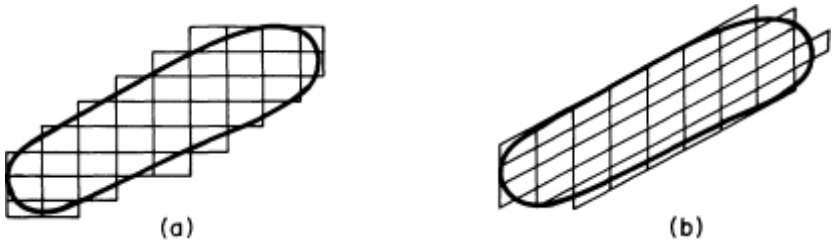


FIG. 7.16. Grids for a cross-sectional model of a reservoir with a dip.

Similarly, an areal flow in a repeated five-spot pattern can be better described by the grid on Fig. 7.17b which follows the streamlines than by a rectangular grid (Fig. 7.17a).

Curvilinear co-ordinate systems are common in classical hydro-dynamics. In reservoir simulation, use of curvilinear grid was studied by several authors for multiphase flow applications (Hirasaki and O'Dell, 1970; Sonier and Chaumet, 1974; Robertson and Woo, 1976). However, the concept is equally applicable, and easier to understand, for single-phase flow and therefore will be treated here. Specific comments pertaining to multiphase flow will be given in Chapter 9 (Section 9.7).

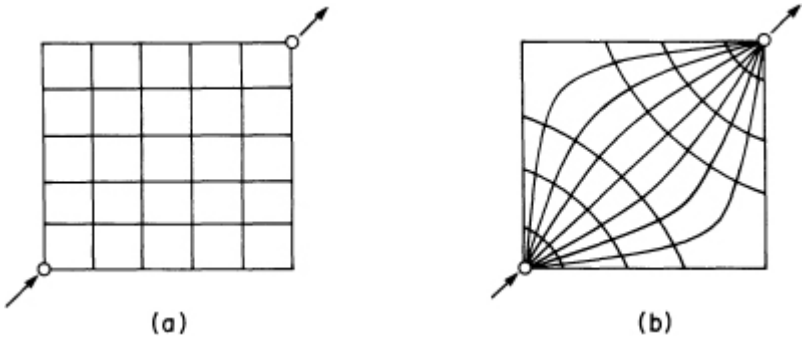


FIG. 7.17. Grids for the simulation of a five-spot in waterflooding.

We will now briefly summarise the transformation of equations into curvilinear co-ordinates. The details can be found in books on tensor analysis, for example, McConnell (1957), Aris (1962) and Sokolnikoff (1951).

Let us consider two co-ordinate systems: a rectangular system x, y, which we will denote here as (y^1, y^2), and a general co-ordinate system (x^1, x^2). The two systems are related by equations

$$x^i = f^i(y^1, y^2) \quad i = 1, 2 \tag{7.103}$$

The functions f^i define the transformation of the co-ordinate system. The form of the differential equation in the x^i-co-ordinates depends on these functions. The new co-ordinate system will be well defined if the determinant of the transformation is nonzero, i.e.,

$$\left|\frac{\partial x^r}{\partial y^s}\right| \equiv \begin{vmatrix} \dfrac{\partial x^1}{\partial y^1} & \dfrac{\partial x^2}{\partial y^1} \\ \dfrac{\partial x^1}{\partial y^2} & \dfrac{\partial x^2}{\partial y^2} \end{vmatrix} \neq 0$$

The operator A for the flow terms can be written using the summation convention as

$$\mathrm{A} = \frac{\partial}{\partial y^i}\left[\lambda \frac{\partial}{\partial y^i}\right] \equiv \frac{\partial}{\partial y^1}\left[\lambda \frac{\partial}{\partial y^1}\right] + \frac{\partial}{\partial y^2}\left[\lambda \frac{\partial}{\partial y^2}\right]$$

This can be expressed in the x^i-co-ordinates as

$$\begin{aligned} \mathrm{A} &= \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^i}\left[\sqrt{g}\, g^{ij} \lambda \frac{\partial}{\partial x^j}\right] \\ &\equiv \frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial x^1}\left(\sqrt{g}\, g^{11} \lambda \frac{\partial}{\partial x^1}\right) + \frac{\partial}{\partial x^2}\left(\sqrt{g}\, g^{22} \lambda \frac{\partial}{\partial x^2}\right)\right. \\ &\quad \left. + \frac{\partial}{\partial x^1}\left(\sqrt{g}\, g^{12} \lambda \frac{\partial}{\partial x^2}\right) + \frac{\partial}{\partial x^2}\left(\sqrt{g}\, g^{21} \frac{\partial}{\partial x^1}\right)\right] \end{aligned} \tag{7.104}$$

where g is the determinant of the metric tensor g_{ij}:

$$g_{ij} = \frac{\partial y^k}{\partial x^i}\frac{\partial y^k}{\partial x^j} \qquad g = |g_{ij}| \tag{7.105}$$

and g^{ij} are the components of the conjugate metric tensor, which can be calculated as $g^{ij} = G^{ij}/g$ where G^{ij} are the cofactors of g_{ij} in the matrix $\{g_{ij}\}$. They can also be expressed more conveniently as (McConnell, 1957)

$$g^{ij} = \frac{\partial x^i}{\partial y^k}\frac{\partial x^j}{\partial y^k} = \frac{\partial f^i}{\partial y^k}\frac{\partial f^j}{\partial y^k} \tag{7.106}$$

which does not require inverse functions to f^i.

In general, the operator A in the form (7.104) involves cross-derivative terms. However, these terms will disappear for an important class of co-ordinate systems, called *orthogonal*.

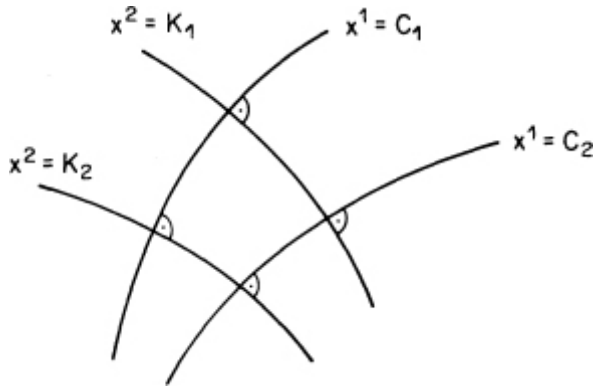


FIG. 7.18. An orthogonal co-ordinate system.

A system x^i is called *orthogonal* if the co-ordinate curves are mutually orthogonal everywhere (Fig. 7.18).

Any orthogonal system has the following important properties:

1. *The necessary and sufficient condition for a system x^i to be orthogonal is*

$$g_{ij} = 0 \qquad for \quad i \neq j \tag{7.107}$$

2. *If the system x^i is orthogonal, then*

$$g^{ij} = 0 \qquad for \quad i \neq j \tag{7.108}$$

$$g = g_{11}g_{22}g_{33} \qquad g^{ii} = \frac{1}{g_{ii}} \tag{7.109}$$

Therefore, for an orthogonal system the operator A will be

$$\mathrm{A} \equiv \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^i}\left[\sqrt{g} g^{ii} \lambda \frac{\partial}{\partial x^i}\right] \tag{7.110}$$

$$\equiv \frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial x^1}\left(\sqrt{g} g^{11} \lambda \frac{\partial}{\partial x^1}\right)+\frac{\partial}{\partial x^2}\left(\sqrt{g} g^{22} \lambda \frac{\partial}{\partial x^2}\right)\right]$$

Note that in the foregoing development we have assumed λ to be a scalar function of x. The analysis becomes more complicated when the tensorial character of k (in λ) is considered.

An example of an orthogonal system is the polar (or in 3-D cylindrical) co-ordinate system where $x^1 = r$, $x^2 = \theta$. The transformation is shown in Exercise 7.1 and it results in r-direction terms stated earlier for eqn. (7.4).

The discretisation of the conservation equations in the form of (7.104) or (7.110) are developed rigorously by Hirasaki and O'Dell (1970). When the system is orthogonal, the transmissibilities can be calculated from intuitive geometric concepts, defined in Fig. 7.19.

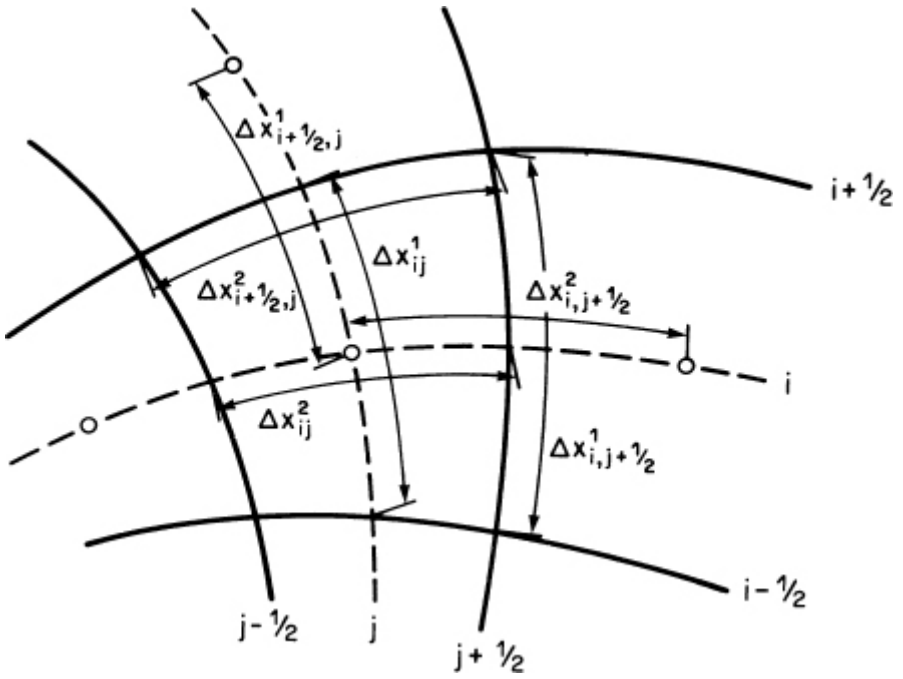


FIG. 7.19. Grid for a curvilinear co-ordinate system.

Since in general, $\mathrm{T} = \lambda A/\Delta L$ where A is a cross-sectional area and ΔL the distance between grid points, we can write immediately

$$TX^1_{(i+1/2),j} = \lambda X^1_{(i+1/2),j} \frac{\Delta x^2_{(i+1/2),j}\, \Delta z_{(i+1/2),j}}{\Delta x^1_{(i+1/2),j}} \tag{7.111}$$

$$TX^2_{i,j+1/2} = \lambda X^2_{i,j+1/2} \frac{\Delta x^1_{i,j+1/2}\, \Delta z_{i,j+1/2}}{\Delta x^2_{i,j+1/2}}$$

where the pertinent quantities are defined on the figure.

Similarly, the pore volume is defined as

$$V_{p_{ij}} = \Delta z_{ij}\, \Delta x^1_{ij}\, \Delta x^2_{ij} \tag{7.112}$$

This approach was used by Sonier and Chaumet (1974), who in addition neglected the difference between $A_{(i+\frac{1}{2}),j}$, $A_{i,j+\frac{1}{2}}$ and A_{ij}. Finite-difference modelling of heat conduction problems in general orthogonal curvilinear co-ordinate systems is also considered by Schneider *et al.* (1975).

As we have seen above, eqns. (7.111) and (7.112) apply only if the grid is derived from an orthogonal co-ordinate system. A convenient way of defining such a grid is through the solution of the potential flow distribution with sources and sinks in well locations. As it is well known, equipotential lines and streamlines of the solution are mutually orthogonal (Lamb, 1932). Potential solutions for many configurations of wells have been published (Muskat, 1937; Morel-Seytoux, 1966). Actual flow in the reservoir will not follow the potential flow lines, because the potential solution is based upon the assumptions of constant permeability and porosity and incompressible flow. This is of no consequence, however, since the potential flow assumption is used only to construct the grid and does not preclude the numerical model from simulating flow in the direction perpendicular to the grid lines which are the potential flow streamlines. This is in contrast to the so-called 'stream-tube' model, in which the crossflow between the stream-lines is neglected.

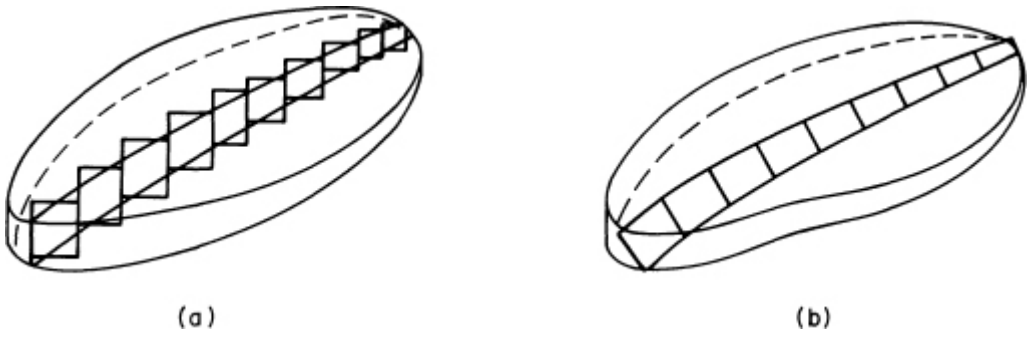


FIG. 7.20. Two types of grid in a cross-sectional model.

The method of grid construction just described has been suggested by Sonier and Chaumet (1974) in conjunction with areal simulation of multiphase flow, but it can be used equally well for any reservoir configuration.

A more detailed analysis of curvilinear grids is found in Hirasaki and O'Dell (1970). These authors show that sizeable errors can result when a dipping reservoir is modelled areally by a block model, as shown for one particular cross-section on Fig. 7.20a.

This is because the cross-sectional area in the transmissibility term is calculated incorrectly. The grid should be constructed such that the co-ordinate surface coincides with the top and bottom of the reservoir. As shown by these authors, this approach will give small errors even when the cross-derivative terms are neglected for a grid that is not orthogonal, provided all distances are measured along the surface and also perpendicular to it (Fig. 7.20b). A grid that is approximately orthogonal can be constructed numerically, graphically or by means of an analog computer (Karplus, 1958).

One problem associated with curvilinear grid is the proper calculation of directional permeabilities. When the permeability is not isotropic, the correct treatment would require the cross-derivative terms even if the grid is orthogonal. The derivation of the transmissibility terms is considered in Exercise 7.2 for the case when the principal values of the permeability tensor are k_x and k_y in the directions of the Cartesian co-ordinates. The magnitude of the cross-derivative term is proportionate to $|k_x - k_y|$, but the effect of neglecting this term has apparently not been studied in the literature. As a consequence, use of curvilinear grid is not recommended in cross-sectional and single-well problems, where stratification into layers is often present. Grid lines in this case should follow geological layers. In such problems, the benefit of curvilinear grid would be small because the actual flow pattern will be far from the potential flow and large errors will be introduced unless the cross-derivative terms are included in the finite-difference equations. On the other hand, curvilinear grid can be quite effective in an areal situation and it brings in additional benefits for multiphase flow (Chapter 9, Section 9.7).

7.11 CONCLUDING REMARKS

The material covered in this chapter is important for development in later chapters. Although single-phase flow is of limited use in simulation, several multiphase-flow techniques generate single-phase type of equations (IMPES, SEQ). Single pressure equation is also solved in miscible problems and compositional problems.

Of course, the equations describing single-phase problems are not peculiar to reservoir mechanics and, consequently, there is a large body of literature in other areas

dealing with their numerical solution. It would not be proper to deal with these applications here, but the reader should be aware that ideas developed in other branches of physics and engineering can prove fruitful in reservoir simulation.

EXERCISES

Exercise 7.1

Expand the left-hand side of

$$\nabla[\lambda(\nabla p - \gamma \nabla z)] = \beta \frac{\partial p}{\partial t} + q \tag{A}$$

in 2-D polar $(r - \theta)$ *co-ordinates.*

Solution Outline

Use eqn. (7.110) with $x^1 = r$, $x^2 = \theta$. Since

$$y^1 = x = x^1 \cos x^2$$
$$y^2 = y = x^1 \sin x^2$$

the metric tensor is

$$\{g_{ij}\} = \begin{bmatrix} \cos^2\theta + \sin^2\theta & 0 \\ 0 & r^2(\cos^2\theta + \sin^2\theta) \end{bmatrix} \qquad g = r^2$$

$$g^{11} = G^{11}/g = 1, g^{22} = G^{22}/g = 1/r^2, g^{12} = g^{21} = 0$$

Therefore eqn. (7.104) used in (A) gives

$$\frac{1}{r}\frac{\partial}{\partial r}\left[\lambda r\left(\frac{\partial p}{\partial r} - \gamma \frac{\partial z}{\partial r}\right)\right] + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left[\lambda\left(\frac{\partial p}{\partial \theta} - \gamma \frac{\partial z}{\partial \theta}\right)\right] = \beta \frac{\partial p}{\partial t} + q$$

Exercise 7.2

Derive the expansion of

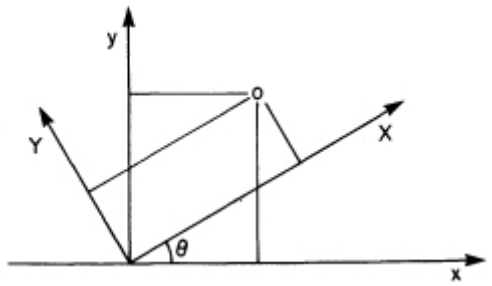
$$\nabla(\lambda \nabla \Phi)$$

for a Cartesian co-ordinate system rotated by an angle θ *to the main axes of tensor* λ.

Solution Outline

Consider two systems: x,y and $X,\ Y$ (below) and assume that the main axes of λ are in the X and Y directions:

$$\nabla(\lambda\nabla\Phi) = \frac{\partial}{\partial X}\left(\lambda_X \frac{\partial \Phi}{\partial X}\right) + \frac{\partial}{\partial Y}\left(\lambda_Y \frac{\partial \Phi}{\partial Y}\right) \tag{A}$$



Since

$$x = X\cos\theta - Y\sin\theta$$
$$y = X\sin\theta - Y\cos\theta$$
$$\frac{\partial}{\partial X} = \frac{\partial}{\partial x}\frac{\partial x}{\partial X} + \frac{\partial}{\partial y}\frac{\partial y}{\partial X} = \cos\theta\frac{\partial}{\partial x} + \sin\theta\frac{\partial}{\partial y}$$
$$\frac{\partial}{\partial Y} = \frac{\partial}{\partial x}\frac{\partial x}{\partial Y} + \frac{\partial}{\partial y}\frac{\partial y}{\partial Y} = -\sin\theta\frac{\partial}{\partial x} + \cos\theta\frac{\partial}{\partial y}$$

and substitution gives

$$\nabla(\lambda\nabla\Phi) = \frac{\partial}{\partial x}\left(\lambda_{xx}\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial x}\left(\lambda_{xy}\frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial y}\left(\lambda_{yx}\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda_{yy}\frac{\partial \Phi}{\partial y}\right) \tag{B}$$

where

$$\lambda_{xx} = \lambda_X\cos^2\theta + \lambda_Y\sin^2\theta \tag{C}$$
$$\lambda_{yy} = \lambda_X\sin^2\theta + \lambda_Y\cos^2\theta$$
$$\lambda_{xy} = \lambda_{yx} = (\lambda_X - \lambda_Y)\sin\theta\cos\theta$$

CHAPTER 8

SOLUTION OF PENTADIAGONAL MATRIX EQUATIONS

8.1 INTRODUCTION

Many of the single-phase two-dimensional problems discussed in the last chapter can be written in the form

$$\frac{\partial}{\partial x}\left[\lambda X \frac{\partial U}{\partial x}\right] + \frac{\partial}{\partial y}\left[\lambda Y \frac{\partial U}{\partial y}\right] = q + P\frac{\partial U}{\partial t} \tag{8.1}$$

where λX, λY, q and P are arbitrary functions of x and y for linear problems. For nonlinear problems some or all of these coefficients may also depend upon U. The boundary conditions for most reservoir simulation problems are of the Neumann type, i.e.,

$$\frac{\partial U}{\partial n} = 0 \tag{8.2}$$

where n is the directional normal to the bounding surface. As discussed in the last chapter the discretisation of this boundary value problem results in a set of finite difference equations of the form

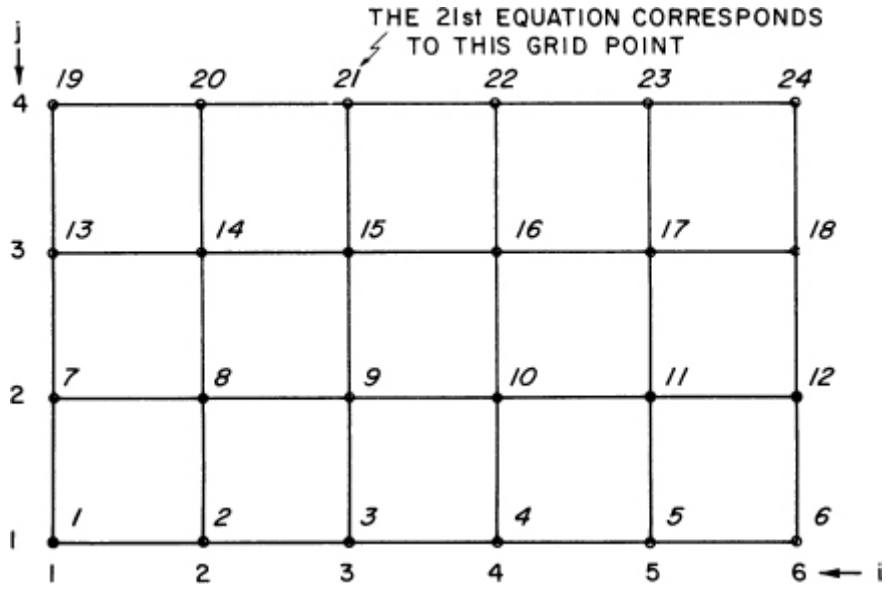


FIG. 8.1a. Standard ordering by rows (or i-index).

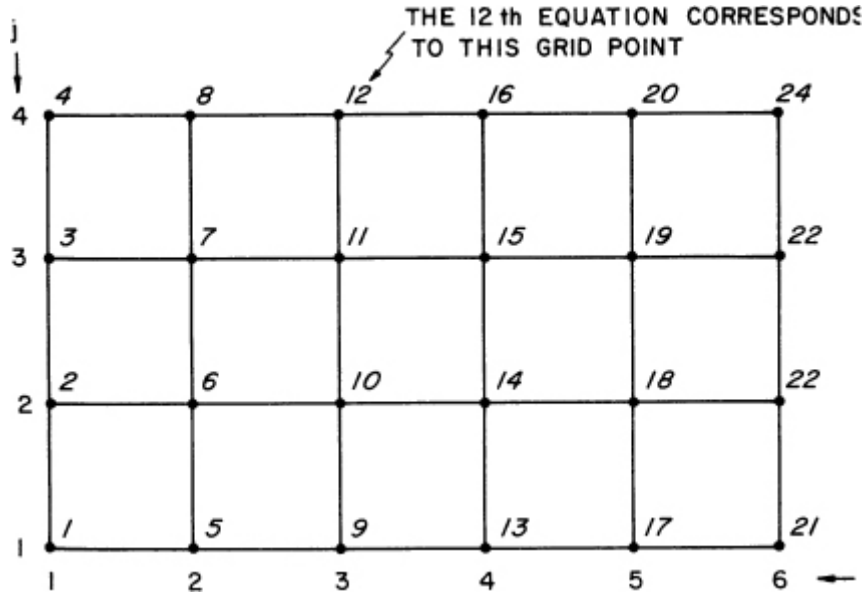


FIG. 8.1b. Standard ordering by columns (or j-index).

$$g_{ij}u^n_{i,j-1} + c_{ij}u^n_{i-1,j} + a_{ij}u^n_{ij} + b_{ij}u^n_{i+1,j} + f_{ij}u^n_{i,j+1} = d_{ij} \tag{8.3}$$

The matrix form of the difference equations depends upon the grid system and the scheme used for ordering of equations. For a uniform grid as shown in Fig. 8.1 (a, b), the coefficients are defined by

$$g_{ij} = -\left(\frac{\Delta x}{\Delta y}\right)\lambda Y_{i,j-1/2}$$

$$c_{ij} = -\left(\frac{\Delta y}{\Delta x}\right)\lambda X_{(i-1/2),j}$$

$$a_{ij} = -(g_{ij} + f_{ij} + c_{ij} + b_{ij}) + \varphi_{ij}$$

$$b_{ij} = -\left(\frac{\Delta y}{\Delta x}\right)\lambda X_{(i+1/2),j}$$

$$f_{ij} = -\left(\frac{\Delta x}{\Delta y}\right)\lambda Y_{i,j+1/2}$$

$$d_{ij} = -\Delta x\,\Delta y q_{ij} + \varphi_{ij} u_{ij}^{n-1}$$

$$\varphi_{ij} = \frac{\Delta x\,\Delta y}{\Delta t} P_{ij}$$

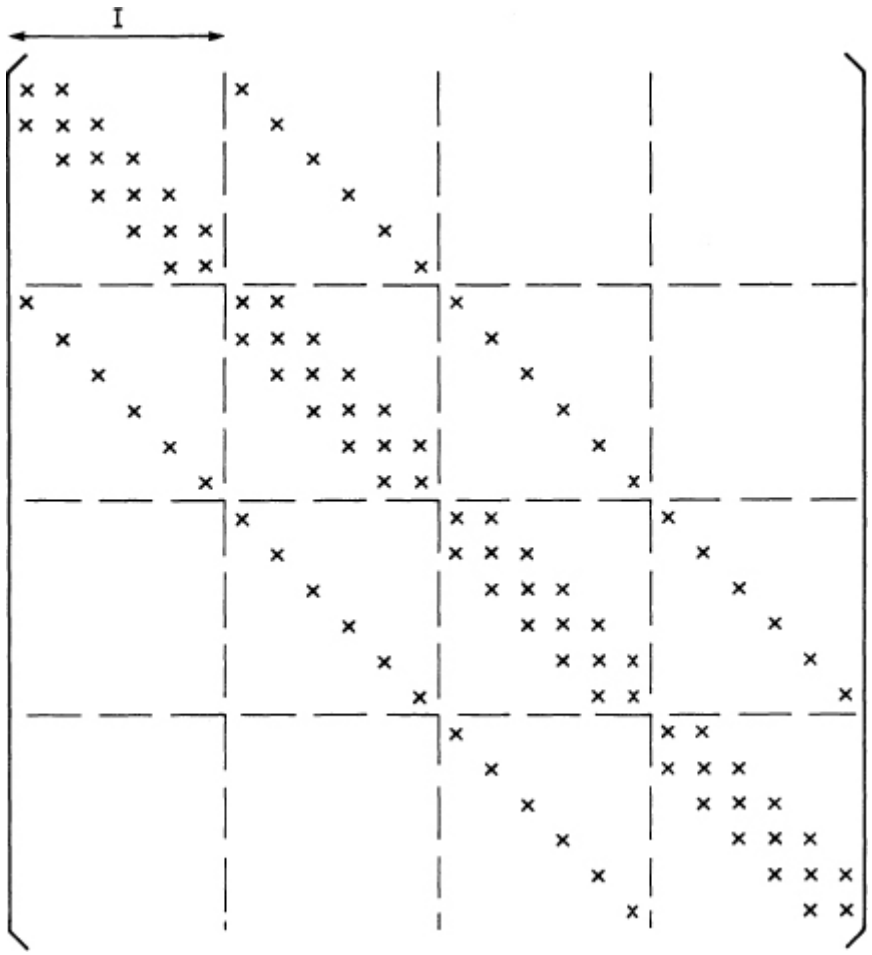


FIG. 8.2a. Form of matrix **A** for standard ordering by rows shown in Fig. 8.1a. Non-zero entries are indicated by ×. For example, these entries for the ninth row ($i = 3, j = 2$) are $g_{3,2}; c_{3,2}; a_{3,2}; b_{3,2;} f_{3,2}$.

It will be convenient for later use to also define

$$ax_{ij} = -(c_{ij} + b_{ij})$$
$$ay_{ij} = -(g_{ij} + f_{ij})$$

We note that if $\Delta x = \Delta y = h$ and $P = \lambda X = \lambda Y = 1$, the coefficients take on the following values:

$$g_{ij} = c_{ij} = b_{ij} = f_{ij} = -1$$
$$a_{ij} = 4 + \frac{h^2}{\Delta t}$$
$$\varphi_{ij} = \frac{h^2}{\Delta t}$$

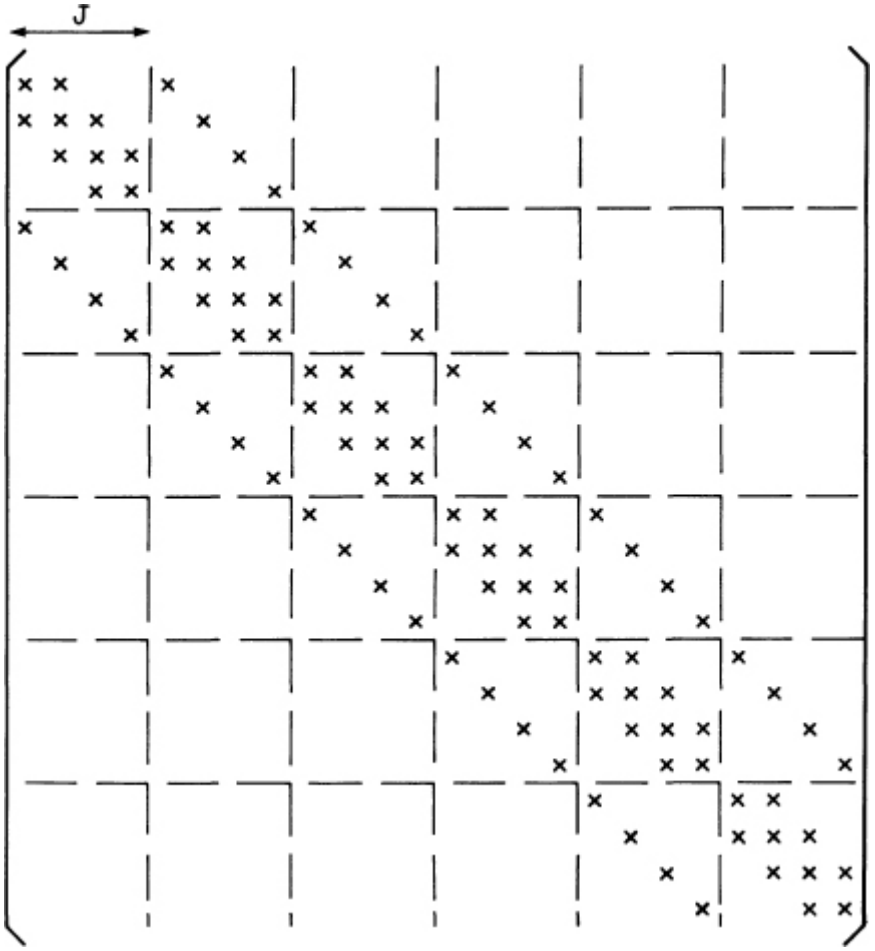


FIG. 8.2b. Form of matrix **A** for standard ordering shown by columns in Fig. 8.1b. Non-zero entries are indicated by ×. For example, these entries for the tenth row ($i = 3, j = 2$) are $c_{3,2}$; $g_{3,2}$; $a_{3,2}$; $f_{3,2}$; $b_{3,2}$.

The interpretation of coefficients in eqn. (8.3) for irregular grid has been discussed in detail in the previous chapter.

The standard (Price and Coats, 1974) or natural ordering of the unknowns is obtained if the unknowns are ordered by lines (rows or columns). These orderings are shown on Figs. 8.1a and 8.1b (numbers 1 to 24 on the grid points) and the forms of the resulting matrices are shown in Figs. 8.2a and 8.2b, respectively. The form of the two matrices shows that the ordering by columns (shorter direction) results in a matrix of smaller band-width. The matrix shown in Fig. 8.2a can be divided into a block-tridiagonal form with each block being 6 × 6 (or $I \times I$) and the number of block-rows being 4 (or J). Similarly, in Fig. 8.2b we see that the sub-matrices are 4 × 4 (or $J \times J$) and the number of block-rows is 6 (or I).

We note that all elements along a given row have the same subscript. This convenient notation was introduced by Stone (1968) and Dupont *et al*. (1968) and has been used by many investigators including Settari and Aziz (1973).

The matrix equation to be solved may be written as

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d} \tag{8.4}$$

where **A** is the band matrix shown in Figs. 8.2. For the ordering shown in Fig. 8.1b the block-tridiagonal form is:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & & & & \\ \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B}_2 & & & \\ & \mathbf{C}_3 & \mathbf{A}_3 & \mathbf{B}_3 & & \\ & & \mathbf{C}_4 & \mathbf{A}_4 & \mathbf{B}_4 & \\ & & & \mathbf{C}_5 & \mathbf{A}_5 & \mathbf{B}_5 \\ & & & & \mathbf{C}_6 & \mathbf{A}_6 \end{bmatrix} \tag{8.5}$$

where $\mathbf{A}_i$ are 4 × 4 tridiagonal matrices and $\mathbf{C}_i$ and $\mathbf{B}_i$ are 4 × 4 diagonal matrices.

The matrix version of the Thomas algorithm discussed in Chapter 4 could be applied here, but it turns out to be a relatively inefficient process for this problem, essentially equivalent to the band elimination.

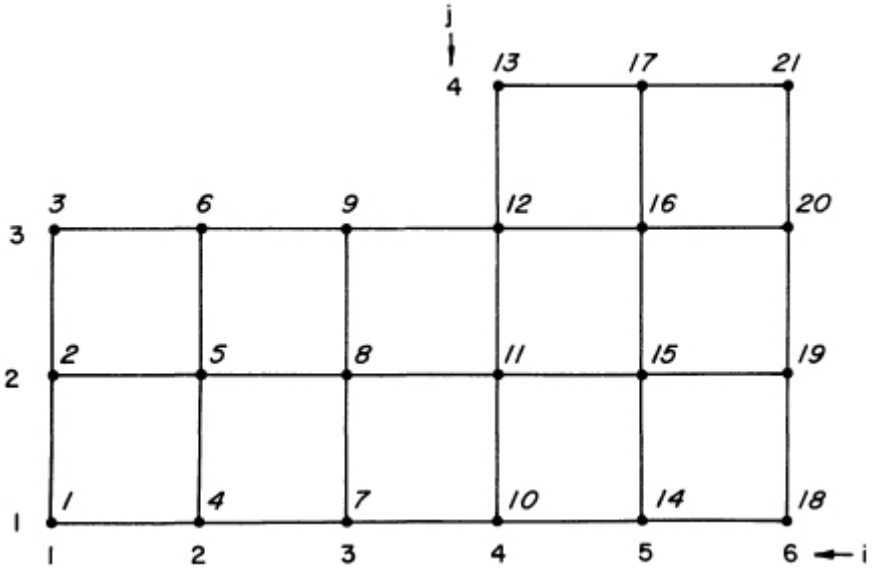


FIG. 8.3. Network consisting of more than one rectangle.

The **A** matrix of Fig. 8.2 was obtained for a network where the region is a rectangle, i.e., the number of grid points in each row is *I* and in each column is *J*. For a network of the type shown in Fig. 8.3 the resulting form of the matrix will be as shown in Fig. 8.4. The matrix for this case has a variable band-width.

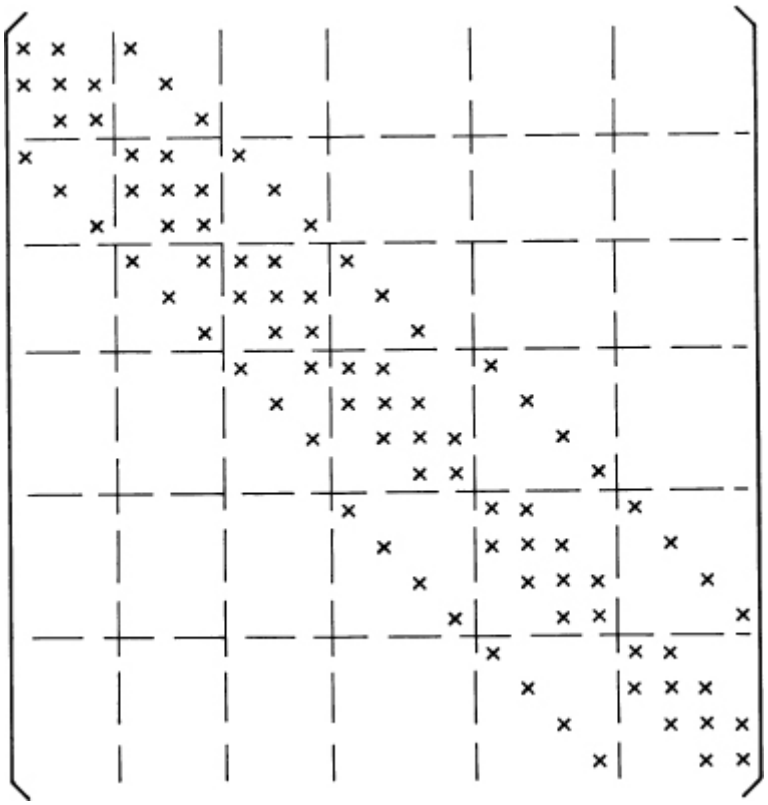


FIG. 8.4. Form of matrix **A** for the ordering shown in Fig. 8.3.

It is possible to develop very efficient methods when the matrices on the diagonal are equal to each other and matrices on the subdiagonal are also equal to each other. An example of this is the method of odd/even reduction and factorisation developed by Buzbee *et al*. (1970). However, methods of this type have found no use in the simulation of difficult reservoir problems.

In this chapter we will consider various methods for solving eqn. (8.4) for various types of matrix **A**. Towards the end of this chapter some guidelines will be provided about the selection of a suitable method for a given problem. We will restrict our discussion to methods of general applicability.

8.2 DIRECT METHODS OF SOLUTION

8.2.1 LU Factorisation

For the kind of matrices that are of interest in reservoir simulation, it is possible to factor the matrix into a lower and an upper triangular form as was done for the tridiagonal matrix in Chapter 4. This is often referred to as the **LU** factorisation (see eqn. (4.7)). The main disadvantage of this approach is that zeros within the band are treated as non-zeros and **L** + **U** is generally fuller than **A,** although it does not have a larger band-width. More precisely, the non-zero entries in **L** + **U** are confined to the *envelope* of **A**. The envelope is defined by the positions which are to the right of the first non-zero element in each row and below the first non-zero element in each column (George and Liu, 1975).

The algorithm for **LU** decomposition may be derived in the same manner as we have derived the Thomas algorithm in Chapter 4, i.e., by multiplying **LU** and equating its elements with elements of **A**. This is very cumbersome. Instead, **LU** decomposition may be viewed simply as a formalisation of the classical Gaussian elimination (Faddeev and Faddeeva, 1963) performed on the envelope of **A**. Because of its fundamental rôle, this process (often called 'band elimination') will be described in detail.

Consider a $N \times N$ matrix A with a band-width $B = 2M + 1$. This means that if $\mathbf{A} = \{a_{ij}\}$ in the standard matrix notation, then

$$a_{ij} = 0 \qquad \text{if } |j - i| > M$$

For example, for the matrix of Fig. 8.2b, $M = 4$. The elimination consists of matrix row operations to transform **A** into an upper triangular matrix.

The first step is therefore to form matrix $\mathbf{A}^{(1)}$ with zeros in the first column below the main diagonal:

$$a_{ij}^{(1)} = a_{ij} - a_{1j}(a_{i1}/a_{11}) \qquad j = 1, \ldots, M+1 \qquad (8.6)$$
$$\text{for rows} \qquad i = 2, \ldots, M+1$$
$$a_{ij}^{(1)} = a_{ij} \qquad \text{otherwise}$$

If the above process is continued for second and subsequent columns, the form of matrix $\mathbf{A}^{(k-1)}$ after the $(k-1)$th step will be as shown in Fig. 8.5 (left). Then the kth step is defined by:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{kj}^{(k-1)}[a_{ik}^{(k-1)}/a_{kk}^{(k-1)}] \qquad j = k, \ldots, \min(k+M, N) \qquad (8.7)$$
$$\text{for rows} \qquad i = k+1, \ldots, \min(k+M, N)$$
$$a_{ij}^{(k)} = a_{ij}^{(k-1)} \qquad \text{elsewhere}$$

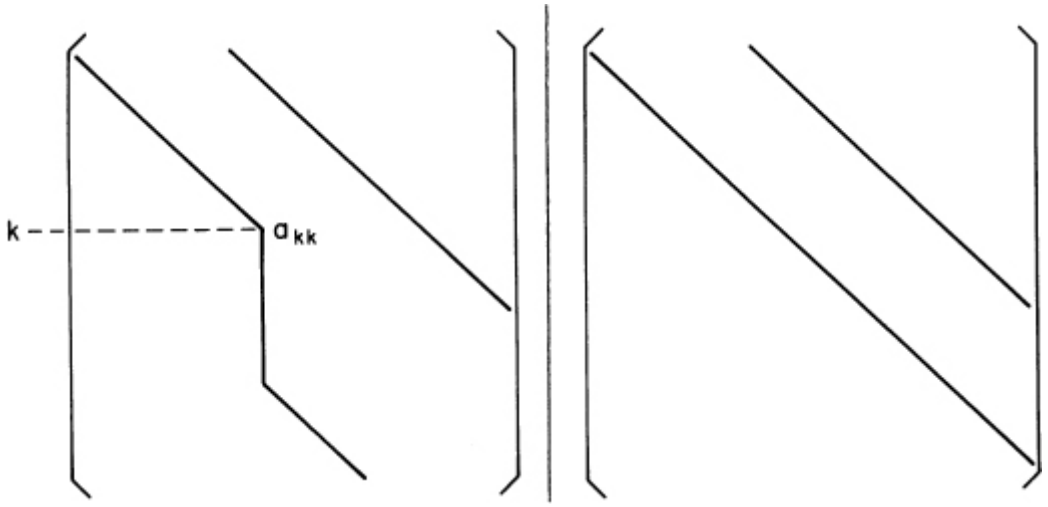


FIG. 8.5. Left, matrix $\mathbf{A}^{(k-1)}$ after $k-1$ steps; right, matrix $\mathbf{A}^{(N-1)}$ after $N-1$ steps.

After $N-1$ steps, the matrix $\mathbf{A}$ is transformed into $\mathbf{A}^{(N-1)} \equiv \mathbf{U}$. Note that we have not explicitly formed matrix $\mathbf{L}$. Instead, by performing the same manipulations on the right-hand side vector $\boldsymbol{d}$, we actually compute $\boldsymbol{g} = \mathbf{L}^{-1}\mathbf{d}$. This is done by computing a sequence of vectors $\boldsymbol{d}^{(k)}$ together with $\mathbf{A}^{(k)}$ as

$$
\begin{aligned}
d_i^{(1)} &= d_i - d_1(a_{i1}/a_{11}) \qquad \text{for} \quad i = 2, \ldots, M+1 \\
d_i^{(1)} &= d_i \qquad \text{elsewhere} \\
&\vdots \\
d_i^{(k)} &= d_i^{(k-1)} - d_k^{(k-1)}[a_{ik}^{(k-1)}/a_{kk}^{(k-1)}] \\
&\qquad\qquad \text{for} \quad i = k+1, \ldots, \min(k+M, N) \\
d_i^{(k)} &= d_i^{(k-1)} \qquad \text{elsewhere}
\end{aligned}
\tag{8.8}
$$

Then $\boldsymbol{d}^{(N-1)} = \boldsymbol{g}$ and the final solution is obtained by solving

$$
\mathbf{U}\boldsymbol{u} = \boldsymbol{g} \tag{8.9}
$$

The work requirement in terms of number of multiplications is

$$
\begin{aligned}
W = (N - 2M + 1)[(M+1)^2 + M] + \frac{M(M-1)(2M-1)}{3} \\
+ M(M-1) + (M+2)^2 - M - 9
\end{aligned}
\tag{8.10}
$$

as derived in Exercise 8.1. Numerous variants of this basic process exist. Only a few that are of practical importance are described briefly:

(a) *Repeated solutions with a constant matrix*. If it is desired to solve several systems $\mathbf{A}\boldsymbol{u} = \boldsymbol{d}$ where only $\boldsymbol{d}$ is different, the forward elimination need be performed only once, provided the lower band is saved. If at the stage k the elements of the lower band in column k are not replaced by zeros then these are precisely the elements required for the operations defined by eqn. (8.8). This is accomplished if eqn. (8.7) is carried out starting with $j = k + 1$. Solutions for different $\boldsymbol{d}$ are then obtained by repeated application of eqns. (8.8) and (8.9).

(b) *Pivoting*. Obviously, the process requires that the diagonal element $a_{kk}^{(k-1)}$ is non-zero. To minimise round-off errors, this element should be as large as possible. Pivoting means that at every stage of elimination we search for the largest element among $a_{kk}^{(k-1)}$, $a_{k+1,k}^{(k-1)}$, ... and then interchange the row with the largest element with row k to maximise the diagonal element. Although pivoting may be necessary for ill-conditioned matrices, it is usually not required for matrices generated in reservoir simulation. For theoretical details related to round-off errors, see Wilkinson (1963).

(c) Accuracy of the solution may be increased by calculating the residual vector $\boldsymbol{R} = \mathbf{A}\boldsymbol{u} - \boldsymbol{d}$ and solving for a correction $\boldsymbol{A}\ \delta\boldsymbol{u} = \boldsymbol{R}$ as described in (a). This is rarely necessary.

(d) It is possible to carry out the process described above by columns rather than by rows, or reduce the matrix to a diagonal instead of an upper triangular form. Such methods may result in savings of storage in some cases.

(e) For a symmetric matrix one can use a different form of the elimination algorithm to take advantage of the symmetry. For large matrices the work required approaches one-half the work required by the standard algorithm (see Section 4.2.3 of Chapter 4 and Exercise 8.2).

Many algorithms for band elimination are available in the literature. Thurnau (1963) has published one such algorithm with column pivoting. It has, however, been found (Woo *et al*., 1973) that column or row pivoting is unnecessary for reservoir simulation problems. In Appendix B a program 'GBAND' for the solution of band equations without pivoting is presented. This program is based on an algorithm developed by Gruska and Poliak (1967). Algorithms for variable band-width symmetric matrices have been published by Jennings (1971) and Wilson (1975). Other algorithms may be found in Schwarz (1968) and Wilkinson and Reinsch (1971).

8.2.2 Ordering of Equations

As mentioned earlier, the form of the matrix depends upon the ordering of unknowns. In this section we present a brief account of some common ordering schemes. Several ordering schemes have been discussed by Woo *et al*. (1973), Price and Coats (1974), and McDonald and Trimble (1977). Some of these schemes are presented in Figs. 8.6 to 8.8 along with the resulting coefficient matrices. In the next section we will show how one can take advantage of the ordering to save computer time and storage.

8.2.3 Sparse Matrix Techniques

We have already seen that matrices arising in reservoir simulation problems are such that most of the elements are zeros. Such matrices are known as *sparse matrices* and they arise in a broad spectrum of application areas (Rose and Willoughby, 1972). In this section some techniques are considered that take advantage of the sparse character of the matrix. There are two steps involved in the application of sparse matrix techniques (Woo *et al*., 1973):

1. Order equations to reduce the size of the envelope of **A**.
2. Solve the resulting equations by using an efficient Gaussian elimination or factorisation algorithm that avoids operating on zeros.

We have seen in the previous section how the envelope of **A** changes with the ordering of equations. Several strategies for reducing the envelope of matrices have

been reviewed by Cuthill (1972), George (1973) and Price and Coats (1974). For a symmetric positive definite $(n + 1) \times (n + 1)$ matrix, **A**, George has shown that with standard symmetric factorisation and standard ordering $O(n^4)$ total arithmetic operations are required, and the storage required is $O(n^3)$. By re-ordering equations, George shows that the same standard factorisation can be accomplished in $O(n^3)$ operations with only $O(n^2 \log_2 n)$ storage locations. This rather startling result provides considerable motivation for ordering matrix equations to reduce work and storage.

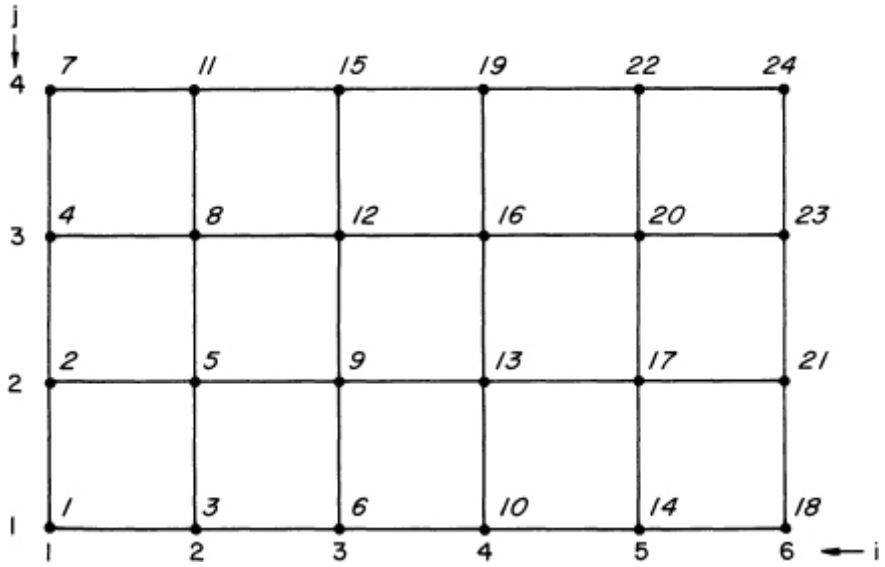


FIG. 8.6a. Ordering D2.

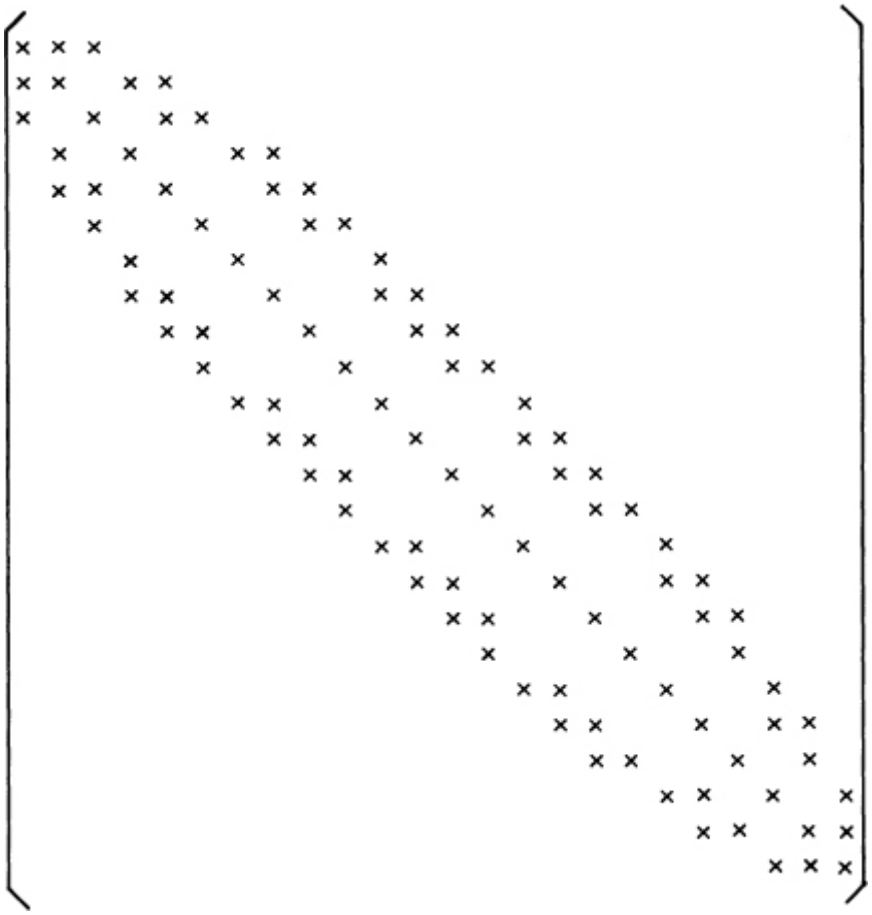


FIG. 8.6b. Matrix **A** for ordering D2.

The matrix **A** being considered here is generally diagonally dominant and asymmetric with a symmetric incidence matrix **M** (Todd, 1962). The incidence matrix **M** is defined by

$$m_{ij} = 1 \qquad \text{if} \quad a_{ij} \neq 0$$

$$m_{ij} = 0 \qquad \text{if} \quad a_{ij} = 0$$

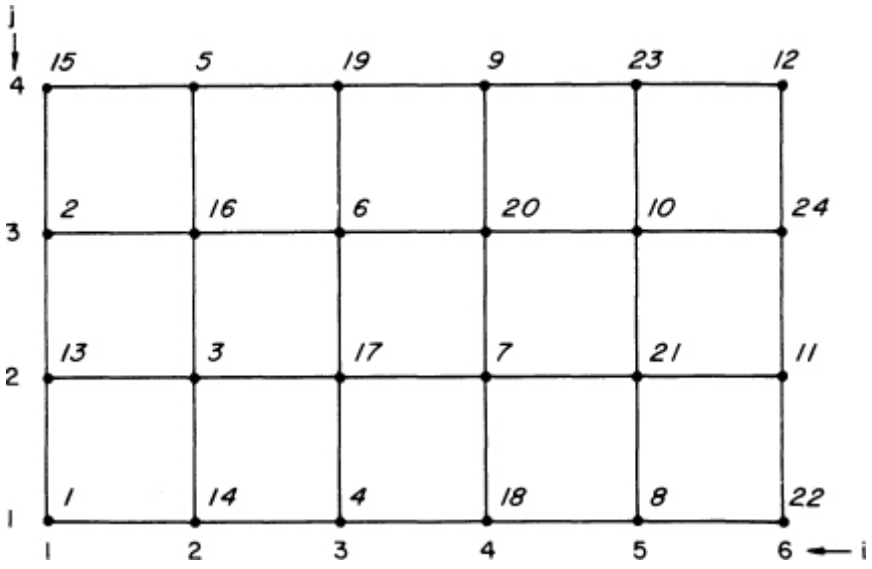


FIG. 8.7a. Ordering D4.

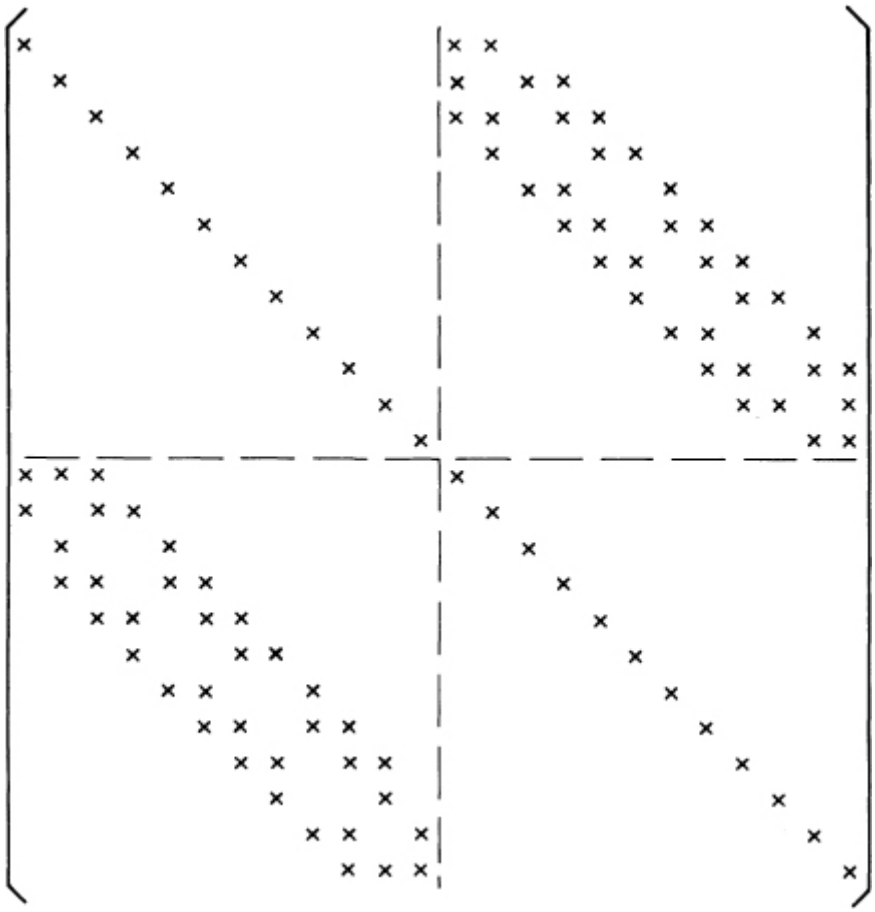


FIG. 8.7b. Matrix **A** for ordering D4.

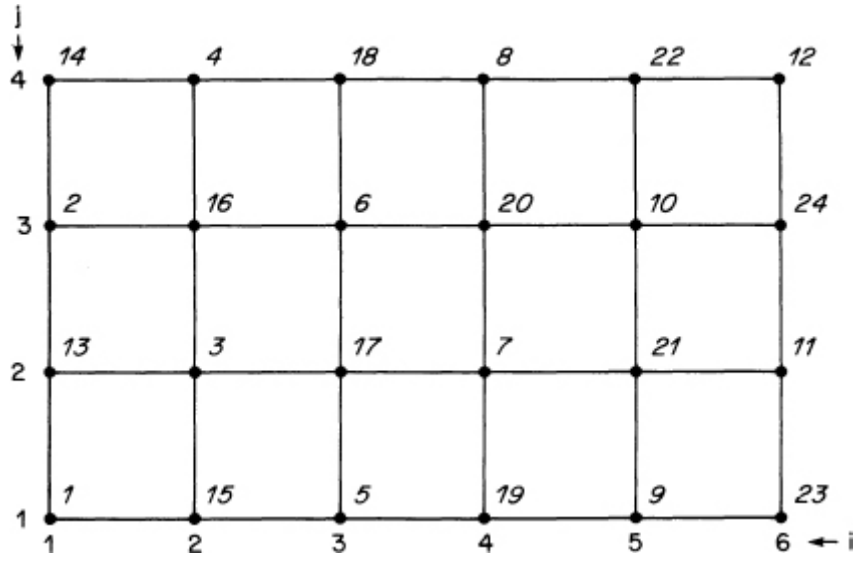


FIG. 8.8a. Cyclic-2 ordering.

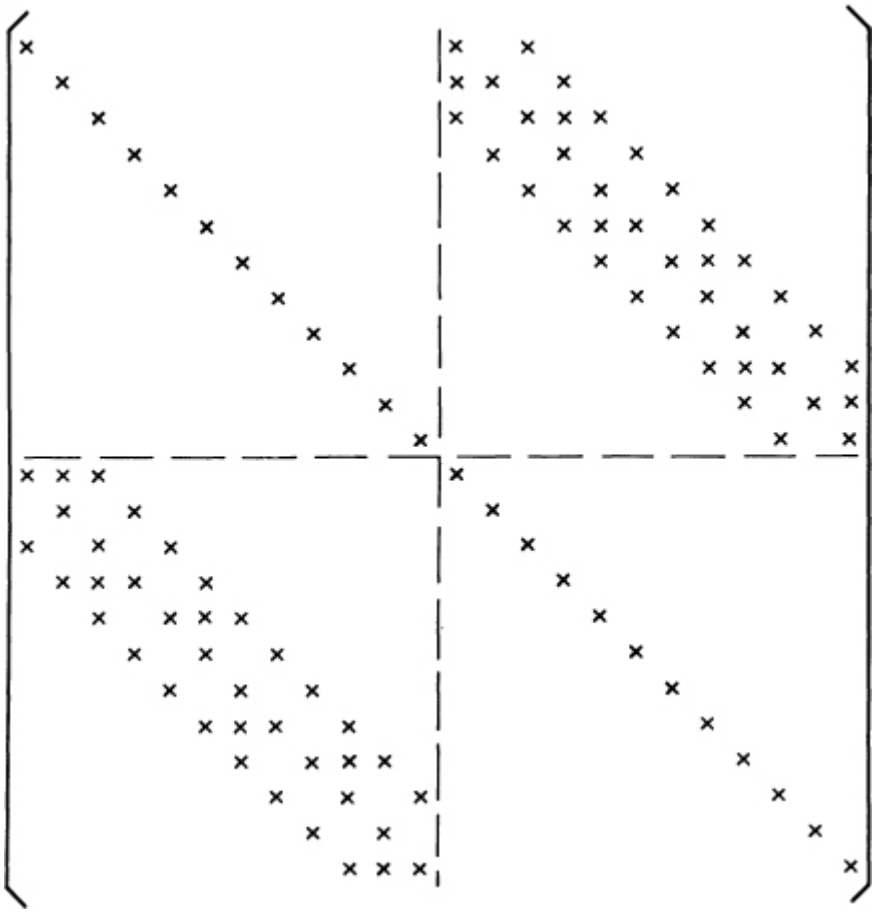


FIG. 8.8b. Matrix **A** for Cyclic-2 ordering.

Price and Coats (1974) have shown that for matrices of this type the work required by Gaussian elimination is

$$W = \sum_{i=1}^{N} [(w_i + 1)^2 + w_i] \tag{8.11}$$

and the corresponding minimum storage requirement is

$$S = \sum_{i=1}^{N} w_i \tag{8.12}$$

where N is the total number of unknowns and w_i is the number of non-zero entries in the ith row to the right of the diagonal, at the ith stage of elimination. The ordering schemes are selected to reduce work by reducing $\sum_{i=1}^{N} w_i^2$. Such schemes are called *sparsity-conserving ordering schemes* and they can be divided into two classes: (1) *matrix-banding schemes*, and (2) *pseudo-optimal ordering schemes.*

Matrix-banding schemes. Schemes in this class yield a matrix with non-zero entries restricted to a relatively narrow band about one of the diagonals (upper left to lower right, or upper right to lower left). In this section we will summarise the results presented by Price and Coats (1974). For a two-dimensional problem we can rewrite eqn. (8.10) for the work for standard Gaussian elimination in terms of the total number of grid points in the x-direction (I) and the total number of grid points in the y-direction (J). Since $N = IJ$ and $M = J$ the following approximation holds for large I and J ($J < I$):

$$W_1 \simeq IJ^3 \tag{8.13}$$

The corresponding storage requirement is

$$S_1 = IJ^2 \tag{8.14}$$

For the D2 ordering shown in Fig. 8.6 and for large I and J the work and storage requirements are

$$W_2 = IJ^3 - \frac{J^4}{2} \tag{8.15}$$

and

$$S_2 = IJ^2 - \frac{J^3}{3} \tag{8.16}$$

For $I = J$ this scheme requires one-half the work and two-thirds the storage as compared to the standard ordering.

For the D4 ordering of Fig. 8.7 the estimates for large I and J are

$$W_3 = \frac{IJ^3}{2} - \frac{J^4}{4} \tag{8.17}$$

$$S_3 = \frac{IJ^2}{2} - \frac{J^3}{6} \tag{8.18}$$

Therefore, for $I = J$ this scheme requires one-quarter the work and one-third the storage of standard ordering.

The estimates for the cyclic-2 ordering of Fig. 8.8 are

$$W_4 = \frac{IJ^3}{2} \tag{8.19}$$

$$S_4 = \frac{IJ^2}{2} \tag{8.20}$$

It is clear that scheme D4 provides the greatest advantage. Price and Coats have computed the ratio W_3/W_1 for various values of I and J (without the assumption of large I, J). Their results are the basis for Fig. 8.9, which shows the possible improvement using D4 ordering as a function of I and the ratio I / J. Note that the improvement is largest when $J = I$, i.e., for square grid.

Having summarised the results, we will now show in detail the implementation of D4 ordering. This will also help to clarify the basic idea behind all ordering methods.

Starting with the matrix on Fig. 8.7b, we observe that the matrix equation can be partitioned as

$$\mathbf{A}\boldsymbol{u} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{bmatrix} \tag{8.21}$$

where $\mathbf{A_1}$ and $\mathbf{A_4}$ are diagonal, and $\mathbf{A_2}$ and $\mathbf{A_3}$ are sparse matrices. As a first step, perform the forward elimination on the lower half of $\mathbf{A}$, which will transform $\mathbf{A_3}$ into a null matrix, i.e.,

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \bar{\mathbf{A}}_4 \end{bmatrix} \qquad \bar{\boldsymbol{d}} = \begin{bmatrix} \boldsymbol{d}_1 \\ \bar{\boldsymbol{d}}_2 \end{bmatrix}$$

Note that $\mathbf{A}_1$, $\mathbf{A}_2$ and $\boldsymbol{d}_1$ have not changed, while $\mathbf{A}_4$ has changed into a band

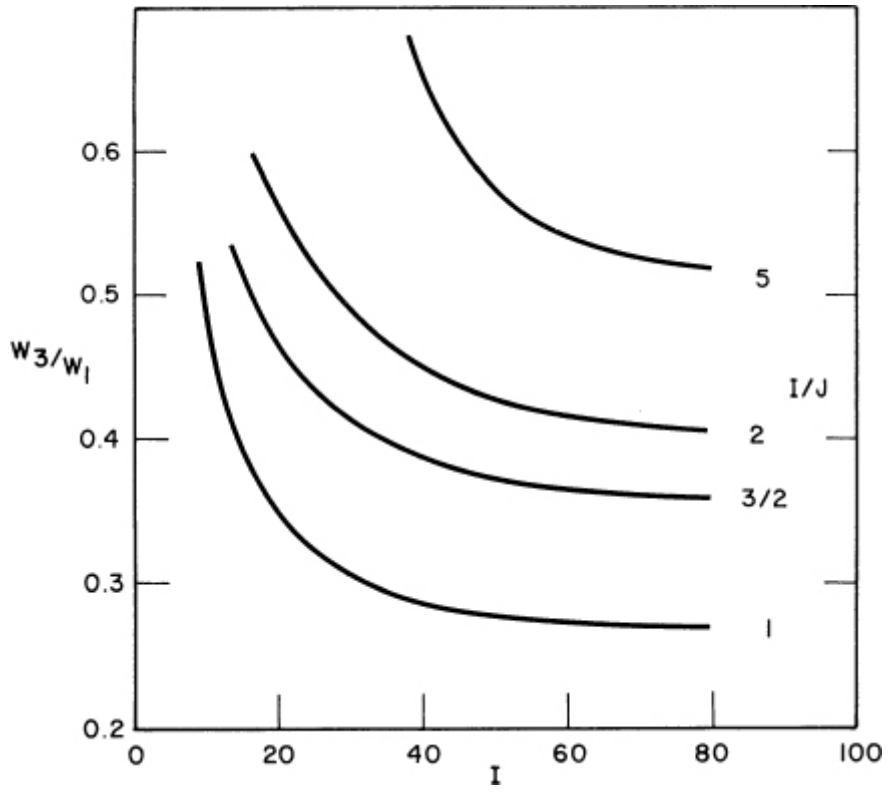


FIG. 8.9. Comparison of work by D4 and standard ordering.

matrix $\bar{\mathbf{A}}_4$. For our example, the structure of matrix $\bar{\mathbf{A}}$ is shown in Fig. 8.10, where the entries resulting from fill-up are denoted by circles. It is important to note that the maximum band-width of the matrix $\bar{\mathbf{A}}_4$ is the same as for the original matrix $\mathbf{A}$. We can now solve the equations for the lower half

$$\bar{\mathbf{A}}_4 \boldsymbol{u}_2 = \bar{\boldsymbol{d}}_2 \tag{8.22}$$

independently by a standard band elimination. After $\boldsymbol{u}_2$ has been obtained, $\boldsymbol{u}_1$ can be computed easily by the back substitution

$$\boldsymbol{u}_1 = \mathbf{A}_1^{-1}\boldsymbol{d}_1 - \mathbf{A}_1^{-1}\mathbf{A}_2\boldsymbol{u}_2 \tag{8.23}$$

since $\mathbf{A_1}$ is diagonal. Thus, we have reduced the size of the matrix problem by suitable ordering in half. It is easily seen from eqn. (8.13) for band elimination that the work for half the unknowns will be reduced by a factor of two for a constant band-width matrix. Although not apparent from our example, matrix $\bar{\mathbf{A}}_4$ will have a variable band-width for large I and J with the maximum equal to the band-width of the original matrix. This causes a further reduction in the work and this reduction is largest for $I = J$. For large $I = J$ this phenomenon gives another factor of two in the reduction of

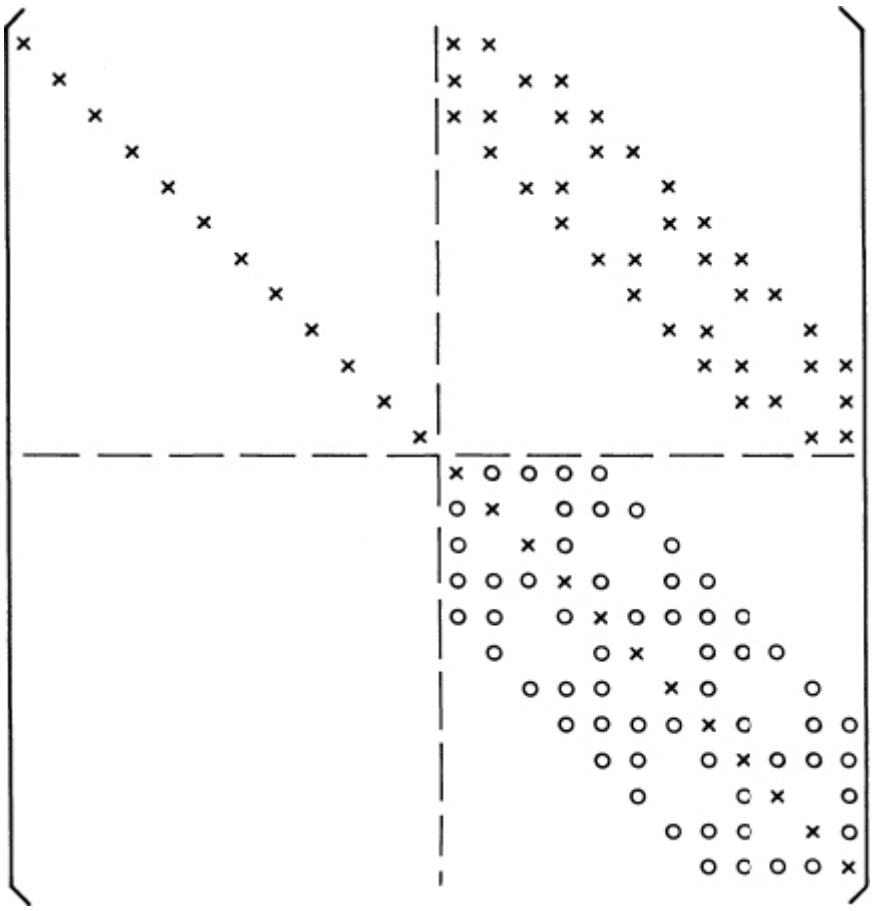


FIG. 8.10. Transformed matrix for D4 ordering.

work; the overall work ratio then tends to $^1/_4$. The storage savings are deduced in the same manner.

Pseudo-optimal ordering schemes. Price and Coats (1974) have considered three schemes described by Tinney and Walker (1967) for optimum conservation of matrix sparsity in Gaussian elimination. Although these schemes are considered to be more efficient than matrix banding schemes for electrical network problems, they are not necessarily superior for algebraic equations resulting from partial differential

equations. Price and Coats have shown that although substantial savings in work over D4 are possible for $I = J$, the same is not true when I is $2J$ or $3J$. These schemes present considerable programming difficulties.

Schemes of this type require further investigation and will not be considered in this book. Some publications are: Brayton *et al*. (1970), Gustavon and Hachtel (1973), Woo *et al*. (1973) and Gustavson (1975).

8.3 ITERATIVE METHODS

It is often more economical in terms of computer work and storage to do a problem by some iterative rather than the direct elimination technique. In this section we will provide the basic concepts of iterative methods and some selected methods. Several excellent books are available on this subject and for this reason the subject is not treated in detail. The books by Varga (1962), Wachspress (1966) and Young (1971) will be of particular interest to the reader.

The methods to be considered may be divided into two broad categories: (1) *point iterative*, and (2) *block iterative*. Point-iterative methods involve no matrix computations and their application to reservoir simulation problems is restricted to relatively simple cases. Block-iterative methods involve simultaneous solution of a block of equations that are easy to solve. Although only the block-iterative methods require matrix computations, both classes of methods must be written in matrix form for analysis. Some basic results from the theory of iterative methods will be presented before we discuss actual methods.

Let us consider a matrix equation of the form

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d} \tag{8.24}$$

where the coefficient matrix **A** may assume the form of eqn. (8.5). Any iterative scheme for this problem may be expressed as

$$\boldsymbol{u}^{(v+1)} = \mathbf{B}\boldsymbol{u}^{(v)} + \boldsymbol{b} \tag{8.25}$$

where the matrix **B** and the vector $\boldsymbol{b}$ depend upon the problem being solved and the method of iteration. If **B** is independent of v then the iterative scheme is called *stationary.*

An iteration scheme is convergent if

$$\lim_{v \to \infty} \boldsymbol{u}^{(v)} = \boldsymbol{u} \tag{8.26}$$

where $\boldsymbol{u}$ is the exact solution of eqn. (8.24). Obviously it is necessary (but not sufficient!) for convergence of a scheme that it satisfies the condition

$$\boldsymbol{u} = \mathbf{B}\boldsymbol{u} + \boldsymbol{b} \tag{8.27}$$

Now if we define the error at various stages of iteration as

$$\boldsymbol{\varepsilon}^{(\nu)} = \boldsymbol{u}^{(\nu)} - \boldsymbol{u} \tag{8.28}$$

then by subtracting eqn. (8.27) from (8.25) and substituting eqn. (8.28) we see that

$$\boldsymbol{\varepsilon}^{(\nu+1)} = \mathbf{B}\boldsymbol{\varepsilon}^{(\nu)} \tag{8.29}$$

or, in terms of the initial error vector $\boldsymbol{\varepsilon}^{(0)}$,

$$\boldsymbol{\varepsilon}^{(\nu)} = \mathbf{B}^{\nu}\boldsymbol{\varepsilon}^{(0)} \tag{8.30}$$

In order that eqn. (8.25) be convergent $\boldsymbol{\varepsilon}^{(\nu)}$ must approach *0* for arbitrary $\boldsymbol{\varepsilon}^{(0)}$. This implies that $\|\mathbf{B}^{\nu}\|$ must approach zero (cf. Appendix A).

Theorem: The iteration scheme

$$\boldsymbol{u}^{(\nu+1)} = \mathbf{B}\boldsymbol{u}^{(\nu)} + \boldsymbol{b} \tag{8.31}$$

converges if and only if

$$\rho(\mathbf{B}) < 1$$

The rate of convergence is also related to $\rho(\mathbf{B})$. The ratio of the norm of the error at (ν) level to that at (0) level is obtained from eqn. (8.30).

$$\frac{\|\boldsymbol{\varepsilon}^{(\nu)}\|}{\|\boldsymbol{\varepsilon}^{(0)}\|} \leq \|\mathbf{B}^{\nu}\| \tag{8.32}$$

and $\|\mathbf{B}^{\nu}\|$ serves as a comparison of different iterative methods.

Definition (*Varga*, 1962, *p*. 62): *Let* $\mathbf{B_1}$ *and* $\mathbf{B_2}$ *be two matrices; then if for some positive integer* ν, $\|\mathbf{B}_1^{\nu}\| < 1$, *then*

$$R(\mathbf{B}_1^{\nu}) \equiv -\ln\left[(\|\mathbf{B}_1^{\nu}\|)^{1/\nu}\right] = \frac{-\ln\|\mathbf{B}_1^{\nu}\|}{\nu}$$

is the average rate of convergence for v iterations of the matrix $\mathbf{B_1}$. *If*

$$R(\mathbf{B}_1^v) < R(\mathbf{B}_2^v)$$

then $\mathbf{B_2}$ *is iteratively faster than* $\mathbf{B_1}$ *for v iterations.*

Another useful result is stated in the following theorem:

Theorem (*Varga*, 1962, *p*. 67):
For a convergent matrix **B**

$$\lim_{v \to \infty} R(\mathbf{B}^v) = -\ln \rho(\mathbf{B}) = R_\infty(\mathbf{B})$$

where $R_\infty(\mathbf{B})$ *is the asymptotic rate of convergence.*
A corollary to the above result is the following:

For any positive integer v for which $\|\mathbf{B}^v\| < 1$

$$R_\infty(\mathbf{B}) \geq R(\mathbf{B}^v)$$

8.3.1 Point Jacobi Method

The *i*th equation in (8.3) may be re-arranged in the form

$$u_{ij}^n = -\frac{1}{a_{ij}}[g_{ij}u_{i-1,j}^n + c_{ij}u_{i,j-1}^n + b_{ij}u_{i,j+1}^n + f_{ij}u_{i+1,j}^n + d_{ij}] \tag{8.33}$$

where the known terms at the old level of time and the source terms have been combined into d_{ij}. In the matrix form of these equations $\boldsymbol{d}$ also includes known boundary values of $\boldsymbol{u}$. Since all of the u terms in eqn. (8.33) are at the time level n we will not write this superscript in most of the discussion to follow. Since we do not know all of the values of u on the right side of eqn. (8.33) it is natural to try the following iteration scheme:

$$u_{ij}^{(v+1)} = -\frac{1}{a_{ij}}[g_{ij}u_{i-1,j}^{(v)} + c_{ij}u_{i,j-1}^{(v)} + b_{ij}u_{i,j+1}^{(v)} + f_{ij}u_{i+1,j}^{(v)} - d_{ij}] \tag{8.34}$$

$$v = 0, 1, 2, \ldots$$

This is the point Jacobi method.

The iteration is started ($v = 0$) with some assumed value of $\boldsymbol{u}^{(0)}$ and continued until

$$\|\boldsymbol{u}^{(v+1)} - \boldsymbol{u}^{(v)}\| < e \tag{8.35}$$

where e is some acceptable tolerance. A different convergence criterion may be defined as

$$\frac{\|\boldsymbol{u}^{(v+1)} - \boldsymbol{u}^{(v)}\|}{\|\boldsymbol{u}^{v}\|} < e \tag{8.36}$$

Unfortunately the above simple criteria can be misleading, since they give no indication of how well the computed solution actually satisfies the original equation. If $\rho(\mathbf{B})$ is only slightly less than unity then the change in $\boldsymbol{u}^{(v)}$ between two iterations will be very small even though $\boldsymbol{u}^{(v)}$ is far from $\boldsymbol{u}$. Another approach is to look at the residual vector and see if it satisfies

$$\|\mathbf{A}\boldsymbol{u}^{(v)} - \boldsymbol{d}\| < e \tag{8.37}$$

This criterion can also fail if the matrix is ill-conditioned (Goult *et al*. 1974, 101). Computation of the residual in double precision can improve the test.

For transient problems a convenient value of the starting vector is obtained from

$$\boldsymbol{u}^{(o)} = \boldsymbol{u}^{n-1}$$

The iteration matrix **B** for the point Jacobi method may be derived by writing

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{H}$$

where **D** is diagonal matrix containing the diagonal elements of **A. L** and **H** are strictly lower and upper triangular matrices. We can now write eqn. (8.24) as

$$(\mathbf{D} - \mathbf{L} - \mathbf{H})\boldsymbol{u}^{n} = \boldsymbol{d} \tag{8.38}$$

and eqn. (8.33) as

$$\boldsymbol{u}^{n} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{H})\boldsymbol{u}^{n} + \mathbf{D}^{-1}\boldsymbol{d} \tag{8.39}$$

The point Jacobi method in matrix form is

$$\boldsymbol{u}^{(v+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{H})\boldsymbol{u}^{(v)} + \mathbf{D}^{-1}\boldsymbol{d} \tag{8.40}$$

or

$$\boldsymbol{u}^{(\nu+1)} = \mathbf{B}_J \boldsymbol{u}^{(\nu)} + \boldsymbol{b} \tag{8.41}$$

where $\mathbf{B_J} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{H})$. The eigenvalues of this matrix may be expressed in terms of the elements of **D, H** and **L** for some simple cases to see if $\rho(\mathbf{B_J}) < 1$ (Exercise 8.3).

8.3.2 Point Gauss-Seidel Method

This method differs from the point Jacobi method only in the fact that the *latest available values* of u are used on the right-hand side of eqn. (8.34). For example, if we use natural ordering, $u_{i-1,j}^{(\nu+1)}$ and $u_{i,j-1}^{(\nu+1)}$ are known when we are solving for $u_{i,j}^{(\nu+1)}$. Hence we can write

$$u_{ij}^{(\nu+1)} = -\frac{1}{a_{ij}}\left[g_{ij}u_{i-1,j}^{(\nu+1)} + c_{ij}u_{i,j-1}^{(\nu+1)} + b_{ij}u_{i,j+1}^{(\nu)} + f_{ij}u_{i+1,j}^{(\nu)} - d_{ij}\right] \qquad \nu = 0, 1, 2, \ldots \tag{8.42}$$

to define the point Gauss–Seidel method. The method is easier to program than the point Jacobi method and its rate of convergence is also higher. The iteration process is started and terminated in exactly the same manner as for the point Jacobi method.

The equation may be written in the form of eqn. (8.25)

$$\boldsymbol{u}^{(\nu+1)} = \mathbf{B}_{GS}\boldsymbol{u}^{(\nu)} + \boldsymbol{b} \tag{8.43}$$

where $\mathbf{B_{GS}} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{H}$. If $0 < \rho(\mathbf{B_J}) < 1$, then $R_\infty(\mathbf{B_{GS}}) > R_\infty(\mathbf{B_J})$ (Stein–Rosenberg Theorem; Varga, 1962, p. 70).

8.3.3 Point Successive Over Relaxation (SOR) Method

The Gauss–Seidel method eqn. (8.42) makes no use of the value $u_{i,j}^{(\nu)}$ in computing $u_{i,j}^{(\nu+1)}$. If we designate $u_{i,j}^{(\nu+1)}$ computed by eqn. (8.42) as $u_{i,j}^{*(\nu+1)}$ then a possible improvement in the iteration process may be obtained by weighting:

$$u_{ij}^{(\nu+1)} = (1 - \omega)u_{ij}^{(\nu)} + \omega u_{ij}^{*(\nu+1)} \tag{8.44}$$

Now we may ask: 'Is it possible to find a weighting factor ω so that the rate of convergence of this method is better than that of the point Gauss–Seidel method?' In order to answer this question, eqn. (8.44) must be written in the form of eqn. (8.25)

$$\boldsymbol{u}^{(\nu+1)} = \mathbf{B}_{SOR}\boldsymbol{u}^{(\nu)} + \boldsymbol{b} \tag{8.45}$$

where

$$\mathbf{B_{SOR}} = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega\mathbf{H}] \tag{8.46}$$

The rate of convergence of the SOR method depends upon the value of ω and the ordering of unknowns. All of the orderings considered in this chapter are called *consistent* (for a detailed discussion of this property see Varga, 1962 and Young, 1971). For a consistent ordering the optimum value of ω, denoted as ω_b, is given by

$$\omega_b = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B_J})}} = 1 + \left[\frac{\rho(\mathbf{B_J})}{1 + \sqrt{1 - \rho^2(\mathbf{B_J})}}\right]^2 \tag{8.47}$$

Also

$$\rho(\mathbf{B_{GS}}) = \rho^2(\mathbf{B_J}) \tag{8.48}$$

Varga (1962, p. 112) provides the following estimate of the asymptotic rate of convergence

$$R_\infty \simeq \sqrt{8}[R_\infty(\mathrm{B_J})]^{1/2} \tag{8.49}$$

for 2-cyclic matrices which occur naturally in many reservoir simulation problems (see Fig. 8.7b).

The improvement in the rate of convergence compared to the Jacobi and Gauss–Seidel methods becomes dramatic when $\rho(\mathbf{B_J})$ is close to 1. Consider, for example, a case when $\rho(\mathbf{B_J}) = 0{\bullet}9999$. Then

$$\begin{aligned} R_\infty(\mathrm{B_J}) &\simeq 0 \cdot 0001 \\ R_\infty(\mathrm{B_{GS}}) &\simeq 0 \cdot 0002 \\ R_\infty(\mathrm{B_{SOR}}) &\simeq 0 \cdot 0283 \quad (\text{with } \omega_b) \end{aligned}$$

The above results indicate that the Gauss–Seidel method will require approximately 100 times the number of iterations required by the SOR method for comparable accuracy. From a practical point it is often difficult to find the optimum value of ω. Figure 8.11 shows how $\rho(\mathbf{B_{SOR}})$ is affected by the value of ω. This figure shows that over-estimation of ω by a small amount is not as bad as the under-estimation of ω by a similar amount.

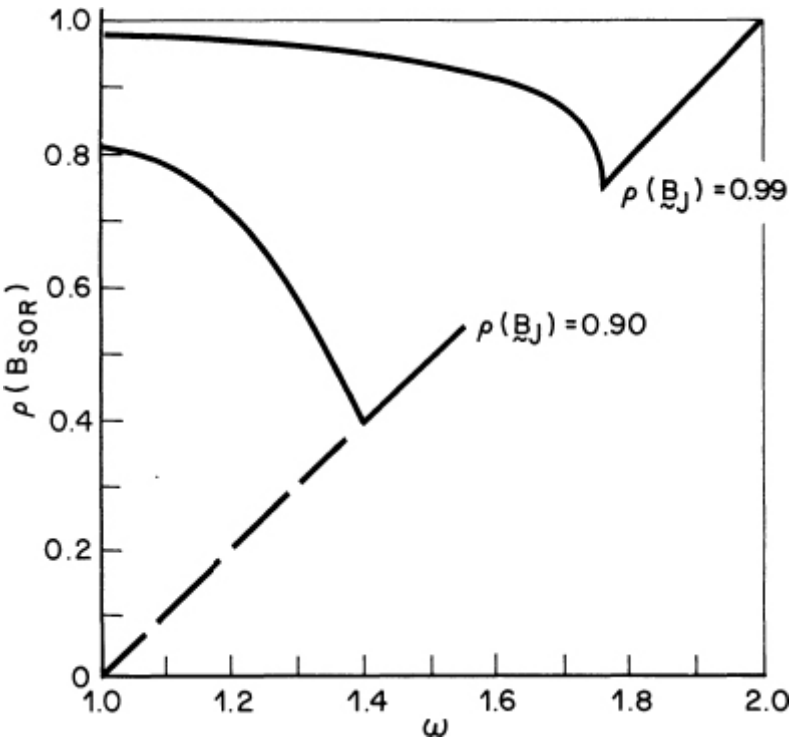


FIG. 8.11. Spectral radius of $\mathbf{B}_{\mathbf{SOR}}$.

Approximate value of optimum ω may be obtained from some of the methods discussed in the books by Varga (1962), Wachspress (1966) and Young (1971); and papers by Carre (1961), Reid (1966), and Hageman and Kellogg (1968).

One of the most common methods used for estimating ω_b is the *power method*. In this method iterations are performed with $\omega = 1$ (Gauss–Seidel) and the ratio of the corresponding elements of $\boldsymbol{\delta}^{(\nu+1)}$ and $\boldsymbol{\delta}^{(\nu)}$ where

$$\boldsymbol{\delta}^{\nu} = \boldsymbol{u}^{(\nu+1)} - \boldsymbol{u}^{(\nu)} \tag{8.50}$$

is taken. For large numbers of iterations this ratio approaches $\rho(\mathbf{B}_{\mathbf{GS}})$ and it can be used in eqn. (8.47) to estimate ω_b. One particularly convenient variation of the power method involves iterations with a sub-optimal ω to find the ω_b. This method is outlined below (Young, 1971):

Let $\boldsymbol{\delta}^{(\nu)}$ be defined by eqn. (8.50) and

$$\theta^{(\nu)} = \|\boldsymbol{\delta}^{(\nu+1)}\|_\infty / \|\boldsymbol{\delta}^{(\nu)}\|_\infty \tag{8.51}$$

Then the spectral radius of the Jacobi matrix may be estimated from

$$\rho(\mathbf{B}_{\mathbf{J}}) \simeq (\bar{\theta} + \omega - 1)/(\omega\bar{\theta}^{1/2}) \tag{8.52}$$

where ω is the estimated value of ω_b used in the generation of $\boldsymbol{u}^{(v)}$ and θ is the stabilised value of $\theta^{(v)}$ from eqn. (8.51). If $\omega \geq \omega_b$ then the value of $\theta^{(v)}$ will oscillate. In such cases ω should be reduced until $\theta^{(v)}$ converges.

Wachspress (1966, p. 109) has given three practical methods for calculating ω_b and he also discusses the advantages and disadvantages of these methods. The third method proposed by Wachspress should be considered when the two methods discussed above prove to be inadequate.

8.3.4 Line and Block SOR Methods

The real power of the SOR method for reservoir simulation lies in its application as a line SOR (LSOR) or block SOR (BSOR) Method. The LSOR method may be developed by writing the matrix **A** in the form of eqn. (8.5). Let us consider a simple example

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{C}_3 & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} \tag{8.53}$$

Now SOR can be applied to block elements as follows:

In the first step solve for $u_1^{(\nu+1)}$ from

$$\mathbf{A}_1 \boldsymbol{u}_1^* = -\mathbf{B}_1 \boldsymbol{u}_2^{(v)} + \boldsymbol{d}_1 \tag{8.54a}$$

$$\boldsymbol{u}_1^{(v+1)} = \omega \boldsymbol{u}_1^* + (1-\omega)\boldsymbol{u}_1^{(v)} \tag{8.54b}$$

followed by the computation of $u_2^{(\nu+1)}$ from

$$\mathbf{A}_2 \boldsymbol{u}_2^* = -\mathbf{C}_2 \boldsymbol{u}_1^{(v+1)} - \mathbf{B}_2 \boldsymbol{u}_3^{(v)} + \boldsymbol{d}_2 \tag{8.55a}$$

$$\boldsymbol{u}_2^{(v+1)} = \omega \boldsymbol{u}_2^* + (1-\omega)\boldsymbol{u}_2^{(v)} \tag{8.55b}$$

and finally $u_3^{(\nu+1)}$ is computed from

$$\mathbf{A}_3 \boldsymbol{u}_3^* = -\mathbf{C}_3 \boldsymbol{u}_2^{(v+1)} + \boldsymbol{d}_3 \tag{8.56a}$$

$$\boldsymbol{u}_3^{(v+1)} = \omega \boldsymbol{u}_3^* + (1-\omega)\boldsymbol{u}_3^{(v)} \tag{8.56b}$$

For this particular example, the above process involves the solution of three tridiagonal matrix equations for each iteration. The generalisation of this method for equations with more than three block rows should be obvious.

Since each partition of matrix **A** corresponds to one line (row) of unknowns, this line is solved simultaneously during the iteration, which gives the method its name. The optimum value of the iteration parameter, ω_b, can be estimated by the methods described for the point SOR method.

The block SOR method is a more general form of the LSOR. It involves the consideration of more than one row (say 2 or 3 as a single block). The convergence properties of the SOR method improve as the block size involved in direct elimination is increased, but the work for performing each step of the iteration also increases. These methods can be very efficient when the size of the problem permits the saving of the inverse of all blocks (in our example $\mathbf{A}_i^{-1}$) in the computer memory.

Finally, we remark that the blocks of the block SOR method can be chosen arbitrarily and do not have to be organised by lines or planes, although this is usually the most convenient choice.

8.3.5 Additive Correction Methods

These methods are based on the idea that the application of some additive correction to the values computed by some iteration scheme at some level of iteration (v) may speed up the convergence of the iteration scheme. There are several methods of deriving the value of the additive correction and they are discussed in the papers by Poussin (1968), Watts (1971, 1973), Aziz and Settari (1972) and Settari and Aziz (1973). The two papers by the authors of this book include a discussion of the Poussin and Watts methods. Although this basic approach is applicable to any iteration scheme, the advantages of this approach have only been demonstrated when it is applied to some form of the SOR method.

Watts or 1*DC method*. The simplest method for finding the additive correction is due to Watts (1971). It will be referred to here as the one-dimensional correction method (1DC) for reasons that will become clear later.

Let us assume that we have obtained a solution $\boldsymbol{u}^{(v)}$ after v iterations with LSOR. Our objective is to apply a correction to $\boldsymbol{u}^{(v)}$ so that the corrected value is 'closer' to the solution in some sense. This corrected value can then be used as the starting point for the $(v+1)$ iteration. The correction is applied as

$$u_{ij}^{(vc)} = u_{ij}^{(v)} + \alpha_j \qquad \begin{matrix} i = 1, 2, \ldots, I \\ j = 1, 2, \ldots, J \end{matrix} \tag{8.57}$$

i.e., a different correction value is added to the elements along each row in the x-direction. The formulation with lines chosen in the y-direction is completely analogous.

The equations for α_j are derived by forcing the sum of the residuals along every row in the x-direction to be zero when $\boldsymbol{u}^{(vc)}$ is substituted in the residual equation for $\boldsymbol{u}^{(v)}$:

$$R_{ij}^{(v)} = -g_{ij}u_{i,j-1}^{(v)} - c_{ij}u_{i-1,j}^{(v)} - a_{ij}u_{ij}^{(v)} - b_{ij}u_{i+1,j}^{(v)} - f_{ij}u_{i,j+1}^{(v)} + d_{ij} \tag{8.58}$$

The condition on the residuals may be written as

$$\sum_{i=1}^{I} R_{ij}^{(vc)} = 0 \qquad j = 1, 2, \ldots, J \tag{8.59}$$

which results in the following equation for the correction vector $\boldsymbol{\alpha}$:

$$g_j^x \alpha_{j-1} + h_j^x \alpha_j + f_j^x \alpha_{j+1} = R_j^x \qquad j = 1, 2, \ldots, J \tag{8.60}$$

where

$$h_j^x = \sum_{i=1}^{I} (a_{ij} + b_{ij} + c_{ij})$$

and similarly superscript 'x' on the other variables indicates summation of the same variable over all i. For example,

$$g_j^x = \sum_{i=1}^{I} g_{ij.}$$

The equations for $\boldsymbol{\alpha}$ (eqn. (8.60)) may be written in the matrix form as

$$\mathbf{S}^x \boldsymbol{\alpha} = \boldsymbol{R}^x \tag{8.61}$$

where $\mathbf{S}^x$ is a tridiagonal symmetric matrix since

$$g_j^x = f_{j-1}^x$$

The matrix $\mathbf{S}^x$ is positive definite. Therefore eqn. (8.61) has a unique solution. The procedure for the use of this method is as follows:

1. Perform l iterations with LSOR or some other iterative scheme.
2. Calculate $\boldsymbol{\alpha}$ from eqn. (8.61).
3. Apply correction by eqn. (8.57) to the last value of $\boldsymbol{u}$.
4. Go to step (1) and repeat this process until convergence is obtained.

Note that although the direction of lines for residual correction is arbitrary, its choice will affect the convergence. For example, with LSOR, it is natural (and best) to also use the same direction of lines for correction as for the LSOR iteration.

Two-dimensional correction method. This method is described in detail by Aziz and Settari (1972) and Settari and Aziz (1973). It is a natural extension of the previous method and involves two correction vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_J \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_I \end{bmatrix} \tag{8.62}$$

and the corrected solution at each grid point is obtained from

$$u_{ij}^{(\mathrm{vc})} = u_{ij}^{(\mathrm{v})} + \alpha_j + \beta_i \qquad \begin{matrix} i = 1, 2, \ldots, I \\ j = 1, 2, \ldots, J \end{matrix} \tag{8.63}$$

The appropriate equations for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are derived by forcing the sum of the residuals over each line in both directions to be zero simultaneously. This results in a complicated matrix equation for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. An adequate approximation is obtained by solving two independent sets of equations

$$\mathbf{S}^x \boldsymbol{\alpha} = \boldsymbol{R}^x \tag{8.64}$$

and

$$\mathbf{S}^y \boldsymbol{\beta} = \boldsymbol{R}^y \tag{8.65}$$

where an arbitrary row of the second equation may be written as

$$c_i^y \beta_{i-1} + h_i^y \beta_i + b_i^y \beta_{i+1} = R_i^y \tag{8.66}$$

where

$$h_i^y = \sum_{j=1}^{J} (g_{ij} + a_{ij} + f_{ij})$$

and superscript 'y' indicates summation over j. We note that two tridiagonal systems of equations (8.64) and (8.65) must be solved to obtain the correction vectors. The approximate method given by these equations becomes exact for an elliptic problem, i.e., if $P = 0$ in eqn. (8.1).

Other methods. A generalisation of the above methods, where correction is made over regions of arbitrary shape, is straightforward and it is formulated in Settari and Aziz (1973). For example, one such method (*Slot method*) was proposed by Poussin (1968) and it is based on the idea of forcing the sum of the residuals to zero over a two-dimensional region (SLOT). We will not present the details of this method here; however, we will compare it with other methods towards the end of this chapter.

8.3.6 Iterative Alternating Direction Implicit (ADI) Methods

A class of methods known as non-iterative alternating direction methods have already been discussed in the previous chapter. These methods are closely related to the form of the partial differential equation and cannot be applied to elliptic partial differential equations. In this section we present some alternating direction methods that can be applied to finite-difference equations of the form of eqn. (8.25). These methods are discussed in several books (e.g., Varga, 1962; Wachspress, 1966; Ames, 1969; Young, 1971); here we will only present a brief discussion of the original method of Peaceman and Rachford (1955) and one related method.

Let us rewrite eqn. (8.3) by separating the contribution to a_{ij} from the two space derivatives:

$$\begin{aligned}[c_{ij}u_{i-1,j} + ax_{ij}u_{ij} + b_{ij}u_{i+1,j}] \\ + [g_{ij}u_{i,j-1} + ay_{ij}u_{ij} + f_{ij}u_{i,j+1}] + [\varphi_{ij}u_{ij}] = d_{ij}\end{aligned} \tag{8.67}$$

where all of the coefficients have the same meaning as given earlier following eqn. (8.3). Note that c, b, g and f will be negative while ax, ay and φ will be positive. Also note that the time level for all u terms is the unknown level and d_{ij} contains all of the right side. Later when we write the equations in matrix form we will also include the boundary contributions with d_{ij}. It is easy to see that the left side of eqn. (8.67) can be written in matrix form as eqn. (8.4) but this time the matrix **A** is split into three parts

corresponding to the terms in the three sets of square brackets on the left side of eqn. (8.67):

$$(\mathbf{H} + \mathbf{V} + \mathbf{\Sigma})\boldsymbol{u} = \boldsymbol{d} \tag{8.68}$$

where **H** contains all of the contributions to matrix **A** of eqn. (8.4) from $\partial/\partial x[\lambda X(\partial U/\partial x)]$ term, **V** contains all of the contributions to matrix **A** from

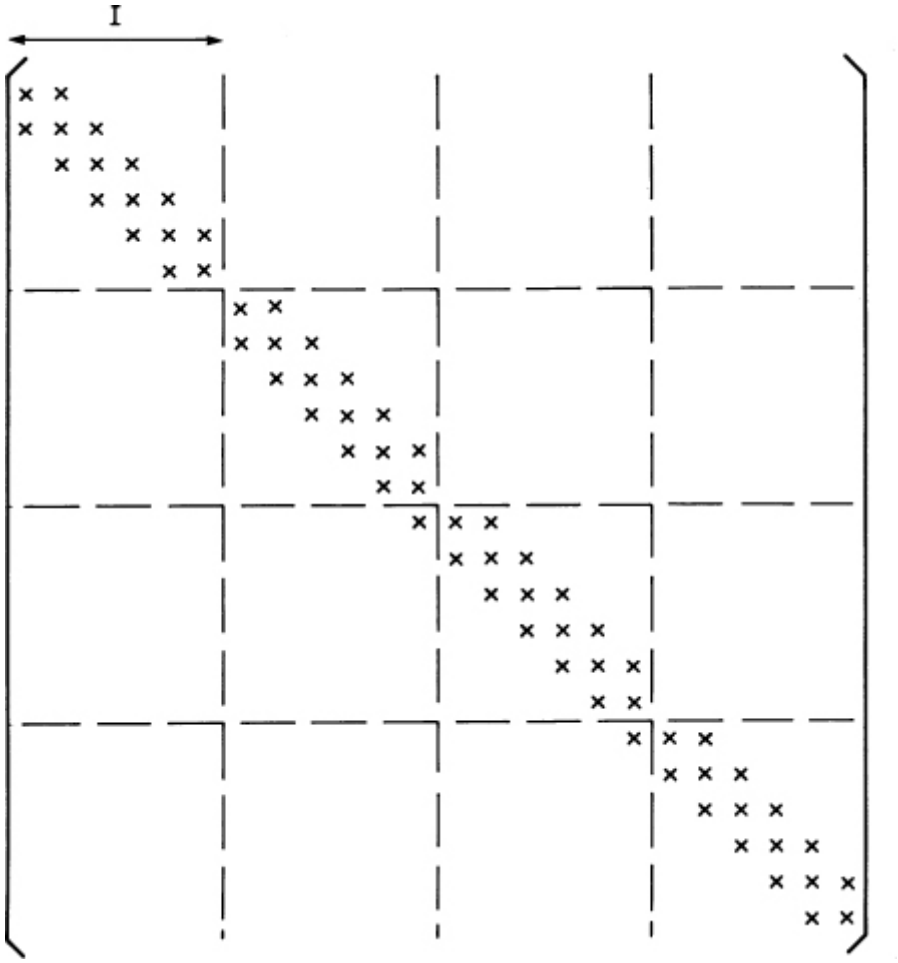


FIG. 8.12. Matrix **H** of eqn. (8.68) when ordering by rows is used. Non-zero entries are indicated by ×. For example, these entries for the tenth row are $c_{3,2}$; $ax_{3,2}$; $b_{3,2}$.

$\partial/\partial y[\lambda Y(\partial U/\partial y)]$ term and $\mathbf{\Sigma}$ contains the contributions from the time derivative. If the standard ordering shown in Fig. 8.1 a is used then **H** and **V** will take the forms shown in Figs. 8.12 and 8.13, respectively, while $\mathbf{\Sigma}$ is diagonal matrix containing φ_{ij}. Furthermore, if the ordering shown in Fig. 8.1b is used, the **V** matrix takes the form shown in Fig. 8.14. It is clear from these figures that: (1) an equation of the form

$$\mathbf{H}\boldsymbol{u} = \boldsymbol{k} \tag{8.69}$$

could be solved by a repeated application of some tridiagonal matrix solution algorithm, and (2) an equation of the form

$$\mathbf{V}\boldsymbol{u} = \boldsymbol{k} \tag{8.70}$$

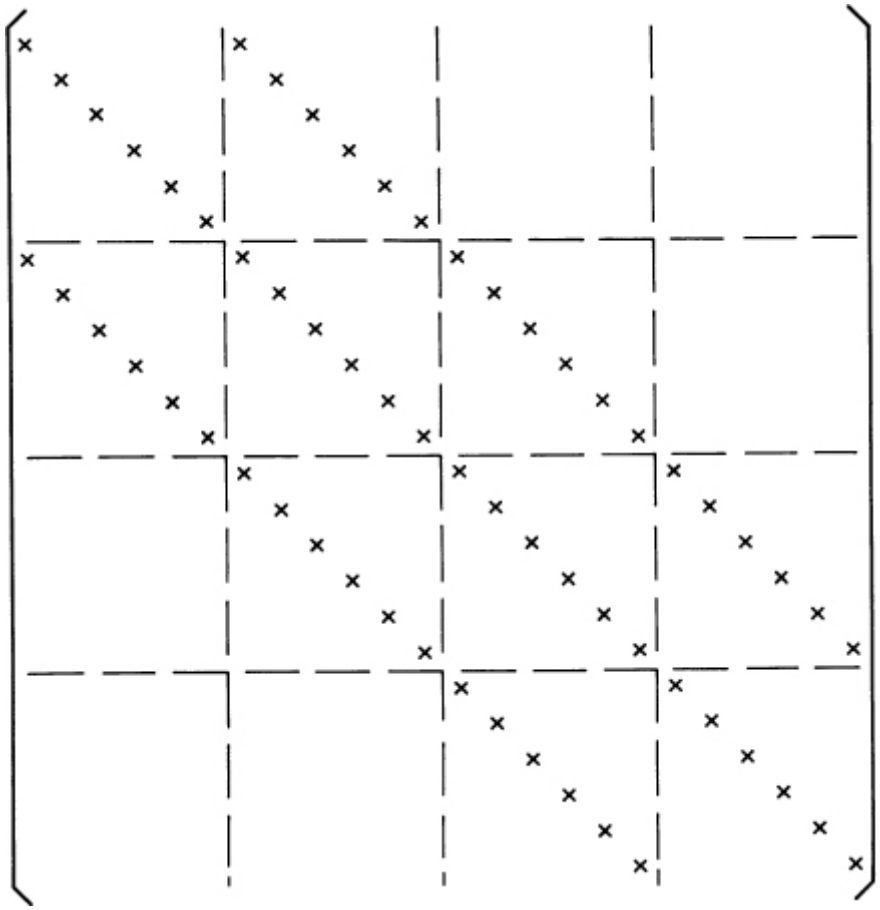


FIG. 8.13. Matrix **V** of eqn. (8.68) when ordering by rows is used. Non-zero entries are indicated by ×. For example, these entries for the tenth row are $g_{3,2}$; $ay_{3,2}$; $f_{3,2}$.

will reduce to the form of eqn. (8.69) as shown by Fig. 8.14 if the unknown vector $\boldsymbol{u}$ is re-ordered by columns (Fig. 8.1b). Motivated by these observations we can write eqn. (8.68) as (Varga, 1962):

$$(\mathbf{H} + \tfrac{1}{2}\boldsymbol{\Sigma} + r\mathbf{I})\boldsymbol{u} = (r\mathbf{I} - \mathbf{V} - \tfrac{1}{2}\boldsymbol{\Sigma})\boldsymbol{u} + \boldsymbol{d} \tag{8.71a}$$

or

$$(\mathbf{V} + \tfrac{1}{2}\boldsymbol{\Sigma} + r\mathbf{I})\boldsymbol{u} = (r\mathbf{I} - \mathbf{H} - \tfrac{1}{2}\boldsymbol{\Sigma})\boldsymbol{u} + \boldsymbol{d} \tag{8.71b}$$

where r is some positive scalar to be discussed later. Both of these equations can be solved readily. If we define

$$\mathbf{H}_1 = \mathbf{H} + \tfrac{1}{2}\boldsymbol{\Sigma} \tag{8.72a}$$

and

$$\mathbf{V}_1 = \mathbf{V} + \tfrac{1}{2}\boldsymbol{\Sigma} \tag{8.72b}$$

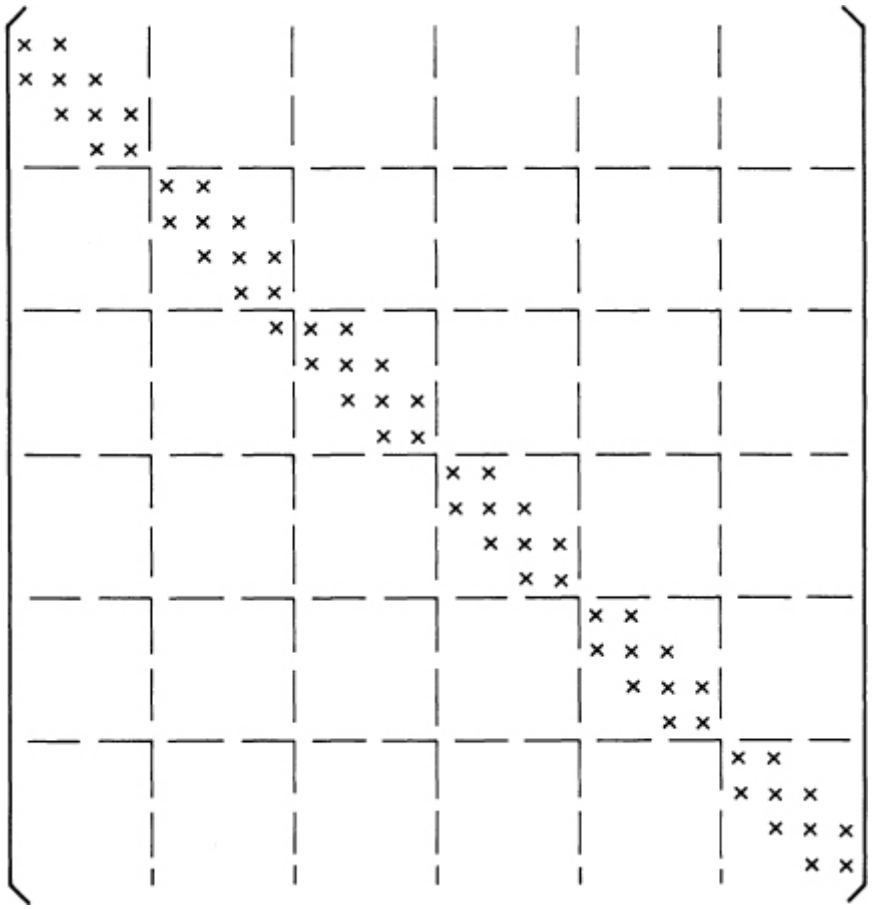


FIG. 8.14. Matrix $\mathbf{V}$ of eqn. (8.68) when ordering by columns is used. Non-zero entries are indicated by ×. For example, these entries for the tenth row are $g_{3,2}$; $ay_{3,2}$; $f_{3,2}$.

then we can write the following iterative process for the solution of eqn. (8.4):

$$(\mathbf{H}_1 + r^{(\nu+1)}\mathbf{I})\boldsymbol{u}^* = (r^{(\nu+1)}\mathbf{I} - \mathbf{V}_1)\boldsymbol{u}^{(\nu)} + \boldsymbol{d} \tag{8.73a}$$

$$(\mathbf{V}_1 + r^{(\nu+1)}\mathbf{I})\boldsymbol{u}^{(\nu+1)} = (r^{(\nu+1)}\mathbf{I} - \mathbf{H}_1)\boldsymbol{u}^* + \boldsymbol{d} \tag{8.73b}$$

$$\nu = 0, 1, 2, \ldots$$

where $\boldsymbol{u}^{(0)}$ is an arbitrary approximation of the solution vector $\boldsymbol{u}$ and $r^{(v)}$ are iteration parameters chosen to accelerate convergence of the iterative process. This is the Peaceman and Rachford (1955) method. Equation (8.73a) is solved first by considering unknowns for each line in the x-direction. The final result of this step is $\boldsymbol{u}^*$. After completing this step (also called 'sweep') we alter the direction in which the unknowns are ordered and now solve for $\boldsymbol{u}^{(v+1)}$ from eqn. (8.73b) by considering unknowns along each line in the y-direction. This process is continued until convergence is obtained. The power of these methods can only be realised by using a sequence of parameters $r^{(m)}$ in a cycle and repeating the cycles until $\boldsymbol{u}^{(v)}$ converges to $\boldsymbol{u}$. This also points out the major problem in the application of ADI methods: What is the optimum number of iterations in a cycle and what are the optimum parameters for the cycle? There are many practical methods for the selection of iteration parameters and many of these are contained in the books dealing with ADI methods mentioned earlier. There are two procedures in common use.

Estimation of iteration parameters by the Peaceman–Rachford method. Let a be the lower bound for the eigenvalues of $\mathbf{V}$ and b the upper bound for eigenvalues of $\mathbf{H}$. The procedure is as follows:

1. Estimate a, b and compute $c = a/b$
2. Find the smallest integer M such that $(0{\bullet}414)^{2M} \leq c$
3. Calculate $r^{(m)}$ as

$$r^{(m)} = b(c)^{(2m-1)/2M} \qquad m = 1, 2, \ldots, M \tag{8.74}$$

4. Estimate the asymptotic rate of convergence from

$$R_\infty(\text{ADI}) = -\frac{2}{M}\ln\left[\frac{1 - c^{1/(2M)}}{1 + c^{1/(2M)}}\right]$$

Estimation of iteration parameters by the Wachspress (1962) *method*

1. Estimate a, b and compute $c = a/b$
2. Find the smallest integer M such that

$$(0{\cdot}172)^{M-1} \leq c$$

3. Calculate $r^{(m)}$ from a geometric sequence:

$$r^{(m)} = b(c)^{(m-1)/(M-1)} \qquad M > 2; m = 1, 2, \ldots, M \tag{8.75}$$

4. Estimate the asymptotic rate of convergence from

$$R_\infty(\mathrm{ADI}) = -\frac{2}{M}\ln\left[\frac{1 - c^{(1/2)(M-1)}}{1 + c^{(1/2)(M-1)}}\right]^2$$

The Wachspress (1962) method is considered superior for most practical problems. In many practical cases it is difficult to estimate a and b. For such cases the largest value of $r^{(m)}$ is selected close to 1 and the parameters distributed in a geometric sequence:

$$r^{(1)} = r_{\min} \qquad r^{(M)} = r_{\max}$$

$$\frac{r^{(m+1)}}{r^{(m)}} = \gamma \qquad m = 1, 2, \ldots, M-1$$

$$\gamma = \left(\frac{r_{\max}}{r_{\min}}\right)^{1/M-1}$$

The above method is the same as eqn. (8.75) based on the work of Wachspress (1962) with $r_{\min} = a$ and $r_{\max} = b$. The lower end of the sequence and M are selected by trial and error (Peaceman, 1966).

There are many other versions of the ADI methods but they will not be discussed here. The interested reader should consult references mentioned earlier. Here we present only one other scheme which is a variant of the Peaceman–Rachford method discussed earlier (Varga, 1962):

$$(\mathbf{H}_1 - r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^* = -(\mathbf{V}_1 + r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^{(\nu)} + \boldsymbol{d} \tag{8.76}$$

$$(\mathbf{V}_1 - r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^{\nu+1} = -(\mathbf{H}_1 + r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^* + \boldsymbol{d}$$

where $\mathbf{D}$ is a diagonal matrix containing $(ax_{ij} + ay_{ij})$. The above form of the ADI method was proposed by Wachspress and Habetler (1960). Another form of this method that is often used in reservoir simulation (Varga, 1962, p. 242) is given by the following equations:

$$(\mathbf{H} + \boldsymbol{\Sigma} + r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^* = (r^{(\nu+1)}\mathbf{D} - \mathbf{V})\boldsymbol{u}^{(\nu)} + \boldsymbol{d} \tag{8.77}$$

$$(\mathbf{V} + \boldsymbol{\Sigma} + r^{(\nu+1)}\mathbf{D})\boldsymbol{u}^{(\nu+1)} = (r^{(\nu+1)}\mathbf{D} - \mathbf{H})\boldsymbol{u}^* + \boldsymbol{d}$$

In the above method Σ is not distributed in a symmetric way between the two sides as was the case for eqn. (8.76). Generalisations of this form for multiphase flow are in common use for reservoir simulation (Chapters 10 and 11).

Iteration parameters for this scheme may be selected by the same procedure as described earlier for the Peaceman–Rachford method. Björdammen and Coats (1969)

have suggested that r_{min} may be estimated from

$$r_{min} = \text{minimum over the grid of} \left[\frac{\pi^2}{(2I^2)\left(1 + \frac{(\Delta x)^2 \lambda Y}{(\Delta y)^2 \lambda X}\right)}, \quad \frac{\pi^2}{(2J^2)\left(1 + \frac{(\Delta y)^2 \lambda X}{(\Delta x)^2 \lambda Y}\right)} \right] \tag{8.78a}$$

$$r_{max} = 1$$

for $\lambda X(\Delta y/\Delta x) - \lambda Y(\Delta x/\Delta y)$. For two-dimensional cross-sectional problems where $\lambda X(\Delta z/\Delta x) < \lambda Z(\Delta x/\Delta z)$they recommend

$$r_{min} = \text{minimum over the grid of} \left[\frac{\pi^2}{(2I^2)\left(1 + \frac{(\Delta x)^2 \lambda Z}{(\Delta z)^2 \lambda X}\right)} \right]$$

and

$$r_{max} = 2 \tag{8.78b}$$

The choice of $r_{max} = 2$ appears to be empirical. In the calculation of r_{min} the points for which λX, λY or λZ is zero should be excluded. In both cases the parameters are distributed geometrically between the minimum and maximum values, as shown earlier. The number of parameters may also be predicted by the Wachspress (1962) method by calculating the smallest integer, M, that satisfies

$$(0{\cdot}172)^{M-1} \leq \frac{r_{min}}{r_{max}}$$

8.3.7 Strongly Implicit Method

As discussed earlier the matrix equation (8.4) may be solved by direct elimination by factoring **A** into the product of a lower triangular matrix **L** and an upper triangular matrix **U**. In general **L** will have non-zero elements from the main diagonal to the diagonal corresponding to the diagonal of f values in Fig. 8.2a. Similarly, the matrix **U** will have non-zero elements from the main diagonal to the diagonal corresponding to g values in Fig. 8.2a. In the elimination process each of the elements of **U** must be computed and stored for later use. As indicated by Stone (1968) for each grid point approximately $I + J$ such elements must be computed which makes the elimination

process slow for large I and J. The basic idea of the *strongly implicit procedure* (SIP) is to alter **A** so that the factoring of the altered matrix is easier and use this altered matrix to define an iteration scheme for the solution of eqn. (8.4).

Let us consider eqn. (8.4):

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d}$$

and add $\mathbf{N}\boldsymbol{u}$ to both sides, and add and subtract $\mathbf{A}\boldsymbol{u}$ from the right side to yield

$$(\mathbf{A} + \mathbf{N})\boldsymbol{u} = (\mathbf{A} + \mathbf{N})\boldsymbol{u} - (\mathbf{A}\boldsymbol{u} - \boldsymbol{d}) \tag{8.79}$$

Stone (1968) has proposed a method of finding **N** so that **A** + **N** is easily

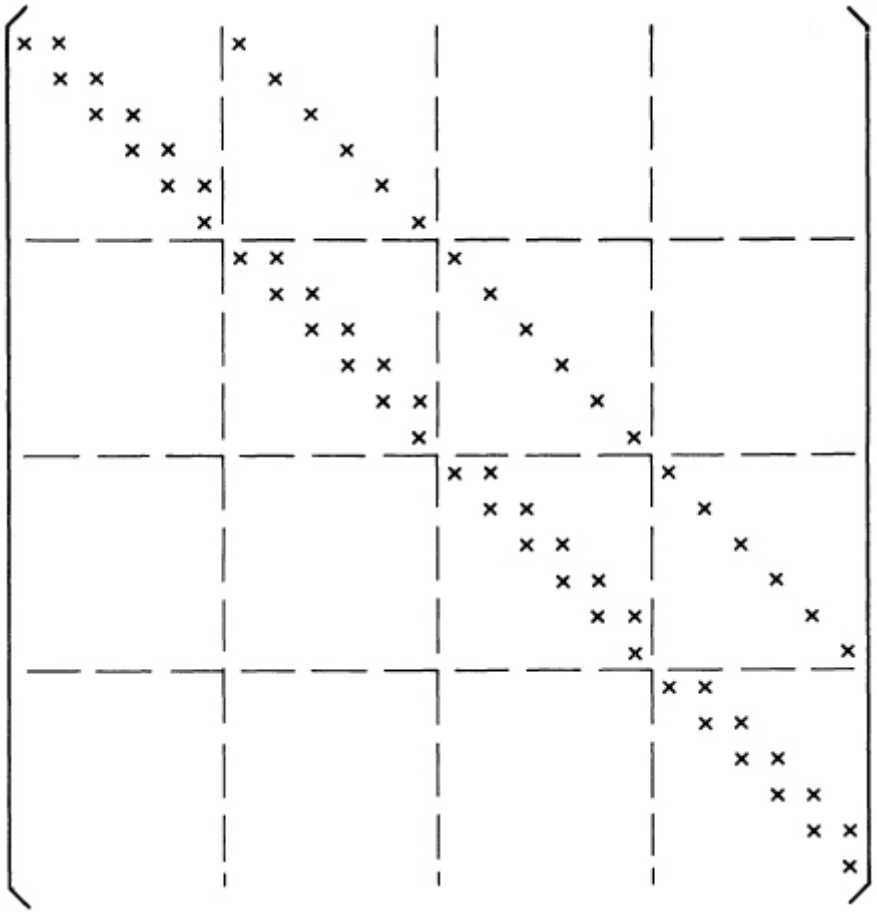


FIG. 8.15. Form of **U** matrix for Stone's method. Non-zero entries are indicated by ×. For example, these entries for the tenth row are 1; $b_{3,2}$; $f_{3,2}$.

factored into a product **LU**, such that **L** and **U** have only three non-zero elements in each row as shown in Figs. 8.15 and 8.16 for the grid of Fig. 8.1 a. The elements of **L** and **U** matrices may be computed recursively from

$$
\begin{aligned}
g_{ij} &= g_{ij}/(1 + \alpha b_{i,j-1}) \\
c_{ij} &= c_{ij}/(1 + \alpha f_{i-1,j}) \\
a_{ij} &= a_{ij} + \alpha g_{ij} b_{i,j-1} + \alpha c_{ij} f_{i-1,j} - g_{ij} f_{i,j-1} - c_{ij} b_{i-1,j} \\
b_{ij} &= (b_{ij} - \alpha g_{ij} b_{i,j-1})/a_{ij} \\
f_{ij} &= (f_{ij} - \alpha c_{ij} f_{i-1,j})/a_{ij}
\end{aligned}
\tag{8.80}
$$

where α is an iteration parameter to be discussed later. We can now write an iteration procedure using eqn. (8.79) as

$$
(\mathbf{A} + \mathbf{N})\boldsymbol{u}^{(\nu+1)} = (\mathbf{A} + \mathbf{N})\boldsymbol{u}^{(\nu)} - (\mathbf{A}\boldsymbol{u}^{(\nu)} - \boldsymbol{d}) \tag{8.81}
$$

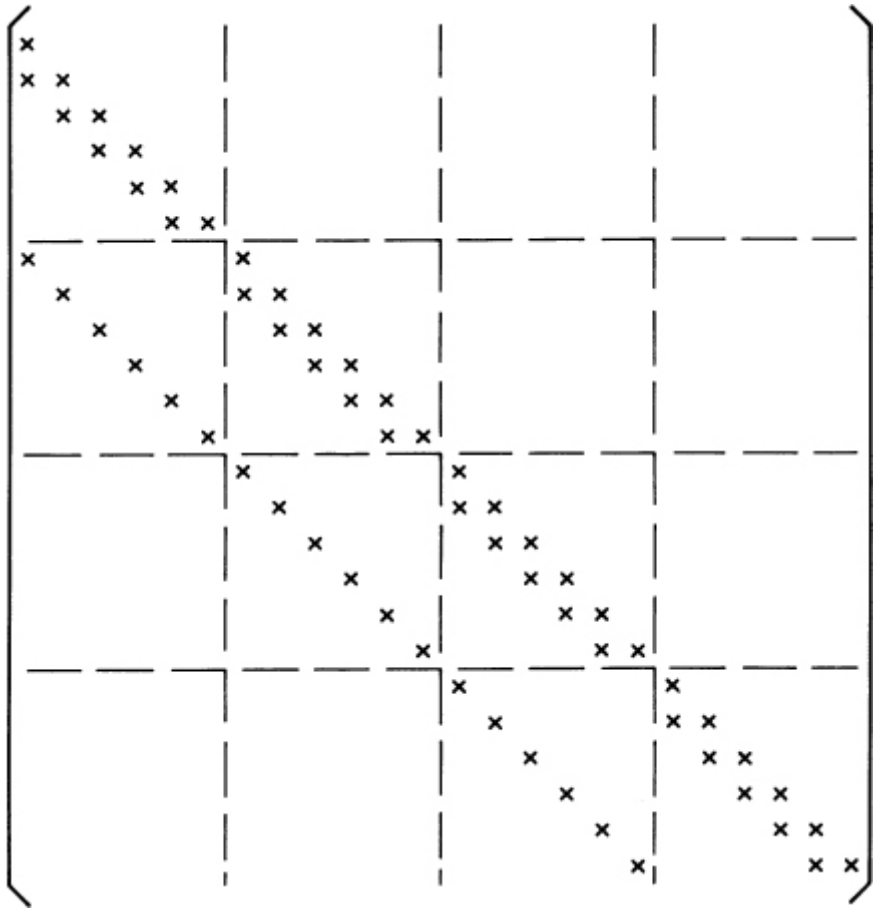


FIG. 8.16. Form of **L** matrix for Stone's method. Non-zero entries are indicated by ×. For example, these entries for the tenth row are $g_{3,2}$; $c_{3,2}$; $a_{3,2}$.

From a computational point of view it is better to write eqn. (8.81) in residual form. Let

$$
\boldsymbol{R}^{(\nu)} = \mathbf{A}\boldsymbol{u}^{(\nu)} - \boldsymbol{d} \tag{8.82}
$$

$$\boldsymbol{\delta}^{(v+1)} = \boldsymbol{u}^{(v+1)} - \boldsymbol{u}^{(v)} \tag{8.83}$$

and rewrite eqn. (8.81) as

$$(\mathbf{A} + \mathbf{N})\boldsymbol{\delta}^{(v+1)} = -\boldsymbol{R}^{(v)} \tag{8.84}$$

or

$$\mathbf{LU}\boldsymbol{\delta}^{(v+1)} = -\boldsymbol{R}^{(v)} \tag{8.85}$$

The elements of **L** and **U** are computed from eqn. (8.80). The solution of (8.85) is obtained by defining a vector $\boldsymbol{v}$ so that

$$\mathbf{L}\boldsymbol{v} = -\boldsymbol{R}^{(v)} \tag{8.86}$$

The elements of vector $\boldsymbol{v}$ may be computed by forward substitution. From eqns. (8.85) and (8.86) we see that

$$\mathbf{U}\boldsymbol{\delta}^{(v+1)} = \boldsymbol{v} \tag{8.87}$$

and this equation may be used to compute $\delta^{(v+1)}$ by backward substitution. Stone (1968) recommends the use of a sequence of iteration parameters in a cycle. These parameters are geometrically spaced between 0 and α_{max}, where

$(1 - \alpha_{max})$ = minimum over the grid of

$$\left[\frac{\pi^2}{(2I^2)\left(1 + \dfrac{(\Delta x)^2\lambda Y}{(\Delta y)^2\lambda X}\right)}, \quad \frac{\pi^2}{(2J^2)\left(1 + \dfrac{(\Delta y)^2\lambda X}{(\Delta x)^2\lambda Y}\right)}\right] \tag{8.88}$$

The iteration parameters are computed from

$$(1 - \alpha_m) = (1 - \alpha_{max})^{m/(M-1)} \qquad m = 0, 1, 2, \ldots, M-1 \tag{8.89}$$

where M is the number of parameters per cycle. Stone (1968) recommends the use of a minimum of four parameters, each used twice, per cycle (Section 10.3.3, Chapter 10). Note that the right side of eqns. (8.77) and (8.88) are the same.

In each iteration, elements of matrices **L** and **U** and vector $\boldsymbol{v}$ can be obtained in a recursive fashion according to the ordering of unknowns (in our case in the order of increasing i for rows $j = 1, 2, \ldots, J$). The elements of $\boldsymbol{\delta}$ are then obtained in the reverse order. In the practical application of the above procedure, the ordering of the equations

is changed for every iteration. This change in ordering consists of carrying out the calculations described above in the reverse order for j while allowing i to vary in the same manner. Thus for *odd numbered iterations*

$$i = 1, 2, \ldots, I$$

for each value of j in the order

$$j = 1, 2, \ldots, J$$

and for *even number iterations* i is allowed to increment as before,

$$i = 1, 2, \ldots, I$$

but this is done for each value of J in the reverse order

$$j = J, J-1, \ldots, 1$$

This reversal in direction is important for convergence of the method in some cases and may be implemented without a separate coding for even numbered iterations. Furthermore, it is also possible to use more than two different orderings by varying the direction of i also.

Letkeman (1976) has presented a similar method that provides a different method of altering **A** in order to get an easy **LU** factoring. For some problems Letkeman claims his method to be superior to Stone's method.

8.3.8 Other Methods

In a book of this type it is not possible to provide even a brief discussion of all available methods for the solution of eqn. (8.4). The methods presented in previous sections are in common use for reservoir simulation problems. In this section selected other methods that may be applicable to reservoir simulation problems under certain conditions are mentioned.

All of the iterative methods discussed so far were of the form

$$\boldsymbol{u}^{(\nu+1)} = \mathbf{B}\boldsymbol{u}^{(\nu)} + \boldsymbol{k} \tag{8.90}$$

This procedure can be generalised by making use of the solution from several previous iteration levels. This is accomplished by writing

$$\boldsymbol{v}^{(\nu)} = \mathbf{B}\boldsymbol{v}^{(\nu-1)} + \boldsymbol{k} \qquad \nu = 1, 2, \ldots \qquad \text{and} \qquad \boldsymbol{v}^{(0)} \text{arbitrary} \tag{8.91}$$

$$\boldsymbol{u}^{(\nu)} = b_\nu \boldsymbol{v}^{(\nu)} + \sum_{l=0}^{\nu-1} b_{l,\nu} \boldsymbol{v}^{(l)}$$

Usually only one or two terms of the summation are required. Different approaches for the selection of b values in the above equations result in different methods. These include the method of Lanczos (1950, 1952), *minimised iterations*, *conjugate gradients and steepest descents* (Wachspress, 1966; Engeli *et al*., 1959; Faddeev and Faddeeva, 1963).

Another class of methods based on an approach similar to that used by Stone (1968) has been proposed by Meijerink and Van der Vorst (1977). They call these methods *incomplete* **LU** *decomposition* and combine some of these methods with conjugate gradient methods. The results are impressive for symmetric matrices. Approaches of this type are worthy of detailed study.

8.3.9 Comparison of Iterative Methods

Settari and Aziz (1973) compared LSOR, ADI, SIP and LSOR with various correction schemes. The results were presented for seven different problems. In this section we summarise their results and provide a comparison with some recent methods. The ADI method used here is given by eqn. (8.76).

Test Problems

The test problems used are extensions of those used by Stone (1968). In all cases a 31×31 uniform grid is considered on a unit square.

Problem No. 1

$KX = KY = 1$

$P = 0$

Homogeneous Neumann boundary conditions.

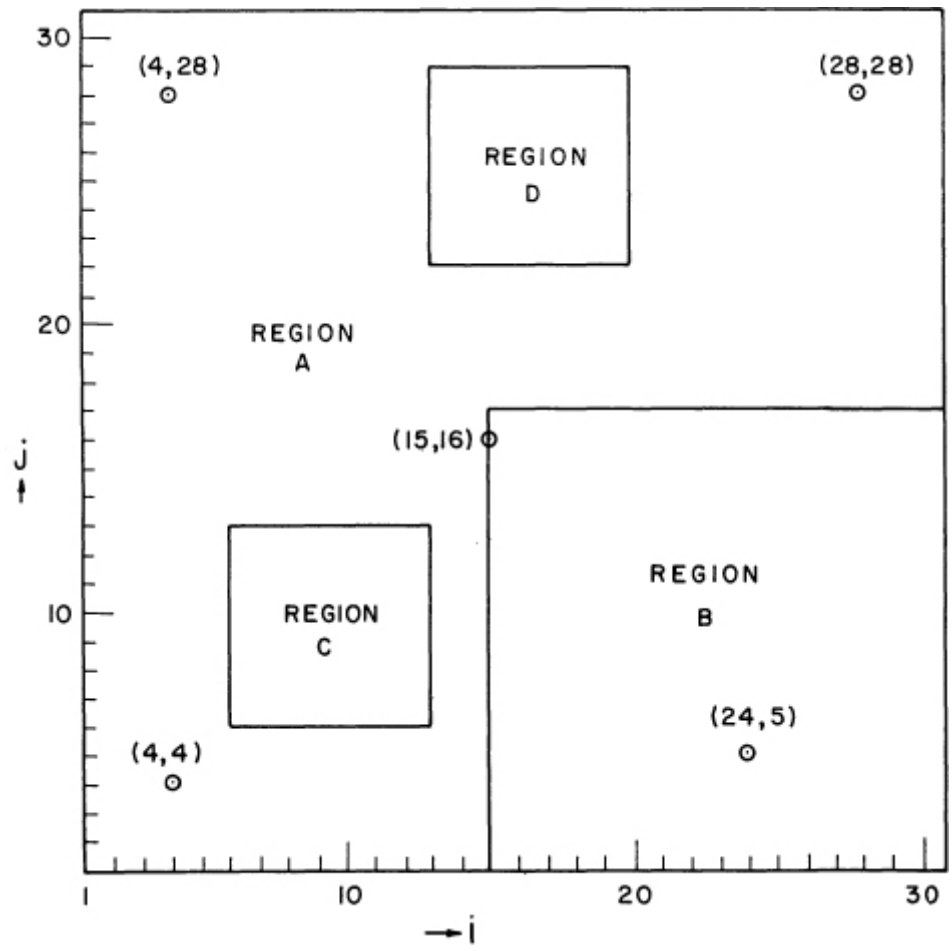


FIG. 8.17. Description of Problem No. 3 (after Stone, 1968). $q_{4,4} = 1{\bullet}0$; $q_{15,16} = -1{\bullet}83$; $q_{4,28} = 0{\bullet}5$; $q_{28,28} = -0{\bullet}27$; $q_{24,5} = 0{\bullet}6$.

Problem No. 2

$KX = 1 \quad KY = 0{\bullet}01$

$P = 0$

Homogeneous Neumann boundary conditions.

Problem No. 3 (see Fig. 8.17)

$KX = KY = 1$ in Region A

$KX = 1 \quad KY = 100$ in Region B

$KX = 100 \quad KY = 1$ in Region C

$KX = KY = 0$ in Region D

$P = 0$

Homogeneous Neumann boundary conditions.

Problem No. 4

$KX_{(1+½),j} = 0{\bullet}01 \qquad KX_{(30+½),j} = 1{\bullet}0$
$KY_{i,1+½} = 0{\bullet}01 \qquad KY_{i,30+½} = 1{\bullet}0$

with linear variation between the above values. (Such a field may be associated with problems in cylindrical geometry.)

$P = 0$

Homogeneous Neumann boundary conditions.

Problem No. 5

$KX = KY = 100$ in two rectangular regions bounded by $i = 6, 20, j = 7, 16$ and $i, j = 23, 29$
$KX = KY = 1$ elsewhere
$P = 0$

Homogeneous Neumann boundary conditions.

Problem No. 6

KX and *KY* as in Problem No. 3 except now Region B is bounded by $i = 15, 29$ and $j = 3, 17$
P = 0

Homogeneous Dirichlet boundary conditions.

Problem No. 7

KX and *KY* as in Problem No. 3
$P = 100$

Homogeneous Neumann boundary conditions.

Figures 8.18 to 8.24 present a comparison of various iterative methods for the seven problems described above. In these figures $\|R\|_\infty$ is the l_∞ norm of the residual vector divided by the sum of the sources, and n is the equivalent number of LSOR iterations based on work ratios given in Table 8.1. Results for each problem are summarised below:

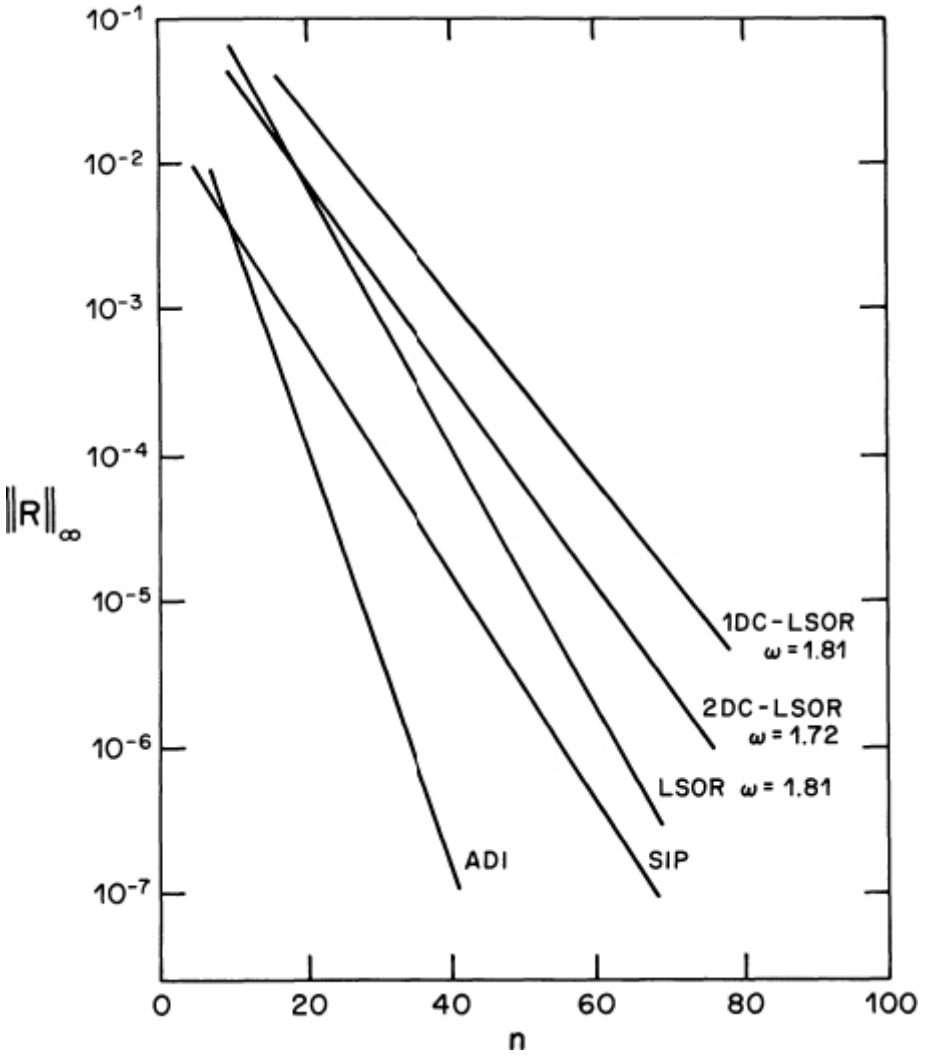


FIG. 8.18. Test results for Problem No. 1 (from Settari and Aziz, 1973).

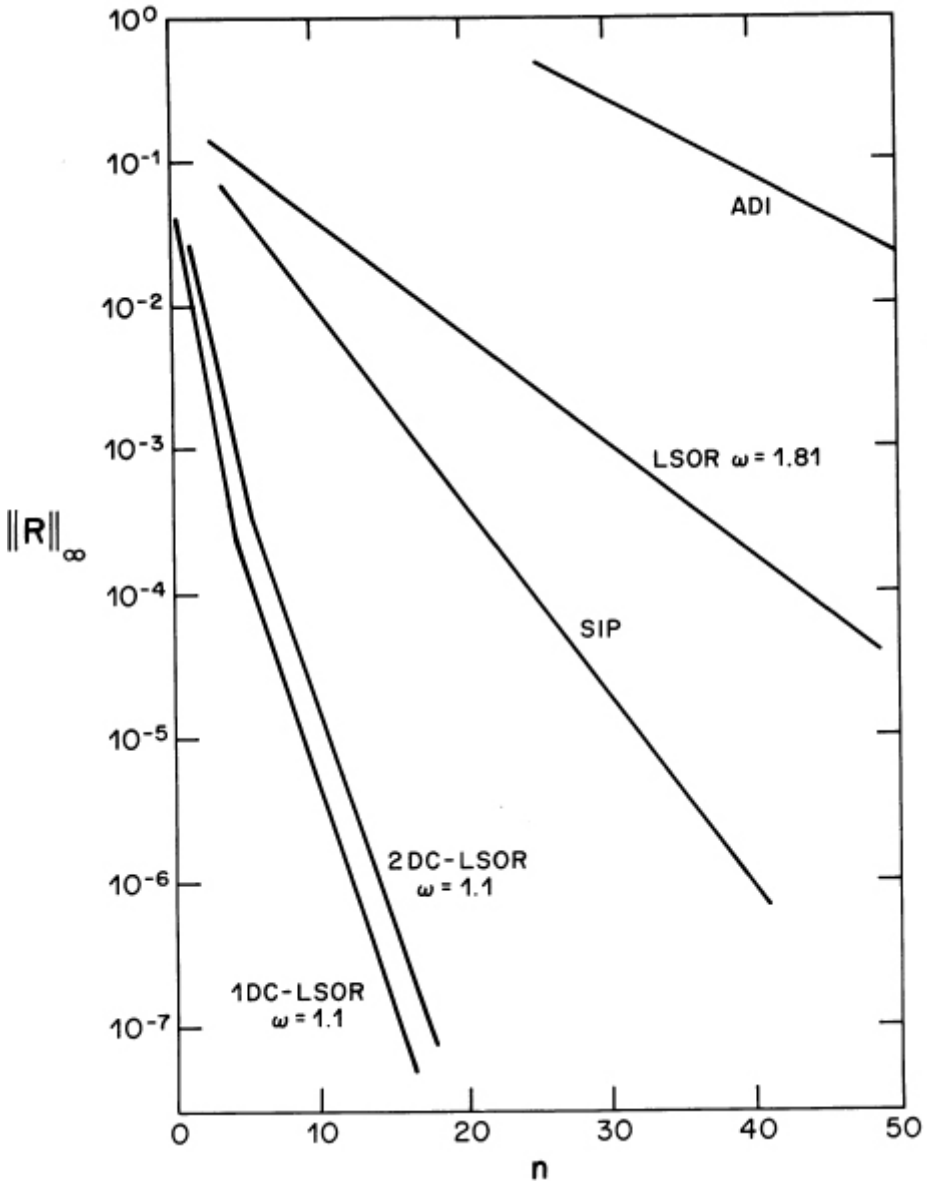


FIG. 8.19. Test results for Problem No. 2 (from Settari and Aziz, 1973).

Problem No. 1 (Fig. 8.18)
Because the coefficient field is uniform, 1DC results in almost no improvement over LSOR, and 2DC gives only a relatively small improvement. Therefore, both are slower than LSOR in terms of computational time. ADI is the best, followed by SIP.

Problem No. 2 (Fig. 8.19)
As expected, 1DC (used in the x-direction) is most efficient, 2DC achieves the same absolute result, however, with more computer time. ADI is the slowest. This problem illustrates the difficulties encountered with ADI for anisotropic problems.

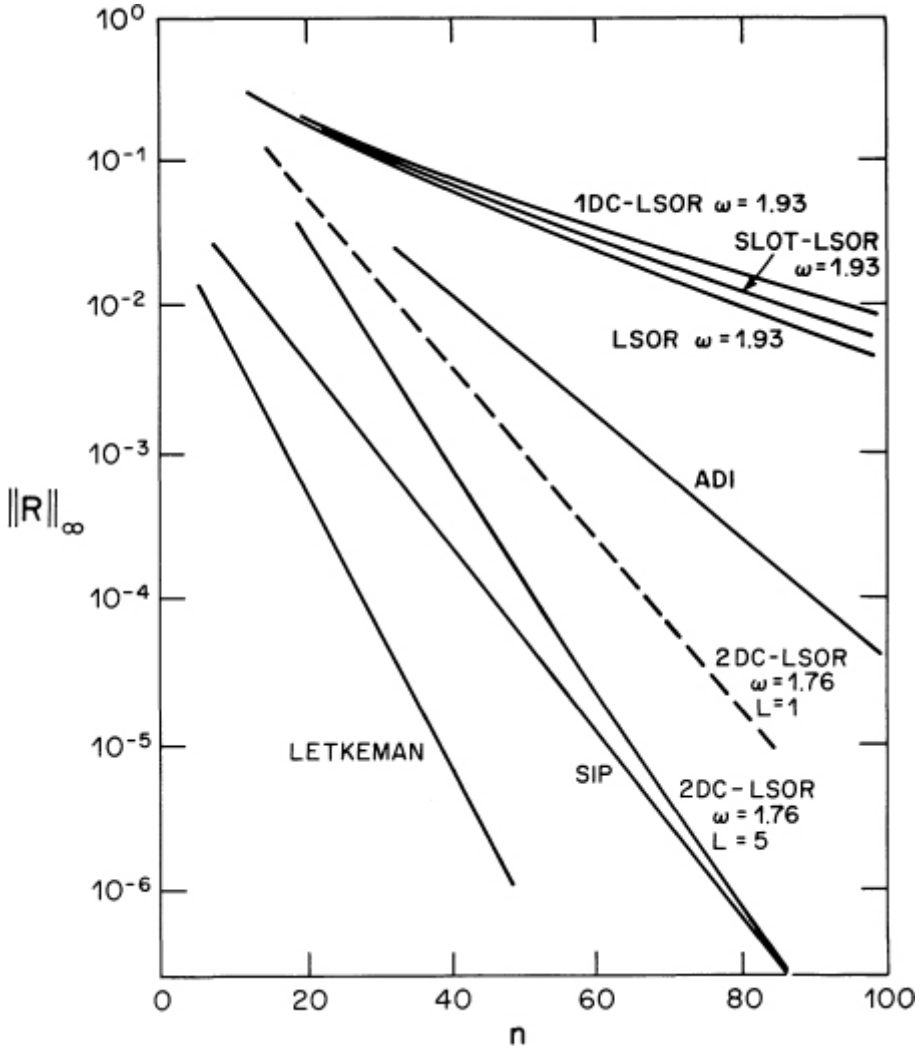


FIG. 8.20. Test results for Problem No. 3 (after Settari and Aziz, 1973).

Problem No. 3 (Fig. 8.20)

This is the first heterogeneous problem. While 1DC fails to improve LSOR, a marked improvement is obtained when 2DC is used with LSOR bringing it close to SIP. This example demonstrates that the correction using partitioning by lines in one direction is not satisfactory for completely general cases. SLOT correction was also applied in this case with each of the homogeneous subregions taken as one partition. The 2DC method maintains a convergence rate comparable with SIP. The method of Letkeman (1976) exhibits the best convergence for this problem.

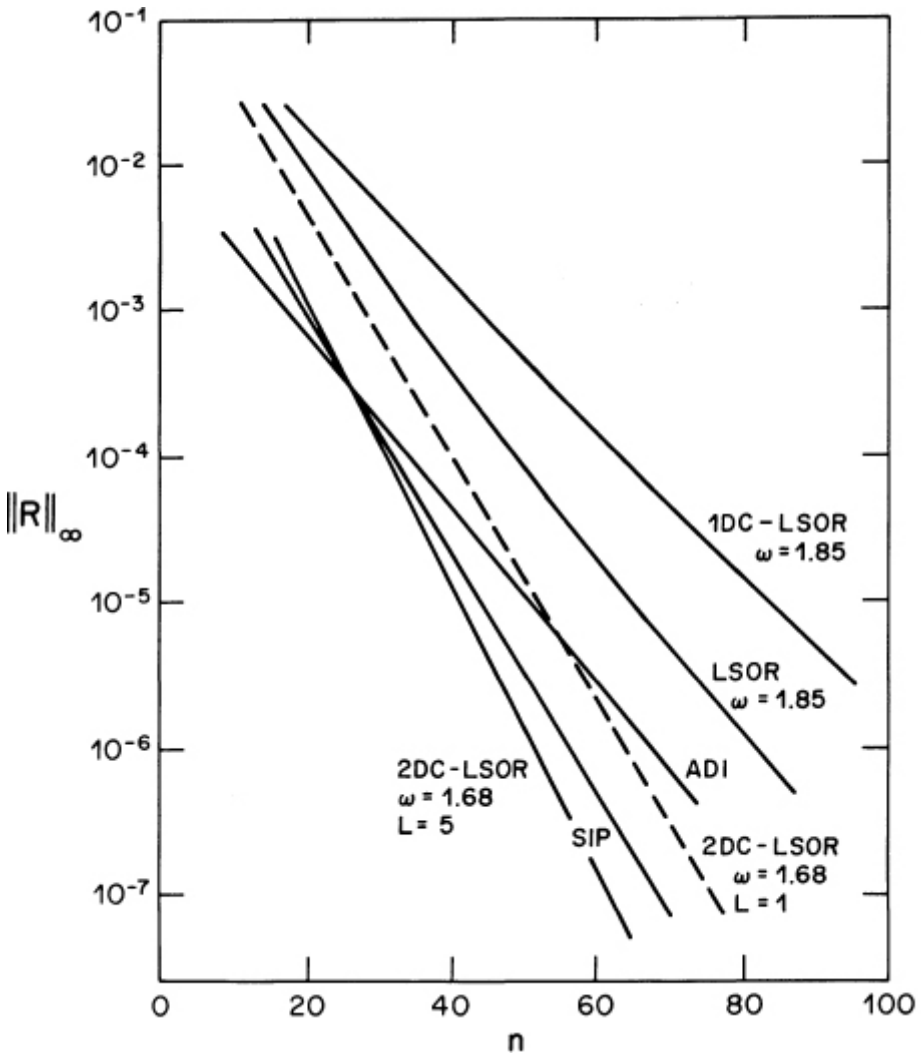


FIG. 8.21. Test results for Problem No. 4 (from Settari and Aziz, 1973).

Problem No. 4 (Fig. 8.21)
The situation is essentially the same as in Problem No. 3 except that ADI and LSOR perform better in this case than in Problem No. 3. This is explained by the fact that the variation in coefficients is much smoother in Problem No. 4 than in Problem No. 3. 1DC again gives no improvement.

Problem No. 5 (Fig. 8.22)
As expected, the SLOT method of Poussin (1968) is very efficient for this problem but 2DC is even better. All other methods (including SIP) are slower. Note that 1DC gives improvement over LSOR.

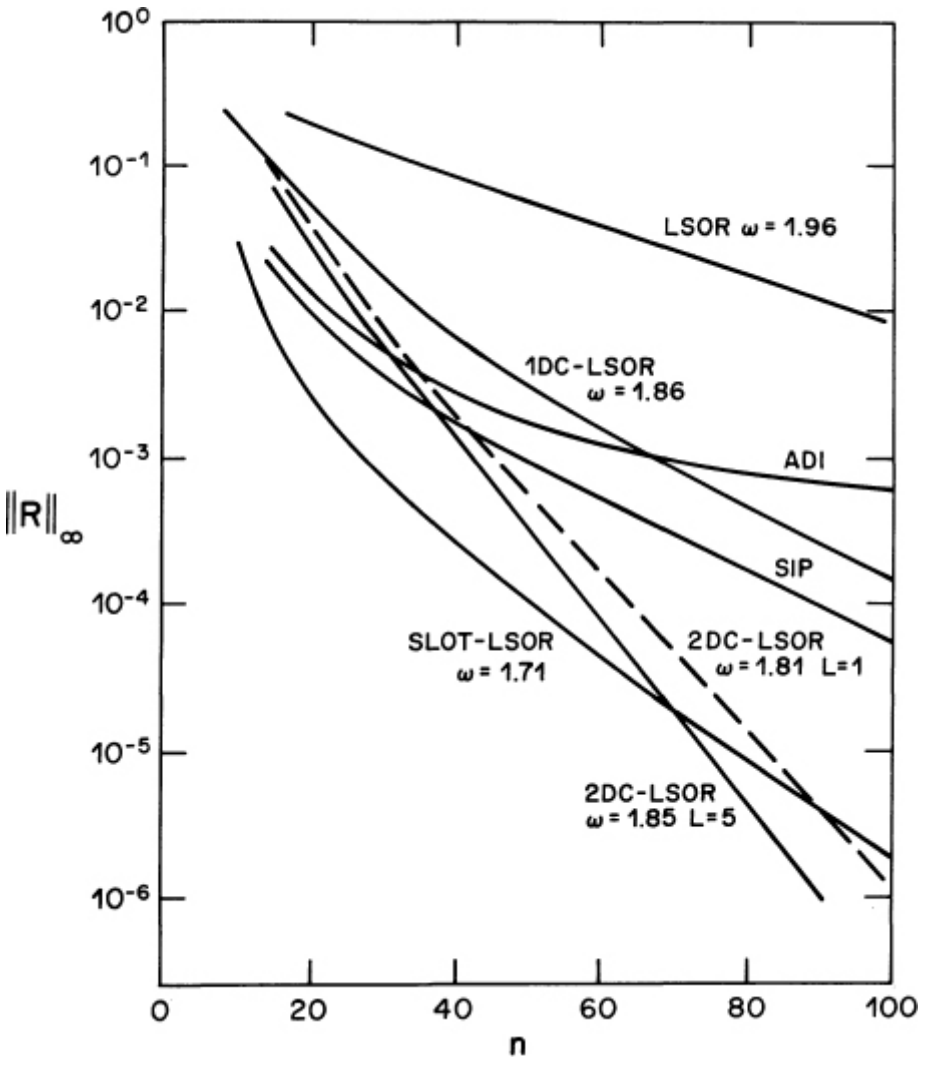


FIG. 8.22. Test results for Problem No. 5 (from Settari and Aziz, 1973).

Problem No. 6 (Fig. 8.23)
This problem shows that 2DC is efficient also for problems with Dirichlet boundary conditions and competes successfully with SIP.

Problem No. 7 (Fig. 8.24)
This is another variant of Problem No. 3 and yields similar results except for ADI (eqn. 8.76) which yields much better results, having the best rate of convergence for this problem.

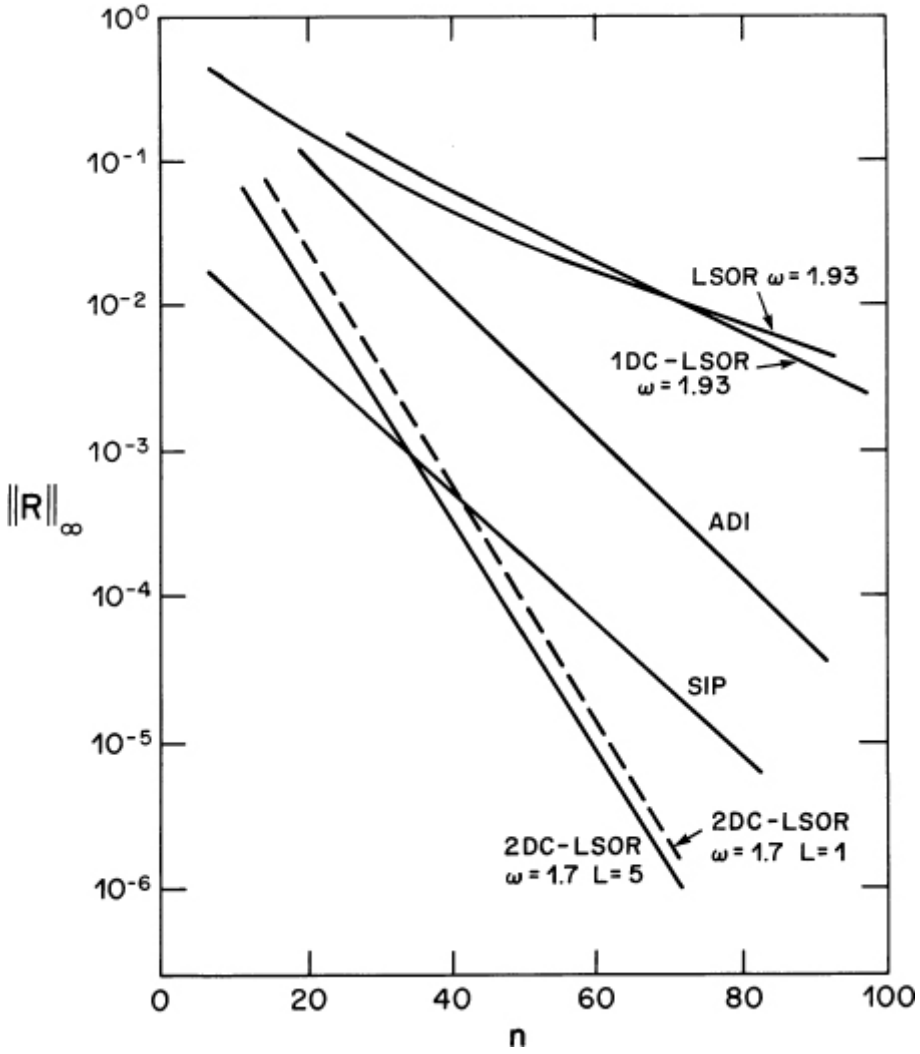


FIG. 8.23. Test results for Problem No. 6 (from Settari and Aziz, 1973).

None of the methods tested is clearly superior to all others in all cases. However, SIP and 2DC seem to be most useful for difficult problems like Nos 3, 5 and 6.

A comparison of iterative techniques for some realistic reservoir problems is given by Traylor and Sheffield (1971). These authors compared various methods for solving the pressure equation of IMPES in two dimensions. Their results confirm conclusions based on our own results presented above and again show that *there is no single iteration method which is best for all problems*. They found that SIP performed best for an areal problem, while LSOR and semi-iterative block Jacobi methods were the best for a cross-sectional problem.

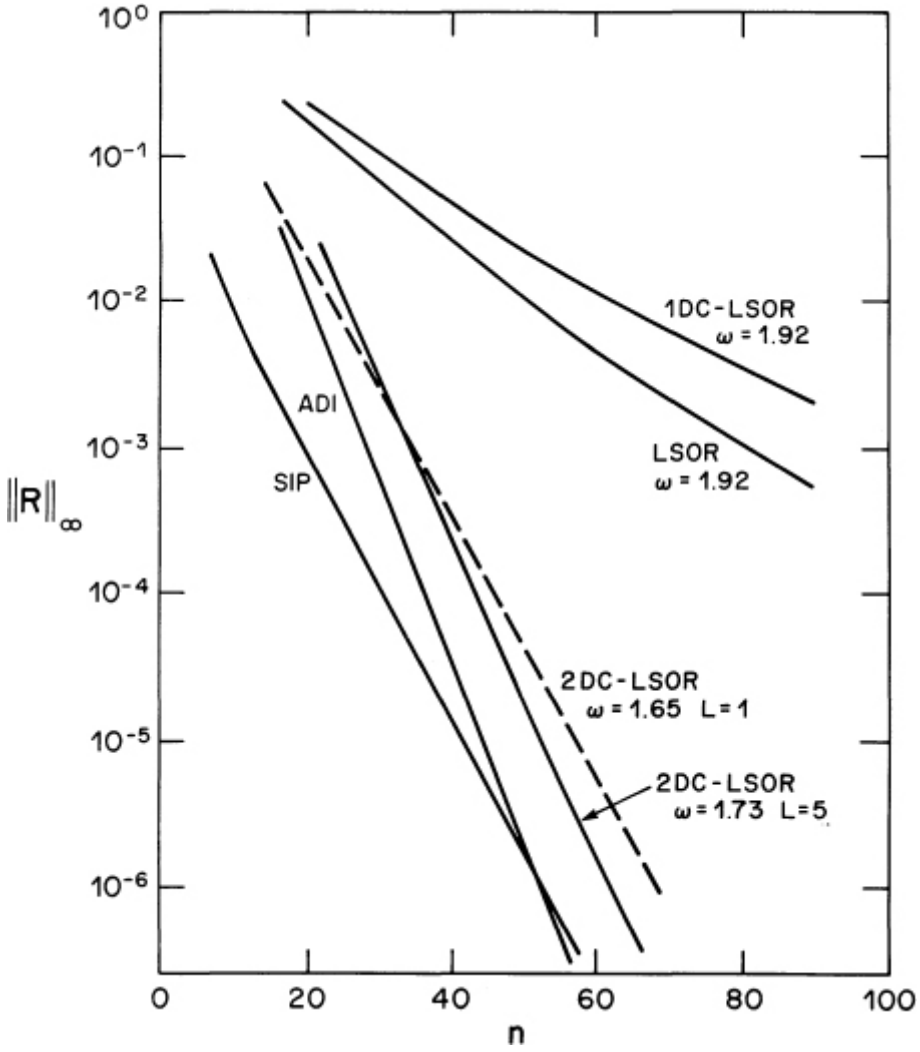


FIG. 8.24. Test results for Problem No. 7 (from Settari and Aziz, 1973).

8.3.10 Practical Considerations in the Use of Iterative Methods

Efficient application of iterative methods is a matter of experience and in many cases requires numerical experimentation. We will discuss some aspects of the problem which are important from a practical standpoint. This is the first and only chapter where iterative methods are discussed in detail; for that reason, some of the comments made here will apply to the solution of pentadiagonal equations arising from the application of IMPES or SEQ to multiphase problems (Chapter 9), and to three-dimensional problems (Chapter 11). This is also true of the discussion in Sections 8.4 and 8.5.

TABLE 8.1

WORK RATIOS USED FOR COMPARISON OF DIFFERENT ITERATIVE METHODS

Method	*Work ratio*
LSOR	1

SLOT–LSOR	1•1
1DC–LSOR	1•3
2DC–LSOR	1•5
SIP	2
ADI	2
Letkeman	4•3

(a) Most point-iterative methods can be formulated as line- (or in general block) iterative methods and they should always be used in such form.

In the LSOR method, the direction of the lines can have a large influence on the rate of convergence for heterogeneous problems. The proper direction can be estimated analytically if the region of interest is approximated by a rectangle with constant λX and λY and the rates of convergence are calculated and compared for the idealised problem. More often, the best direction is determined directly by comparison of runs using both directions. Note that the optimum iteration parameters will also be different for each direction. For example, for LSOR, we only need to predict the ω_b for each case without actually solving the problem and then select the direction giving the *smallest* ω_b.

As a general rule, *the lines should be chosen in the direction of largest transmissibilities*. For example, the best choice of lines for cross-sectional (and 3-D) problems is almost always in the vertical direction.

(b) For a given direction of lines, there are two possible sweep directions. In many cases, alternating the direction of sweep improves convergence. This technique is well known for SIP, but it should also be applied with other methods. The direction of sweep becomes particularly important for equations of hyperbolic character such as the implicit saturation equation of the SEQ method (see Chapters 5 and 9). In such a case the sweep should be applied *in the direction of flow.*

(c) The direction of lines in LSOR may also be changed after each iteration. So with changing the direction of sweep and direction of lines there are four possibilities. No experience is reported in the literature with these methods; however, they may be of value for some problems.

(d) Another variation of SOR methods is obtained by changing the ordering of the points, lines or blocks. This idea is related to the Hopscotch method (see Gourlay, 1970; Gourlay and McGuire, 1971). Again no experience is reported in the literature for reservoir simulation problems. Young (1971) provides several versions of the SOR and related methods.

(e) A special case of BSOR where two lines are considered (2LSOR) has some advantages over single-line LSOR under certain conditions (Parter, 1959, 1961; Varga, 1962).

(f) Experience with ADI methods shows that the distribution of parameters between $r_{\min}$ and $r_{\max}$ and their order is not critical and the convergence can be optimised by adjusting $r_{\min}$ only. (In fact, the two methods, (8.74) and (8.75), give values of $r^{(m)}$ clustered at the opposite ends of the range.) Coats (1968) recommends the use of a geometric sequence with 4–5 parameters for problems with $r_{\min} \geq 0{\bullet}01$ and 6–8 parameters if $r_{\min} \simeq 0{\bullet}0001$–$0{\bullet}001$. The same remarks apply to the SIP parameters, as evidenced by Steen and Farouq Ali (1971) and Suarez and Farouq Ali (1976), who failed to find any general rules for the selection of SIP parameters.

(g) When estimating the optimum iteration parameters by trial and error, the asymptotic rate of convergence (approximated by the average rate of convergence after the convergence becomes asymptotic) for each set of parameters should be calculated. In many cases it is quite satisfactory to observe the behaviour of the norm of the residual vector $\|\boldsymbol{R}\|$ after a fixed, sufficiently large number of iterations, as a function of the iteration parameters, and determine the optimum parameter from the minimum of this curve. Examples of such curves generated for different methods applied to Problems 3 and 4 of the previous section are in Fig. 8.25. Note that for LSOR-based methods for Problem No. 4, the curves have the same character as the theoretical curve of the spectral radius as a function of ω (Fig. 8.11). This character is not shown by the curve for LSOR method for Problem No. 3. Here the convergence is so slow that even after 80 iterations the asymptotic range is not reached. The optimum ω in this case is, in fact, 1•93, but at 80 iterations this value will give worse results than any value $\omega < \omega_b$. Therefore for slowly converging problems estimation of ω_b is more difficult. This applies equally to the methods described in Section 8.3.3, which also converge slowly. The convergence of ADI is shown as a function of the minimum parameter $r_{\min}$ in a geometric sequence of six parameters calculated by eqn. (8.75) between $a = r_{\min}$ and b $= r_{\max} = 1$. This curve also shows a distinct minimum.

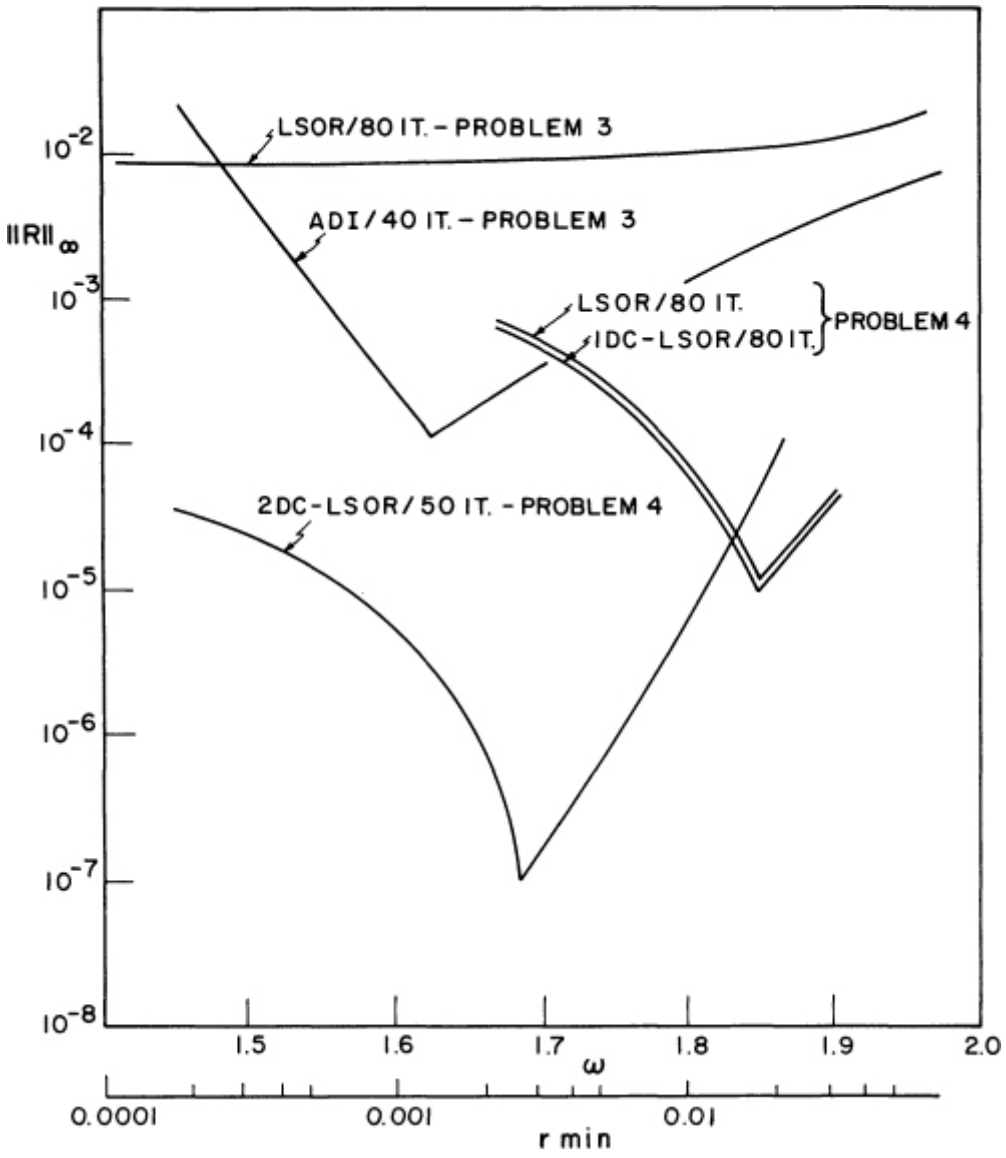


FIG. 8.25. Determination of optimum iteration parameters by trial and error using plots of $\|\boldsymbol{R}\|$ for a fixed number of iterations.

(h) The initial guess is also important for convergence. A bad initial guess can increase the number of iterations in the initial stage before the asymptotic behaviour is reached. Sensitivity of SIP to the initial guess was reported by Traylor and Sheffield (1971). Sensitivity of LSOR to the initial guess and/or a bad selection of direction of sweep may be observed by solving Exercise 8.4.

The initial guess is less important when the correction methods are used, because the application of a correction eliminates the average value of large initial error.

(i) Criteria for convergence are usually based on experience. Monitoring the convergence is usually accompanied by monitoring material balance (although material balance itself does not guarantee convergence). A practical approach of this type is

discussed by Traylor and Sheffield (1971). A more desirable approach is to relate convergence criteria to the truncation error of the solution since the true accuracy of the solution cannot be improved below the level of truncation errors. This approach (Brandt, 1977) has not been reported in the simulation literature to date.

8.4 COMPARISON OF ITERATIVE AND DIRECT METHODS

Direct as well as iterative methods have their advantages and disadvantages. The main practical disadvantage of direct methods is the large storage required, which increases more than proportionately with the number of equations. Also, direct methods may suffer from round-off errors for large multiphase problems. The main advantage of direct methods is their reliability. A direct method will solve a heterogeneous problem as easily as a homogeneous one, and the work and storage requirements will depend only on the method used. No trial runs are necessary to optimise the solution technique.

The main disadvantage of the iterative methods is their sensitivity to the problem being solved and to the iteration parameters (which are also problem dependent). For many difficult problems convergence of some iterative methods is so slow that their use becomes completely impractical. By a proper choice of an iterative method and associated iteration parameters, it is usually possible to solve a given problem iteratively. However, the processes of selection can be time consuming and expensive. On the other hand, iterative methods generally require very little storage in addition to the coefficients of the difference equations and therefore can be easily applied to very large systems.

Comparison of work for solving Problem No. 1, Section 8.3.9, by LSOR and direct methods is in Fig. 8.26. The problem was solved using different grid sizes ranging from 21×21 to 41×41, and the location of sources and sinks was interpolated to the closest grid point. Optimum ω was determined for each grid. The work reported is in CDC 6600 CPU seconds. The curve for elimination with *Natural Ordering* was derived from clocked times for D4 ordering and Fig. 8.8. Note that for large $I = J$ the work for direct methods approaches $O(I^4)$ while for iterative methods it is only $O(I^3)$. Therefore, direct elimination with D4 ordering is faster for smaller grids and LSOR is faster for larger grids. The cross-over point depends on the accepted tolerance level. For a tolerance of 10^{-5}, LSOR becomes faster for approximately a 58×58 grid, while for a tolerance of 10^{-4} it becomes competitive with D4 ordering for a 44×44 grid. Figure 8.25 also shows that for the problem considered Natural Ordering is never competitive with LSOR. This figure could also be used to estimate relative behaviour of iterative and direct methods for other problems presented in Section 8.3.9.

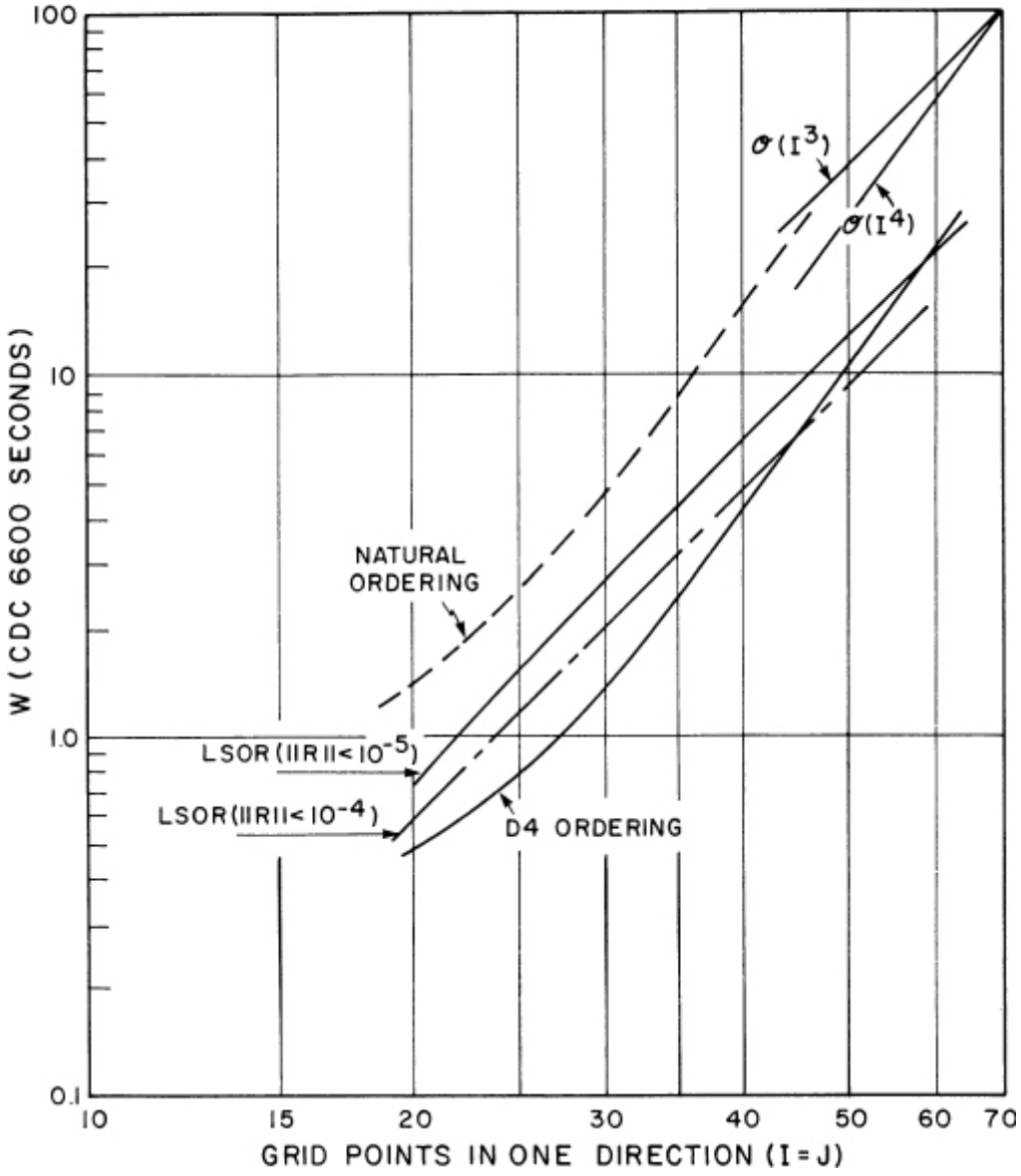


FIG. 8.26. Comparison of direct methods (D4 and natural ordering) with LSOR.

Some results of Price and Coats (1974) to be presented in Chapter 11 (Section 11.2.5) also indicate a cross-over point between iterative and direct methods. For typical reservoir simulation problems when $I > J$, direct elimination with D4 ordering will be competitive with iterative methods provided $J > 38$ for full systems with no inactive blocks. When the number of inactive blocks is substantial, the D4 scheme can compete with iterative methods for much larger values of J (up to about 70).

Woo *et al*. (1973) and Price and Coats (1974) have presented some limited comparisons of direct and iterative methods for practical reservoir problems. The comparisons are difficult to generalise because the work of a given iterative method depends upon the number of iterations, which in turn depends upon

(1) acceptable tolerance for convergence,

(2) nature of the problem (heterogeneity, shape, etc.), and
(3) for time-dependent problems on the time step size.

In general, larger time steps decrease the diagonal dominance of the matrix and therefore result in an increase in the number of iterations required. Also, the number of iterations depends on the change of the solution between time steps. For example, in reservoirs with pressure maintenance, the number of iterations per time step will decrease significantly after the initial transient period and the same effect will be apparent after each rate change.

All of the comparisons presented by Price and Coats (1974) are for three-dimensional problems and will be discussed in Chapter 11.

Woo *et al*. (1973) have considered several reservoir simulation problems of varying difficulty and applied several iterative and direct methods to them. The problems considered included areal, cross-sectional and coning simulations. The number of equations solved in various cases ranged from 100 to 2500. The direct methods were used with natural ordering and D4 ordering. They also applied a pseudo-optimum ordering scheme to both natural ordering and D4 ordering. This optimum ordering scheme is based on a modification of the Markowitz (1957) criterion and it involves pivoting along the diagonals. The iterative methods considered were LSOR, LSOR with IDC and SIP. Important conclusions based on their work are:

1. D4 ordering can be significantly improved for nearly square grid by the use of modified Markowitz criterion.
2. Sparse matrix techniques are more reliable and generally faster than iterative methods considered.

Other comparisons of iterative and direct methods are available in the literature; however, their results are difficult to compare because of different testing conditions. As an example, Brandon (1974) found that for a well-behaved parabolic problem, direct elimination and SIP were best for a 5×5 grid, but Gauss–Seidel with alternating direction of sweep was best for a 15×15 grid.

In practice, direct methods are always preferred for small to medium sets of equations, resulting from 2-D cross-sectional or radial grid. Iterative methods become eventually faster for large matrices resulting from large areal or 3-D problems (to be considered in Chapter 11). For such problems, direct methods also pose serious storage problems.

8.5 CONCLUDING REMARKS

The literature on the solution of matrix equations is already vast and rapidly expanding. In this chapter we have discussed in detail only the methods that have been successfully applied to reservoir simulation. Many of these and other methods are discussed in books edited by Reid (1971), Rose and Willoughby (1972), and Bunch and Rose (1976).

The largest current research effort is in the area of direct methods, which involve revival of the'tearing' method of Kron (1963). (Harary, 1971; Ledet and Himmelblau, 1970.) Methods of this type, also called 'marching methods', are fast (for example, the method proposed by Shacham and Kehat (1976) requires work of $O(I^3)$ for $I = J$), but they are unconditionally unstable with respect to round-off errors (Bank, 1976). Eisenstat *et al*. (1976) discuss a method of increasing the efficiency of Gaussian elimination by taking advantage of zeros within the envelope.

In the area of iterative methods, Nicolaides (1975) and Brandt (1977) suggested multilevel iterative methods which are generalisations of the correction methods discussed in Section 8.3.5. Work for these methods is claimed to be optimal (i.e., proportional to the number of unknowns).

Research is also being carried out in semi-iterative methods, conjugate gradients (Vinsome, 1976) and SIP-related methods (Letkeman, 1976).

EXERCISES

Exercise 8.1

Derive the work requirements for standard Gaussian elimination for matrix **A** *when*

(a) **A** *is a full matrix*
(b) **A** *is a five-diagonal matrix resulting from 2-D finite difference equations.*

Solution Outline

(a) The number of multiplications or divisions to eliminate the first column is

$$(N-1)N + (N-1) = N^2 - 1$$

For the second column

$$(N-2)(N-1) + (N-2) = (N-1)^2 - 1$$

etc. The total work for the forward pass is therefore

$$W_F = \sum_{i=1}^{N} (i^2 - 1) \tag{A}$$

Since

$$\sum_{i=1}^{I} i^2 = \frac{I(I+1)(2I+1)}{6} \tag{B}$$

the work is

$$W_F = \frac{N(N+1)(2N+1)}{6} - N \tag{c}$$

The work for the backward substitution is

$$W_B = \sum_{i=1}^{N} i = N(N+1)/2 \tag{D}$$

Therefore the total work is

$$W = \frac{N(N+1)(2N+1)}{6} + \frac{N(N-1)}{2} \tag{E}$$

(b) For a banded matrix, the work to eliminate and back substitute the last M rows is

$$W_M = \sum_{i=1}^{M} (i^2 + i - 1) = \frac{M(M+1)(2M+1)}{6} + \frac{M(M-1)}{2}$$

and for the rest

$$W = \sum_{i=1}^{N-M} [(M+1)^2 + M] = (N-M)[(M+1)^2 + M]$$

Therefore the total work is

$$W = \frac{M(M+1)(2M+1)}{6} + \frac{M(M-1)}{2} + (N-M)[(M+1)^2 + M] \qquad \text{(F)}$$

This formula ignores the fact that during elimination of the first M rows there is an incomplete fill-up. If this is taken into account, the work is slightly lower, i.e.,

$$W = (N-2M+1)[(M+1)^2 + M] + \frac{M(M+1)(2M+1)}{3} \qquad \text{(G)}$$
$$+ M(M-1) + (M+2)^2 - M - 9$$

This formula is given by Price and Coats (1974) (slightly incorrectly).

Exercise 8.2

(a) *Formulate an algorithm for the symmetric decomposition of a banded matrix with an arbitrary band-width.*

(b) *Derive the work required and compare with the work for standard elimination as a function of band-width.*

Solution Outline

(a) For a symmetric matrix **S** of band-width $B = 2M + 1$, $\mathbf{S} = \mathbf{W}\mathbf{W}^{\mathbf{T}}$
where

$$\mathbf{W} = \begin{bmatrix} w_{11} & & & & & & & \\ w_{21} & w_{22} & & & & & & \\ w_{31} & w_{32} & w_{33} & & & & & \\ . & . & & . & & & & \\ w_{M+1,1} & w_{M+1,2} & & & . & & & \\ . & & & & & . & & \\ . & . & . & . & . & . & w_{NN} & \end{bmatrix}$$

$$\mathbf{W}^{\mathrm{T}} = \begin{bmatrix} w_{11} & w_{21} & w_{31} & . & w_{M+1,1} & . & . \\ & w_{22} & w_{32} & & & & . \\ & & w_{33} & & & & . \\ & & & . & & & . \\ & & & & . & & . \\ & & & & & . & . \\ & & & & & & w_{NN} \end{bmatrix}$$

By comparison of terms, the elements of **W** are

$$w_{ij} = \frac{1}{w_{jj}}\left(s_{ij} - \sum_{r=J}^{j-1} w_{ir}w_{jr}\right) \qquad \begin{matrix} J = \max(1, j - M) \\ j = I, \ldots, i-1 \end{matrix} \tag{A}$$

$$w_{ii} = \left(s_{ii} - \sum_{r=I}^{i-1} w_{ir}^2\right)^{1/2} \qquad I = \max(1, i - M) \tag{B}$$

for all $i = 1, \ldots, N$.

(b) For a full matrix, the work to obtain elements of ith row by (A) and (B) is

$$i + (i-1) + (i-2) + \cdots = \frac{i(i+1)}{2} = \frac{1}{2}(i^2 + i)$$

and the work for back substitution is $2i$, since we have to solve $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{g} = d$ and $\boldsymbol{W}\boldsymbol{u} = \boldsymbol{g}$.

Total work is therefore

$$W_s = \frac{1}{2}\sum_{i=1}^{N} i^2 + \frac{1}{2}\sum_{i=1}^{N} i + 2\sum_{i=1}^{N} i \qquad \text{(C)}$$
$$= \frac{1}{2}\left[\frac{N(N+1)(2N+1)}{6} + \frac{1}{2}N(N+1)\right] + N(N+1)$$

For a banded matrix, the work to obtain first M rows is

$$\sum_{i=1}^{M} \frac{1}{2} i(i+1)$$

for remaining entries

$$\sum_{i=M+1}^{N} \frac{1}{2}(M+1)(M+2)$$

and the work for back substitution is twice

$$\sum_{i=1}^{M} i + (N-M)(M+1)$$

Therefore

$$W_s = \frac{1}{2}\left[\frac{M(M+1)(2M+1)}{6} + \frac{1}{2}M(M+1) + (N-M)(M+1)(M+2)\right] \qquad \text{(D)}$$
$$+ M(M+1) + 2(N-M)(M+1)$$

Comparison of (C) and (D) with the standard ordering [eqns. (E) and (F) of Exercise 8.1] shows that $W_s/W \to ½$ as $N \to \infty$. For a fixed N, the symmetric decomposition becomes more efficient with increasing M, as illustrated in the following Table for $N = 10$ and $N = 100$:

M	$N = 10$		$N = 100$	
	W	W_s	W	Ws

1	46	66	496	696
2	94	106	1084	1186
3	150	148	1860	1768
5	270	230	3960	3200
10			12220	8250
20			39940	23800

Exercise 8.3

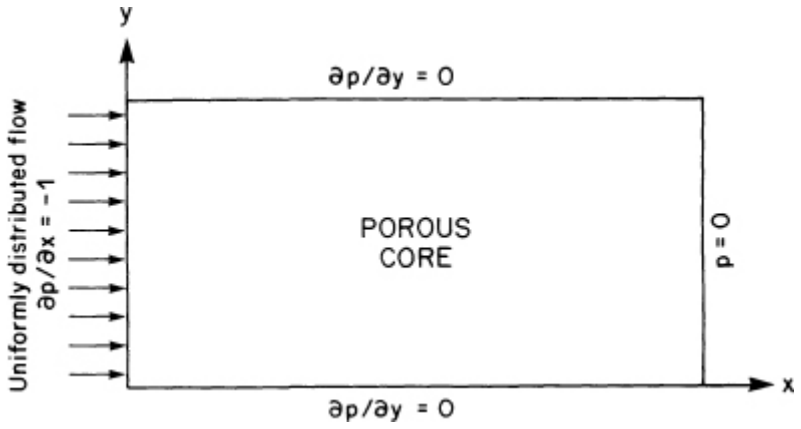
Find the eigenvalues of the point Jacobi matrix

$\mathbf{B_J = D^{-1}(L + H)}$ (*Young*, 1971; *Section* 4.6).

Exercise 8.4

Consider steady-state two-dimensional flow in the system shown below, described by:

$$k_x \frac{\partial^2 p}{\partial x^2} + k_y \frac{\partial^2 p}{\partial y^2} = 0 \qquad \begin{matrix} 0 \le x \le L_x \\ 0 \le y \le L_y \end{matrix} \tag{A}$$



Solve the matrix equation by LSOR with different initial guesses and different direction of lines and sweep. Consider the case of $k_x \approx k_y$ *and* $k_x \gg k_y$, $L_x = L_y$ *and* $L_x \gg L_y$ *and combinations of these.*

CHAPTER 9

MULTIPHASE FLOW IN TWO DIMENSIONS

9.1 INTRODUCTION

This chapter deals with multiphase flow in two dimensions. Since all important formulations and solution techniques required for reservoir simulation have already been developed in Chapter 5, this chapter will deal mainly with some practical topics related to two-dimensional (2-D) models. Some of the new ideas to be discussed here include treatment of production and injection, simulation of aquifers, and wellbore-reservoir interaction in coning problems. These topics are more typical of multidimensional problems and could not be discussed in a one-dimensional (1-D) setting. Because simulation of problems in 2-D is very common in practice, there are many aspects related to reservoir engineering that are associated with simulation of real 2-D problems. In the extreme, one could say that almost every reservoir engineering problem has some unique feature which should be reflected in the numerical model for that problem. It is, however, not the objective of this chapter (and this book in general) to cover this area, except for some special topics which are essential from the point of view of numerical computations. Some discussion of these topics is included in Chapter 12.

9.2 CLASSIFICATION OF 2-D PROBLEMS

Since all real physical reservoirs are three dimensional, every formulation of the mathematical problem in two dimensions represents a simplification (this is of course even more true of one-dimensional formulation). It is important to understand clearly the assumptions made in each case so that the limitations of the model may be appreciated.

Simulation of reservoirs in 2-D is quite common. In many instances, 2-D models will give adequate information, and the cost for a 2-D as opposed to a 3-D simulation is much lower. As discussed in Chapter 7, there are three basic types of 2-D models. We will now describe these models for the multiphase case.

9.2.1 Areal Problems (x–y)

Most reservoirs have a thickness which is small compared to their areal extent. Such reservoirs look like 'blankets' (Coats *et al*., 1971) and it is natural to represent them by a two-dimensional areal grid, schematically shown on Fig. 9.1. Areal models cannot simulate flow in the vertical direction and must assume *uniform properties with no flow in the z-direction*. The 3-D character of the problem is, however, partially retained in the model by defining the thickness Δz and elevation h as functions of x and y. All properties and variables such as S and p are also functions of x and y only and therefore must represent integral averages (averaged over the thickness of the formation Δz).

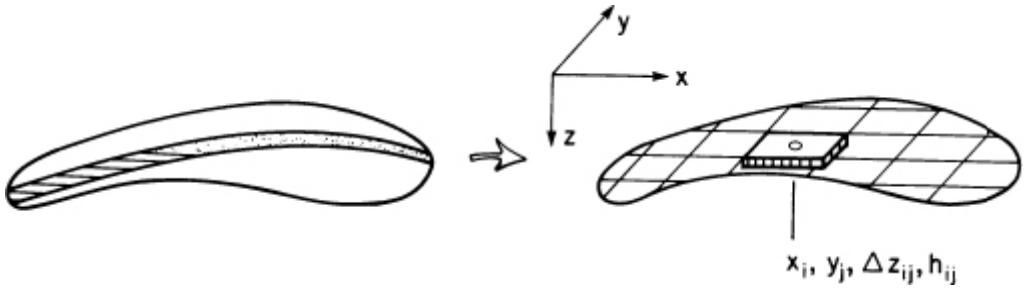


FIG. 9.1. Representation of a reservoir by an areal grid.

With this in mind, we can write the equations for two-phase flow as

$$\frac{\partial}{\partial x}\left[\Delta z \lambda X_l\left(\frac{\partial p_l}{\partial x} - \gamma_l \frac{\partial h}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\Delta z \lambda Y_l\left(\frac{\partial p_l}{\partial y} - \gamma_l \frac{\partial h}{\partial y}\right)\right] = \left[\frac{\partial}{\partial t}\left(\phi \frac{S_l}{B_l}\right) + q_l\right]\Delta z \qquad l = \mathrm{w}, \mathrm{n} \tag{9.1}$$

where

$$S(x,y) = \left[\int_0^{\Delta z} S(x,y,z)\,\mathrm{d}z\right]/\Delta z, \quad \text{etc.}$$

$$\lambda X_l = \frac{k_x k_{rl}}{\mu_l B_l} \qquad \text{and} \qquad \lambda Y_l = \frac{k_y k_{rl}}{\mu_l B_l}$$

The assumption of uniform properties (complete vertical mixing) is valid, if the thickness Δz is small compared to the thickness of the transition zone. When this is not the case, the rock properties k_{rl} and P_c must be modified according to the degree of vertical segregation of fluids in order to simulate correctly the flow between blocks. The methods of calculating these pseudo-functions (also called 'vertical equilibrium' or

VE functions) which are different from laboratory rock properties, are discussed in Chapter 12.

Typical uses of areal models are simulation of 3-D problems with VE and study of areal sweep.

9.2.2 Cross-sectional Problems (x-z)

This type of model can represent a reservoir, where vertical flow is important, but the horizontal flow is predominantly in one direction (Fig. 9.2) which we choose to be the x-direction. We *neglect the flow in the y-direction* and account for the variation of properties in the y-direction by the definition of thickness Δy and other variables as functions of x and z. This means that all properties and variables are now averages in the y-direction, and the partial differential equations for flow can be written as

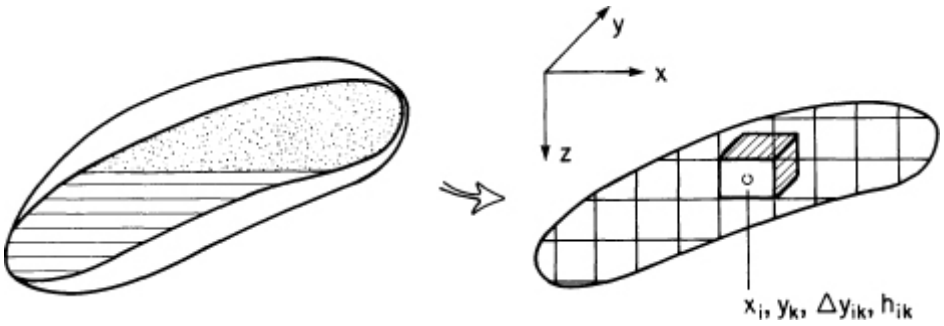


FIG. 9.2. Representation of a reservoir by a cross-sectional grid.

$$\frac{\partial}{\partial x}\left[\Delta y \lambda X_l\left(\frac{\partial p_l}{\partial x} - \gamma_l \frac{\partial h}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[\Delta y \lambda Z_l\left(\frac{\partial p_l}{\partial z} - \gamma_l \frac{\partial h}{\partial z}\right)\right] = \left[\frac{\partial}{\partial t}\left(\phi \frac{S_l}{B_l}\right) + q_l\right]\Delta y \qquad l = \mathrm{w}, \mathrm{n} \tag{9.2}$$

Note that in this case the vertical segregation of the fluids can be correctly simulated and therefore the functions k_{rl} and P_c should be the true rock properties. However, when the thickness of the transition zone is small compared to the vertical block size, pseudo-functions within the interval Δz should be considered (Chapter 12).

A typical use of a cross-sectional formulation is for generating pseudo-functions for use in areal models and for the study of vertical sweep efficiency of displacement processes.

9.2.3 Single-Well Problems (r–z)

Saturations and pressures change most rapidly around wells. It is often desired to obtain a detailed solution of the movement of water and/or gas towards producing wells. Solution techniques for single-well problems were developed primarily for this application and these problems are usually called ‘coning problems’.

Because the flow near a well is always three-dimensional in Cartesian co-ordinates, it cannot be approximated by either of the 2-D formulations discussed above. The natural approach here is to assume that the flow is symmetrical around the wellbore, and use cylindrical co-ordinates with the z-axis being identical with the axis of the well (Fig. 9.3).

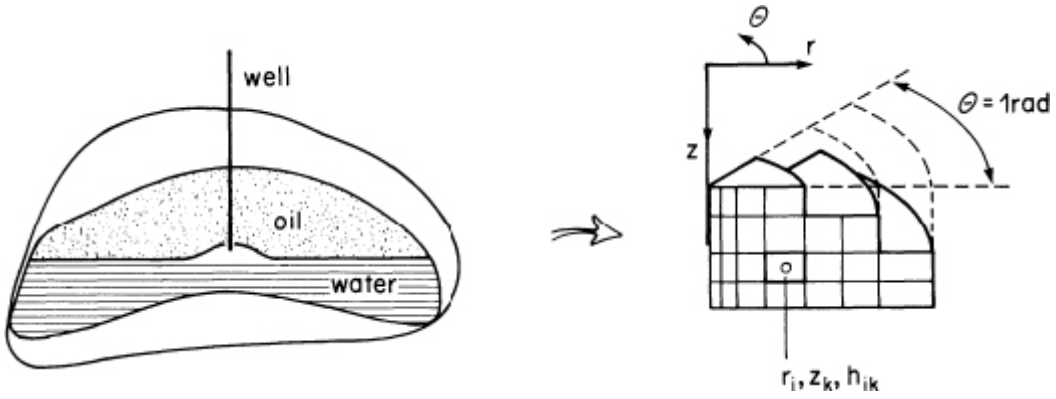


FIG. 9.3. Representation of a reservoir by a grid in cylindrical co-ordinates.

This model assumes that the solution domain is axisymmetric and all rock properties as well as boundary conditions are functions of *r, z* and *t* only. Furthermore, the initial conditions (and the gravity forces) must also maintain this symmetry which implies that the *z-axis must be vertical*. After the transformation of equations in cylindrical co-ordinates (see Chapter 7, Section 7.10), we obtain

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\lambda R_l\left(\frac{\partial p_l}{\partial r}-\gamma_l\frac{\partial h}{\partial r}\right)\right]+\frac{\partial}{\partial z}\left[\lambda Z_l\left(\frac{\partial p_l}{\partial z}-\gamma_l\frac{\partial h}{\partial z}\right)\right] \tag{9.3}$$

$$=\frac{\partial}{\partial t}\left(\phi\frac{S_l}{B_l}\right)+q_l \qquad l=\mathrm{w},\mathrm{n}$$

Equation (9.3) is written for an angular sector of $\theta = 1$ radian and must be multiplied by 2π to obtain the correct volumes. Furthermore, all properties must be averaged in the angular direction.

As a rule, coning models simulate only a small part of a reservoir (equivalent to a few blocks of an areal grid) and the communication of the selected part with the rest of

the reservoir must be represented by boundary conditions (flow across external boundaries).

Typical uses of single-well models include simulation of water coning, gas coning and well testing.

9.2.4 General Comments

(a) In writing the flow equations, it was assumed that the permeabilities k_x, k_y, k_z are the principal axes of the permeability tensor. In the single-well case, use of k_r also implies $k_x = k_y = k_r$ (cf. Chapter 2, Section 2.3.1).

(b) We have retained gravity terms in certain cases where they would normally be zero, e.g. $\gamma_l(\partial h/\partial x)$ in eqn. (9.2) and $\gamma_l(\partial h/\partial r)$ in eqn. (9.3). This is because these terms may be non-zero in general curvilinear or inclined co-ordinate systems.

9.3 METHODS OF SOLUTION AND THEIR COMPARISON

9.3.1 Discretisation in 2-D

The methods for discretising and solving the equations developed in 1-D in Chapter 5 can be directly applied to 2-D and 3-D problems. The only difference is in the additional terms which include the flow in the additional directions being considered.

Consider the 2-D equations (9.1). A typical finite-difference element is shown in Fig. 9.4.

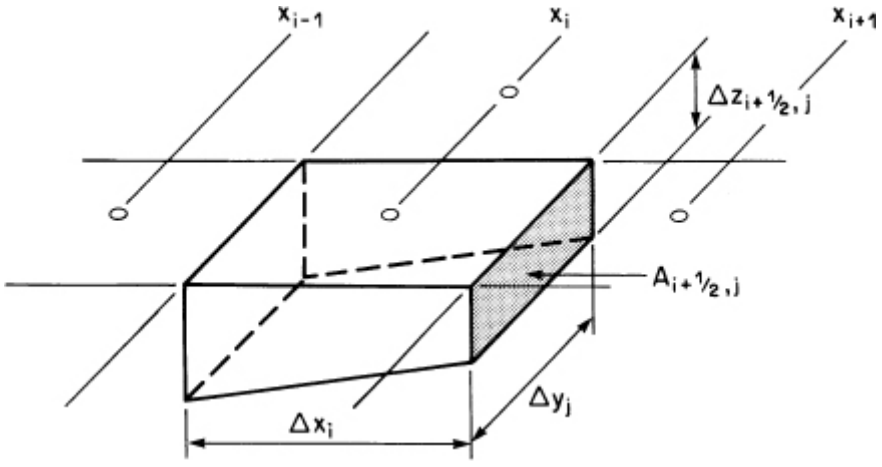


FIG. 9.4. A typical element of an areal grid.

Note that Δx_i is defined as:

$$\Delta x_i = \tfrac{1}{2}[x_{i+1} - x_i) + (x_i - x_{i-1})] = \tfrac{1}{2}[\Delta x_{i+1/2} + \Delta x_{i-1/2}] \tag{9.4}$$

and a similar definition is used for Δy (Chapter 3, Section 3.5).

The flow term in the x-direction is now discretised in the same way as for a 1-D case (eqn. (5.6)). After multiplying the equation by $\Delta x_i \, \Delta_{yj}$ we obtain

$$\begin{aligned} \Delta x_i \, \Delta y_j \frac{\partial}{\partial x}\left[\lambda X_l\left(\frac{\partial p_l}{\partial x} - \gamma_l \frac{\partial h}{\partial z}\right)\right] &\simeq \Delta_x \mathrm{T} X_l \, \Delta_x (p_l - \gamma_l h) \\ &\equiv \mathrm{T} X_{l(i+1/2),j}[p_{li+1} - p_{li} - \gamma_{li+1/2}(h_{i+1} - h_i)]_j \\ &\quad + \mathrm{T} X_{l(i-1/2),j}[p_{li-1} - p_{li} - \gamma_{li-1/2}(h_{i-1} - h_i)]_j \end{aligned} \tag{9.5}$$

The transmissibility for phase l is

$$\mathrm{T} X_{l(i+1/2),j} = \lambda X_{(i+1/2),j} \frac{A_{(i+1/2),j}}{\Delta x_{i+1/2}} = \lambda X_{(i+1/2),j} \frac{\Delta y_j \, \Delta z_{(i+1/2),j}}{\Delta x_{i+1/2}} \tag{9.6}$$

The y-direction terms are analogous:

$$\begin{aligned} \Delta_y \mathrm{T} Y_l \, \Delta_y (p_l - \gamma_l z) &\equiv \mathrm{T} Y_{li,j+1/2}[p_{lj+1} - p_{lj} - \gamma_{lj+1/2}(h_{j+1} - h_j)]_i \\ &\quad + \mathrm{T} Y_{li,j-1/2}[p_{lj-1} - p_{lj} - \gamma_{lj-1/2}(h_{j-1} - h_j)]_i \end{aligned} \tag{9.7}$$

where

$$\mathrm{T} Y_{li,j+1/2} = \lambda Y_{i,j+1/2} \frac{\Delta x_i \, \Delta z_{i,j+1/2}}{\Delta y_{j+1/2}} \tag{9.8}$$

Now the finite-difference equation may be written in a compact form as

$$\begin{aligned} [\Delta \mathrm{T}_l \, \Delta (p_l - \gamma_l h)]_{ij} &= [\Delta_x \mathrm{T} X_l \, \Delta_x (p_l - \gamma_l h) + \Delta_y \mathrm{T} Y_l \, \Delta_y (p_l - \gamma_l h)]_{ij} \\ &= \frac{V_{ij}}{\Delta t} \Delta_t \left(\phi \frac{S_l}{B_l}\right)_{ij} + Q_{lij} \qquad l = \mathrm{w}, \mathrm{n} \end{aligned} \tag{9.9}$$

where the associated block volume is

$$V_{ij} = \Delta z_{ij} \, \Delta x_i \, \Delta y_j \tag{9.10}$$

and Q_{ij} is the total source term for the block.

The finite-difference equations for eqn. (9.3) are formally the same. As discussed at length in Chapter 3 (Section 3.6) the block boundaries for calculating the volume are chosen by (Fig. 9.5):

$$r_{i+1/2} = \left(\frac{r_{i+1}^2 - r_i^2}{\ln (r_{i+1}^2/r_i^2)} \right)^{1/2}$$

$$z_{k+1/2} = \tfrac{1}{2}(z_k + z_{k+1})$$

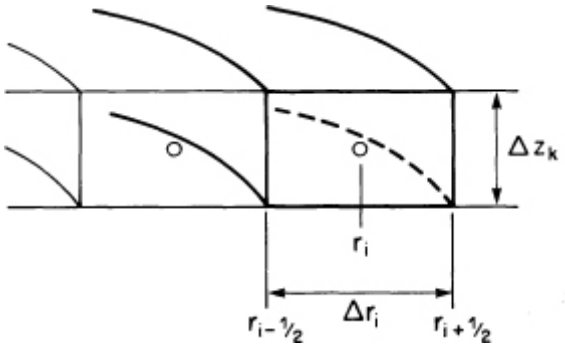


FIG. 9.5. A typical element of cylindrical grid.

and the transmissibilities T*R* and T*Z* for this case are

$$\mathrm{T}R_{l(i+1/2),k} = 4\pi \frac{r_{i+1/2}^2 \, \Delta z_k}{r_{i+1}^2 - r_i^2} \lambda R_{l(i+1/2),k} \tag{9.11}$$

$$\mathrm{T}Z_{li,k+1/2} = \pi \frac{r_{i+1/2}^2 - r_{i-1/2}^2}{\Delta z_{k+1/2}} \lambda Z_{li,k+1/2} \tag{9.12}$$

The volume of the block is

$$V_{ik} = \pi \, \Delta z_k (r_{i+1/2}^2 - r_{i-1/2}^2) \tag{9.13}$$

It should now be apparent that the only difference between eqns. (5.5) and (9.9) is in the left-hand-side terms. Equation (9.9) can be written in any of the matrix forms derived in Chapter 5 with the only difference being the form of matrix **T** (and **T**′ for implicit transmissibility methods). For the SS method, matrix **T** will be *block-five-diagonal*, while for the IMPES method **T** will be five diagonal. It is easy to see that the results on existence and uniqueness of solution (Chapter 5, Section 5.4.2) are equally valid in 2-D. This follows from the symmetry of coefficients. However, the stability limits will be different.

9.3.2 Stability of SS and IMPES in 2-D

Stability with respect to P_c. Let us derive the stability limit of IMPES for incompressible flow. In the linearised analysis, we can again write the equivalent of eqn. (5.43) as:

$$\mathrm{T_w}\,\Delta^2 p^{n+1} = \frac{V_p}{\Delta t}\Delta_t S_\mathrm{w} + \mathrm{T_w} P_c'\,\Delta^2 S_\mathrm{w}^n + Q_\mathrm{w} \tag{9.14a}$$

$$\mathrm{T_n}\,\Delta^2 p^{n+1} = -\frac{V_p}{\Delta t}\Delta_t S_\mathrm{w} + Q_\mathrm{n} \tag{9.14b}$$

where now

$$\mathrm{T}_l\,\Delta^2 p \equiv \mathrm{T}X_l\,\Delta_x^2 p + \mathrm{T}Y_l\,\Delta_y^2 p \tag{9.15a}$$

$$\mathrm{T_w} P_c'\,\Delta^2 S_\mathrm{w} \equiv \mathrm{T}X_\mathrm{w} P_c'\,\Delta_x^2 S_\mathrm{w} + \mathrm{T}Y_\mathrm{w} P_c'\,\Delta_y^2 S_\mathrm{w} \tag{9.15b}$$

With this notation, we can again write equations for errors e_1, e_2 in the form of eqn. (5.44). Combination of these equations gives

$$\mathrm{T_n}\,\Delta^2 e_2^{n+1} + \mathrm{T_w}\,\Delta^2 e_2^{n+1} = \mathrm{T_w} P_c'\,\Delta^2 e_1^n \tag{9.16}$$

We now seek a solution in the form

$$e_{kij}^n = \xi_{kij}^n \exp(\sqrt{-1}\alpha_{xk} i)\exp(\sqrt{-1}\alpha_{yk} j) \qquad k = 1, 2$$

where

$$\alpha_{xk} = m_{xk}\,\Delta x \qquad \alpha_{yk} = m_{yk}\,\Delta y \qquad \text{for} \quad m_{xk}, m_{yk} > 0$$

Then by suppressing subscripts 'i' and 'j' we can write

$$\mathrm{T}_l\,\Delta^2 e_2 = -\gamma_{l2} e_2 \qquad l = \mathrm{w}, \mathrm{n} \qquad \mathrm{T_w}\,\Delta^2 e_1 = -\gamma_{\mathrm{w}1} e_1$$

where now γ_{lk} is a sum given by

$$\gamma_{lk} = \mathrm{T}X_l 4\sin^2\frac{\alpha_{xk}}{2} + \mathrm{T}Y_l 4\sin^2\frac{\alpha_{yk}}{2} \qquad l = \mathrm{w}, \mathrm{n} \qquad k = 1, 2 \tag{9.17}$$

Substitution into eqn. (9.16) gives the generalisation of eqn. (5.46):

$$e_2^{n+1} = P_c'[\gamma_{w1}/(\gamma_{w2} + \gamma_{n2})]e_1^n \tag{9.18}$$

Substitution of this in the second equation (5.44)

$$\gamma_{n_2} e_2^{n+1} = -\frac{V_p}{\Delta t}(e_1^{n+1} - e_1^n)$$

gives the amplification factor for the error e_1 as

$$\xi_1 = P_c'\gamma_{w1}\frac{\gamma_{n2}}{\gamma_{w2} + \gamma_{n2}}\frac{\Delta t}{V_p} - 1$$

The most severe restriction on Δt will result when all ($\sin^2$) terms in eqn. (9.17) are close to 1 (Coats, 1968). Then the condition $|\xi| < 1$ yields

$$\Delta t < \frac{1}{2}\frac{V_p}{|P_c'|\dfrac{\Sigma T_w \Sigma T_n}{(\Sigma T_w + \Sigma T_n)}} \tag{9.19}$$

where

$$\Sigma T_l = TX_l + TY_l$$

which is the expected generalisation of eqn. (5.47). This equation is a special case of the general condition derived by Coats. An important consequence of (9.19) is that when Δx and Δy are different, *the stability is controlled by the smaller grid size.*

To see this, rewrite (9.19) using (9.6) and (9.8) as

$$\Delta t < \frac{1}{2}\frac{V_p}{|P_c'|}\left(\frac{1}{\Sigma T_w} + \frac{1}{\Sigma T_n}\right) \leq \frac{1}{2}\frac{V_p}{\Delta z\left(k_x\dfrac{\Delta y}{\Delta x} + k_y\dfrac{\Delta x}{\Delta y}\right)}\min\frac{\left(\dfrac{\mu_w}{k_{rw}} + \dfrac{\mu_n}{k_{rn}}\right)}{|P_c'|} \tag{9.20}$$

Equation (9.20) applies for flow in the x–y system. Its counterpart for a cross-sectional problem can be obtained by the interchange of Δy and Δz:

$$\Delta t < \frac{1}{2}\frac{V_p}{\Delta y\left(k_x\dfrac{\Delta z}{\Delta x} + k_z\dfrac{\Delta x}{\Delta z}\right)}\min\frac{\left(\dfrac{\mu_w}{k_{rw}} + \dfrac{\mu_n}{k_{rn}}\right)}{|P_c'|} \tag{9.21}$$

Assume now that we want to simulate performance of a reservoir by an areal model with block dimensions Δx and Δy_A, and thickness Δz; and with a cross-sectional model with block dimensions Δx and Δz_c, and thickness Δy. Usually, $\Delta x \simeq \Delta y$, but $\Delta z \ll \Delta x$. If $k_x \simeq k_y = k$, we get from (9.20) and (9.21)

$$R = \frac{\Delta t_{\text{cross}}}{\Delta t_{\text{areal}}} = 2\frac{k\,\Delta z_C^2}{k_z\,\Delta x\,\Delta y_A}\frac{K}{J} \tag{9.22}$$

where K is the number of vertical blocks in the cross-sectional model and J is the number of blocks in the y-direction of the areal model. We have assumed that $\Delta z = K\,\Delta z_c$ in (9.20) and $\Delta y = K\,\Delta y_A$ in (9.21).

The ratio R is often very small. For example, if $\Delta z = 10$ ft, $\Delta x = \Delta y = 1000$ ft, $k = 10k_z$ and $K = J$, we obtain $R = 0{\bullet}002$. This leads to the following comment:

> *The stability limit which is not important for an areal model may force very small time steps in cross-sectional* (*or* 3-*D*) *models.*

This limit on stability is a serious shortcoming of the IMPES method. On the other hand, the SS method is unconditionally stable with respect to P_c.

The generalisation of the 1-D case for three-phase flow is accomplished by defining

$$\Sigma T_l = T_o + T_w + T_g$$

in (9.19).

Stability with respect to transmissibilities. It is possible to readily extend 1-D results. We start with the 2-D form of the saturation equation (5.58):

$$-\frac{df_w}{dS_w}\left[Q_x(S_{wi} - S_{wi-1})_j^n + Q_y(S_{wj} - S_{wj-1})_i^n\right] = \frac{V_p}{\Delta t}(S_w^{n+1} - S_w^n)_i \tag{9.23}$$

where Q_x and Q_y are related to total Darcy velocities

$$Q_x = u_x\,\Delta z\,\Delta y \qquad Q_y = u_y\,\Delta z\,\Delta x$$

The Fourier analysis given in Chapter 5 (Section 5.4) can be repeated again with the result that

$$C = \frac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}}\frac{\Delta t}{V_p}(Q_x + Q_y) < 1 \tag{9.24}$$

This gives the following stability condition:

$$\Delta t \leq \frac{V_p}{\dfrac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}}(Q_x + Q_y)} \tag{9.25}$$

The above result could be expressed in terms of the rate of advance of the flood front. In analogy with the 1-D case, we assume that the velocity of a constant saturation line in the x–y plane is given by

$$u_{xs} = \left(\frac{\partial x}{\partial t}\right)_{S_{\mathrm{w}}} = \frac{u_x}{\phi}\frac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}} \qquad u_{ys} = \left(\frac{\partial y}{\partial t}\right)_{S_{\mathrm{w}}} = \frac{u_y}{\phi}\frac{\mathrm{d}f_{\mathrm{w}}}{\mathrm{d}S_{\mathrm{w}}}$$

Then we can rewrite (9.25) as

$$\Delta t\left(\frac{u_{xs}}{\Delta x} + \frac{u_{ys}}{\Delta y}\right) = \frac{l_x}{\Delta x} + \frac{l_y}{\Delta y} \leq 1 \tag{9.26}$$

where l_x and l_y are the distances advanced by the front in x- and y-direction. For comparison, the 1-D case can be written as

$$\frac{l_x}{\Delta x} \leq 1 \tag{9.27}$$

It is obvious that *the stability limit in 2-D is always lower than in the corresponding 1-D case*, and the fact can be easily demonstrated by stability comparison of 1-D radial and 2-D single-well simulators. Because Δz is usually small, stability limit in 2-D may be drastically reduced if considerable vertical flow occurs (Section 5.5 of Chapter 5).

9.3.3 Comparison of Various Solution Methods and Computer Requirements

There are four basic types of solution methods: IMPES, explicit (with respect to transmissibilities) SS method, sequential (SEQ), and implicit SS methods.

The choice of the most efficient method for a given problem depends on two basic factors:

(a) The work necessary to obtain the solution of difference equations for one time step. This depends upon the grid size, the method itself and the algorithm used for solving the algebraic equations.
(b) The stability limit and therefore the maximum time step size that can be used for a given method. This depends upon the type of problem being solved. Note that accuracy requirements may also limit the maximum time step size.

9.3.3.1 *Work Per Time Step and Storage*

For comparison of work per time step, W_{TS}, consider 2-D rectangular grid of $I \times J$ grid points, and assume that the work is spent mostly in solving the matrix equation (this is true for finite-difference methods if I and J are sufficiently large). Let us first assume that the solution is obtained by direct elimination with the work estimate given by

$$W = CB^2N \tag{9.28}$$

where B is the band-width, N the total number of unknowns and C is a constant dependent on machine and program efficiency. Then we can compute the work for all four methods. This comparison for two-phase flow is presented in Table 9.1 and for three-phase flow in Table 9.2. We have assumed for both cases that $I \geq J$. The last column gives the work ratio in relation to the work for IMPES method. We can make several important observations:

(a) *The work depends strongly on the smaller dimension* (J). It is often convenient to compare the work per grid block and time step, which is $W_1 = W_{TS}/IJ$. Then $W_1 \simeq J^2$, i.e., a simulator will solve a problem on a 50×5 grid 4 times faster than on a 25×10 grid.

TABLE 9.1

COMPARISON OF WORK FOR 2-D, 2-PHASE FLOW

Method	*Band-width, B*	*No. of unknowns, N*	*Work per timestep,* W_{TS}	$W_{TS}/W_{TS(IMPES)}$
IMPES	J	IJ	CIJ^3	1
SEQ	J	IJ	$C(IJ^3 + IJ^3)$	2
SS	$2J$	$2IJ$	$8CIJ^3$	8

(b) The work for the SS method is largest and is the same for the explicit and implicit versions. Therefore, if an SS formulation is required because of P_c

instability, it can be made implicit without a significant increase in the running time.

(c) The sequential method (SEQ) is 3–4 times faster than the SS method. This is significant because the SEQ method eliminates P_c instability of IMPES.

(d) The work for IMPES is independent of number of phases, while for SS and SEQ it increases dramatically as we go from two- to three-phase problems. *Three-phase problems require* 3–4 *times more work per time step than two-phase problems when solved by the SS or SEQ methods.*

In general, the work for the SS method increases with the third power of the number of phases (differential equations being solved). For this reason, SS-type methods have so far found little use for solving more than three equations.

Similar analysis can be done for storage requirements. Again, the SS method is most demanding on storage and the requirement increases with the number of phases. If we assume that storage for the entire band is required, the approximate storage S required for two- and three-phase flow is in Tables 9.3 and 9.4.

TABLE 9.2

COMPARISON OF WORK FOR 2-D, 3-PHASE FLOW

Method	*Band-width, B*	*No. of unknowns, N*	*Work per timestep,* W_{TS}	$W_{TS}/W_{TS(IMPES)}$
IMPES	J	IJ	CIJ^3	1
SEQ	J, $2J$	IJ, $2IJ$	$C(IJ^3 + 8IJ^3)$	9
SS	$3J$	$3IJ$	$27CIJ^3$	27

TABLE 9.3

STORAGE REQUIREMENT FOR 2-D, 2-PHASE FLOW

Method	*Storage, S*	S/S_{IMPES}
IMPES	$2IJ^2$	1
SEQ	$2IJ^2$	1
SS	$8IJ^2$	4

The estimates of W_{TS} and S given here are based on some simplifying assumptions. They do not take into account symmetries and the use of techniques which are more efficient than standard elimination. Although the relative estimates of W_{TS} remain true for different ordering of equations, they will be different for large matrices in which the

work for an efficient direct or iterative method should approach CN^3 (Chapter 8). Nevertheless, our comparison illustrates the relationship of the three methods as far as work and storage requirements are concerned.

TABLE 9.4

STORAGE REQUIREMENT FOR 2-D, 3-PHASE

Method	*Storage, S*	S/S_{IMPES}
IMPES	$2IJ^2$	1
SEQ	$8IJ^2$	4
SS	$18IJ^2$	9

9.3.3.2 *Overall Efficiency*

The overall efficiency depends upon work required per time step and the total number of time steps necessary to reach the final solution, i.e., the total work is given by

$$W = W_{\text{TS}} \times \text{Number of time steps required}$$

The number of time steps required to simulate a given problem depends upon the stability of the method for the given problem. In general, it is impossible to predict the number of time steps required without some numerical experiments or experience with similar problems. Therefore, we can give only general guidelines here.

Most areal problems are tractable by the IMPES method. The speed and low-core requirements of the IMPES method allow the use of large numbers of blocks.

Cross-sectional problems with low P_c can be solved by the IMPES method. With large P_c or fine grid, the SEQ method is more efficient than the IMPES method. For most cross-sectional problems, the SEQ method will be sufficiently stable and therefore faster than the SS method. However, difficult problems such as gas injection in an aquifer or three-phase problems may require the SS method.

Finally, for single-well problems the linearised implicit SS method offers the best reliability. Some authors (Spillette *et al*., 1973) claim that the SEQ method can be used as a basis for general-purpose simulators. Our experience is that the stability of the SEQ method is slightly lower compared to the SS method for relatively ‘easy’ coning problems, but it can deteriorate considerably for very difficult problems. Also, the fact that the SEQ method cannot rigorously satisfy material balance may be objectionable in some cases. However, whenever applicable, the SEQ method will result in considerable savings over the SS method.

Finally, efficiency of simulation can be optimised by using the implicit formulation only where it is required and by solving equations explicitly elsewhere. This idea is relatively easy to implement for a combination of the SEQ and IMPES method (MacDonald and Coats, 1970), but implementation becomes difficult when the SS method is included (Seth, 1974).

9.4 BOUNDARY CONDITIONS

The question of proper representation of boundary conditions has received relatively little theoretical analysis in petroleum engineering literature. The boundary conditions typical for hydrology problems are discussed at length by Bear (1972).

In reservoir simulation, flow in and out of the system occurs only at the boundaries, such as the external boundaries of the reservoir and at the boundaries of the wells. The latter must be idealised as Dirac line or point sources in cross-sectional and areal simulations because the radius of the well is very small compared to the grid size. The well boundary, however, can be represented correctly in a single-well model. It is customary in reservoir simulation to represent flow across all boundaries by source/sink terms and replace the actual boundary conditions by homogeneous Neumann (no-flow) boundary conditions on the entire boundary. We have shown in Chapter 7 (Section 7.4.3) that this approach is equivalent, at the finite-difference level, to the discretisation of the original boundary conditions and allows us to treat all boundaries (i.e., external and well boundaries) in the same fashion. Therefore, the source terms in eqns. (9.1) to (9.3) will always be singular (Dirac) functions which will be defined in such a manner as to represent the true boundary conditions.

9.4.1 Differential Formulation

The boundary conditions for eqns. (9.1) to (9.3) are obtained as an extension of the single-phase case discussed in Chapter 7, Section 7.4.

For single-phase flow, specifying flow rate across the boundary gives a formulation with a unique solution. For multiphase flow, the situation is more complicated and flow rates of all phases must be specified, although they are not necessarily independent. As discussed in Chapter 7, there are two basic types of boundaries (see Figs. 7.7 and 7.8):

1. *Closed boundary* ($\Gamma 2$). There is no flow of any phase across a closed boundary; the product of Darcy velocity and the normal vector $\boldsymbol{n}$ vanishes:

$$q_l = \lambda_l(\nabla p_l - \gamma_l \nabla h)\,.\,\boldsymbol{n} = 0, \qquad l = \mathrm{o,w,g} \quad \text{on} \quad \Gamma_2 \tag{9.29}$$

where $\boldsymbol{n}$ is the normal to the boundary and $\lambda_l = kk_{rl}/\mu_l$ is the transmissibility in the direction of the normal. Note that the potential gradient must vanish only if the particular phase is mobile at the boundary ($\lambda_l > 0$). Equation (9.29) is written in terms of q at reservoir conditions since λ_l does not include the formation volume factor B_l.

2. *Boundary open to flow* (Γ_1 *and* (Γ_3). In this case, the flow rates q_l across the boundary are specified. The flow in reservoir units is

$$\lambda_l(\nabla p_l - \gamma_l \nabla h) \cdot \boldsymbol{n} = q_l(\Gamma) \qquad \Gamma = \Gamma_1, \Gamma_2 \tag{9.30}$$

On the boundary Γ_3 where fluids are injected, the flow rates for each phase are controlled and therefore *known*. In the case of real injection, usually only one phase is injected, or the composition of the mixture is known. Injection also often accounts for the influence of the parts of a reservoir beyond the chosen boundary Γ. In this case the injection boundary is usually divided into zones where only one phase is injected. In both cases the known quantity is the total rate Q_{Tl} for a given boundary Γ_3 representing a well or an injection zone. Therefore, while $q_l(\Gamma)$ in eqn. (9.30) is not known, we can specify

$$Q_{Tl} = \int_{\Gamma_3} q_l(\Gamma)\, d\Gamma \qquad l = \text{injected phase} \tag{9.31}$$

This equation is a constraint rather than a boundary condition.

A different situation exists at the boundary Γ_1 where fluids are produced, typically at the wellbore. At this boundary distribution of phases *cannot* be controlled; in fact, distribution of produced fluids is the main result of, for example, a coning simulation. The imposed condition, which again is a constraint rather than a boundary condition, is the specified total oil flow rate of the well

$$Q_{TO} = \int_{\Gamma_1} q_o(\Gamma)\, d\Gamma \tag{9.32}$$

or total liquid rate

$$Q_{TL} = \int_{\Gamma_1} (q_o + q_w)\, d\Gamma \tag{9.33}$$

or the total fluid rate

$$Q_{TT} = \int_{\Gamma_1} (q_o + q_w + q_g)\, d\Gamma \tag{9.34}$$

Also, one may wish to specify the rates at standard rather than reservoir conditions.

The problem of assigning boundary conditions $q_l(z)$ for either production or injection boundary has therefore two possible aspects:

1. At a given point at the boundary, the flow must be distributed among the phases. This problem has been discussed in detail in Chapter 5, Section 5.7.
2. The flow of each phase must be distributed along the boundary.

The correct distribution must, of course, satisfy Darcy's Law. We can write, for example, for the case of total liquid rate specified:

$$\frac{q_l(\Gamma)}{q_o(\Gamma)} = \frac{\left[\lambda_l\left(\frac{\partial p_l}{\partial n} - \gamma_l \frac{\partial h}{\partial n}\right)\right]_{\Gamma_1}}{\left[\lambda_o\left(\frac{\partial p_l}{\partial n} - \gamma_o \frac{\partial h}{\partial n}\right)\right]_{\Gamma_1}} \qquad l = w, g \tag{9.35}$$

and

$$Q_{TL} = \int_{\Gamma_1} \left[\lambda_w\left(\frac{\partial p_w}{\partial n} - \gamma_w \frac{\partial h}{\partial n}\right) + \lambda_o\left(\frac{\partial p_o}{\partial n} - \gamma_o \frac{\partial h}{\partial n}\right)\right] d\Gamma \tag{9.36}$$

Therefore, if the pressures and saturations at the boundary are *known*, these equations can be used to define q_l. The problem of specifying boundary condition is to distribute the rates without knowing the solution.

9.4.2 Compatibility Conditions and Constraints

The distribution of rates along the boundary is not always arbitrary. For example, for a production well the pressure on the boundary must be the same as the pressure in the wellbore. Pressure distribution in the wellbore, which can be calculated from the equations of multiphase flow in vertical pipes, determines the distribution of production along the perforated boundary. This compatibility condition should be taken into account especially in cross-sectional and coning problems and will be discussed in more detail in Section 9.8.

Similarly, compatibility of pressures will determine the flow between the matrix and the fracture in fractured reservoirs.

Conditions of this kind can also arise at injection boundaries, in cases when boundary conditions are used to account for the influence of the rest of the system not being simulated, and when boundary conditions for one model are determined from the simulation of a larger portion of the reservoir by another model.

A constraint of a different kind is often important in practice, when the well is produced at a given rate subject to a minimum flowing bottom hole pressure. Such a case is a combination of rate and pressure boundary conditions.

9.4.3 Finite-Difference Formulation

9.4.3.1 *Dijference Equations for the Boundary Points*

The handling of boundary conditions in the finite-difference equation is exactly the same as in single-phase flow. As discussed at length in Section 7.4 (Chapter 7), the boundaries are assumed to be no-flow and the flow is accounted for by source terms.

The equations for boundary points can again be written in the form (9.9), if the coefficients are suitably modified. For example, for the case shown in Fig. 7.9, it is sufficient to define and use again equations of the type of (9.6) and (9.8) to calculate TX_l and TZ_l.

$$TX_{li-1/2} = 0 \qquad x_{i-1/2} = x_i$$

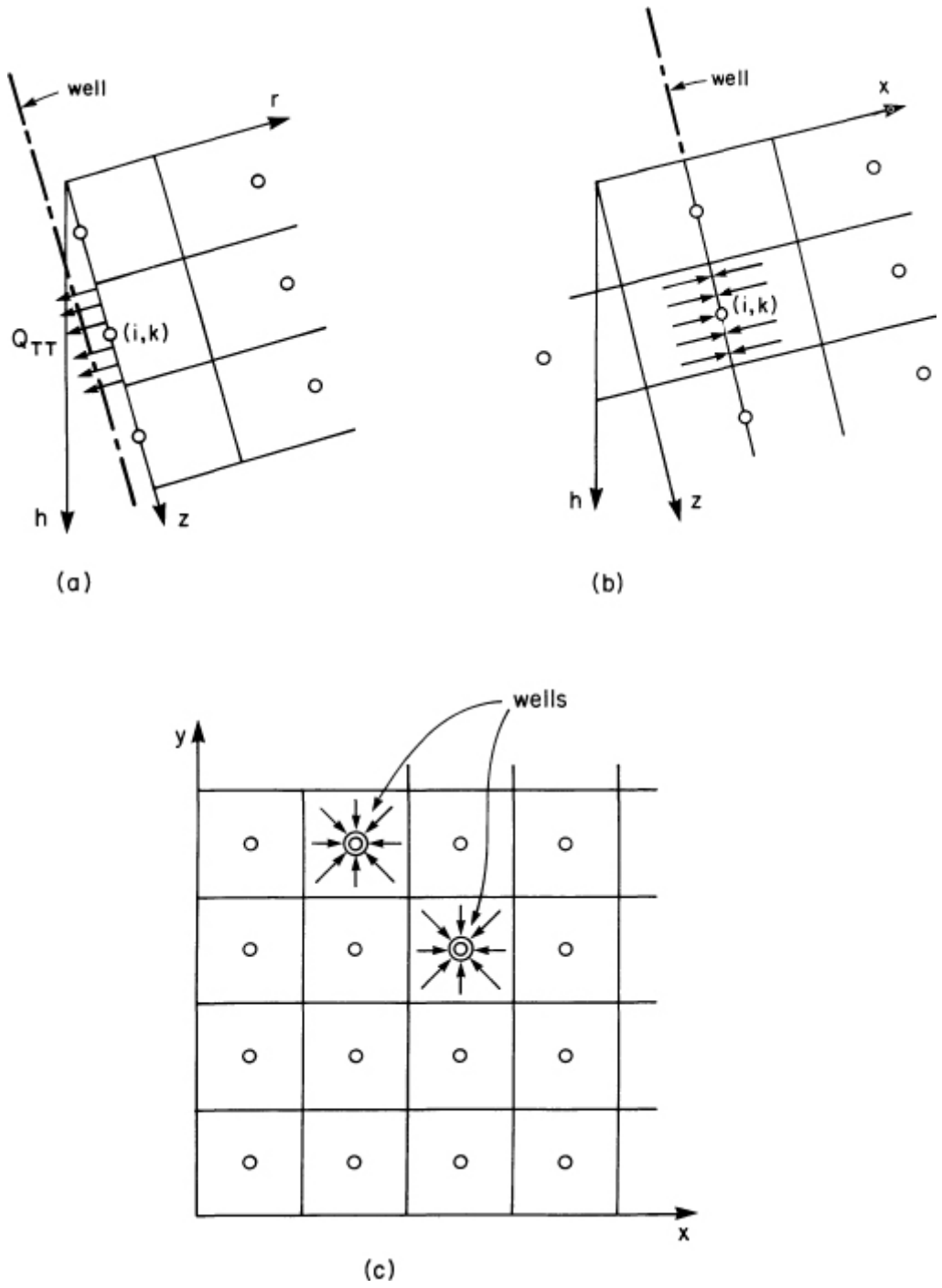


FIG. 9.6. Representation of wells in different types of grids, (a) Single-well; (b) cross-section; (c) areal.

We now turn to the question of determining the source terms q_l at a given boundary point. This is completely analogous to the 1-D case discussed in Section 5.7 (Chapter 5), once the transmissibility terms have been properly defined.

Let us first consider production across a face of a single block (i, k), as shown in Fig. 9.6(a). This can represent a well boundary for a coning model or a boundary across

which there is a movement of fluid. Then we can discretise eqn. (9.30) as in the 1-D case:

$$Q_l = -\lambda X_{li} \frac{\Delta z \, \Delta y}{\Delta x_{i+1/2}} [p_{li+1} - p_{li} - \gamma_{li+1/2}(h_{i+1} - h_i)] \tag{9.37}$$
$$= -\mathrm{T}Q_{li} \, \Delta\Phi_{li+1/2}$$

For a well inside the grid, either in a cross-sectional (Fig. 9.6b) or areal model (Fig. 9.6c), the flow rate can be expressed as

$$Q_l = -WI\bar{\lambda}_l \Delta\Phi_l = -\mathrm{T}Q_l \Delta\Phi_l \tag{9.38}$$

where WI is a known coefficient, $\lambda_l = kk_{rl}/\mu_l$ is a mobility, representative of the area around the well, and $\Delta\Phi$ is the pressure difference between the flowing bottom hole pressure and the average pressure of the block. The coefficient WI is proportional to the customary productivity index of the well, but it does not include the effect of fluid mobility. Equation (9.38) for the oil phase will become the familiar productivity index equation

$$Q_o = PI \, \Delta\Phi_o$$

if the block pressure is close to the pressure at the drainage radius of the well and if $S_w = S_{wc}$ and $S_g = 0$. The coefficient WI can be calculated from PI using $\lambda_o(S_{wc}, S_g = 0)$ and, if necessary, by accounting for the block size (which may be smaller than the drainage radius). This can be done using methods discussed in Section 7.7 (Chapter 7).

Once we have defined the transmissibility terms TQ in (9.37) or (9.38) the various options and methods for distributing the rates considered in Chapter 5 can be directly applied.

9.4.3.2 *Distribution of Production Between Blocks*

The formula for distributing the production between blocks is obtained from formulae for a single block by summation over the blocks. In a cross-sectional or coning simulation, the well will normally penetrate a column of blocks and we can write for the case of total rate Q_{TT}:

$$Q_{lk} = \frac{(B_l \mathrm{T}Q_l)_k \, \Delta\Phi_{lk}}{\sum_k \sum_l (B_l \mathrm{T}Q_l)_k \, \Delta\Phi_{lk}} Q_{TT} \tag{9.39}$$

for the 'potential allocation' method and

$$Q_{lk} = \frac{(B_l \mathrm{T} Q_l)_k}{\sum_k \sum_l (B_l \mathrm{T} Q_l)_k} Q_{\mathrm{TT}} \tag{9.40}$$

for the 'transmissibility allocation' method. It should be recognised that eqn. (9.40) assumes not only zero saturation gradient (i.e., same potential gradient for all phases) but also the same potential gradient for all layers. Therefore the mobility allocation may give erroneous results in the case of large vertical heterogeneity and especially when non-communicating layers exist (Nolen and Berry, 1972).

Neither of the above methods recognise the compatibility condition in the case of the flow in the wellbore. While the rigorous solution to this problem is given in Section 9.8, we will describe here an approximate procedure which can be used successfully in cross-sectional and 3-D models.

Consider again a vertical column of production blocks. We assume $\Delta\Phi_{lk} = \Delta\Phi_k = p_k^{\mathrm{w}} - p_{lk}$ where p_{lk} is known and p_k^{w} is the pressure at the wellbore. The pressure drop in the wellbore can be approximated by the density head, which gives the compatibility condition as

$$p_{k+1}^{\mathrm{w}} = p_k^{\mathrm{w}} + (\bar{\rho}\,\Delta z)_{k+1/2} \tag{9.41}$$

where ρ is a mean density.

Then the rates can be expressed in terms of p_K^{w}, which is the wellbore pressure at the uppermost perforated block, ($k = K$). The total rate is therefore

$$Q_{\mathrm{TT}} = \sum_k \sum_l (B_l \mathrm{T} Q_l)_k \left[p_K^{\mathrm{w}} - p_{1k} + \sum_{m=k}^{k-1} (\bar{\rho}\,\Delta z)_{m+1/2}\right] \tag{9.42}$$

The equation can be solved for p_K^{w} and then (9.41) will give p_k^{w} and $\Delta\Phi_k$ for each layer for the determination of Q_{lk}:

$$Q_{lk} = (B_l \mathrm{T} Q_l)_k (p_k^{\mathrm{w}} - p_{1k}) \tag{9.43}$$

Equation (9.42) must be solved using pressures at the known time level n, and therefore $p_k^{\mathrm{w}} = p_k^{\mathrm{w}n}$. Explicit treatment of pressure in eqn. (9.43) introduces instability and therefore the rates must be treated implicitly:

$$Q_{lk} = (B_l \mathrm{T} Q_l)_k (p_k^{\mathrm{wn}} - p_{1k}^n) - (B_l \mathrm{T} Q_l)_k (p_{1k}^{n+1} - p_{1k}^n) \tag{9.44}$$
$$= Q_{lk}^n - (Q'_{1k})_{pl} \Delta_t p_{1k}$$

This formulation is satisfactory when transmissibilities can be treated explicitly. However, the total rate of the well will not be maintained with the formulation (9.44). The deviation from the prescribed rate is proportional to the pressure changes over a time step and if the rate is required accurately, some iterative procedure must be used.

With implicit treatment of transmissibilities, the rates must also include implicit terms with respect to $\Delta_t S_l$, as discussed in Section 5.7 (Chapter 5).

9.5 INITIAL CONDITIONS

The usual initial condition of a reservoir, first treated in the literature by Muskat (1949a), is the state of static equilibrium at which velocities of all phases are zero. Then the pressures are functions of z only and

$$\frac{\mathrm{d}p_l}{\mathrm{d}z} = \gamma_l \qquad l = \mathrm{o}, \mathrm{w}, \mathrm{g} \tag{9.45}$$

Due to differences in gravity and capillary forces fluids segregate (Fig. 9.7). If it is assumed that the same capillary pressure curves and critical saturations that apply to the production phase can also be used for the equilibrium state, then the pressure and saturation distribution is uniquely determined by specifying one pressure and two reference saturations: $S_{\mathrm{w}}^r > S_{\mathrm{wc}}$ at a depth z_{ow} in the oil-water transition zone, and $S_{\mathrm{g}}^r > S_{\mathrm{gc}}$ in the oil-gas transition zone. In the transition zone both phases are mobile. Note that the condition (9.45) applies only when a phase is mobile. In regions where a phase is immobile, the forces are not necessarily in equilibrium. For example, for elevations above z_{wc} in Fig. 9.7, the water saturation is S_{wc} and therefore $P_{\mathrm{cow}} = P_{\mathrm{cow}}(S_{\mathrm{wc}})$. Hence water in this zone will follow the *oil* density gradient as shown on the figure by a dashed line. Similarly, when water saturation reaches S_{wmax} and consequently $k_{\mathrm{ro}} = 0$, oil pressure gradient follows the water gradient. However, it is usually not valid to assume some finite critical saturation for the oil phase in the water zone formed by a natural aquifer, where $S_{\mathrm{w}} = 1$. The saturation distribution shown by the dashed line on the left part of Fig. 9.7, could result from previous production of the reservoir by a bottom water drive, followed by a shut-in period.

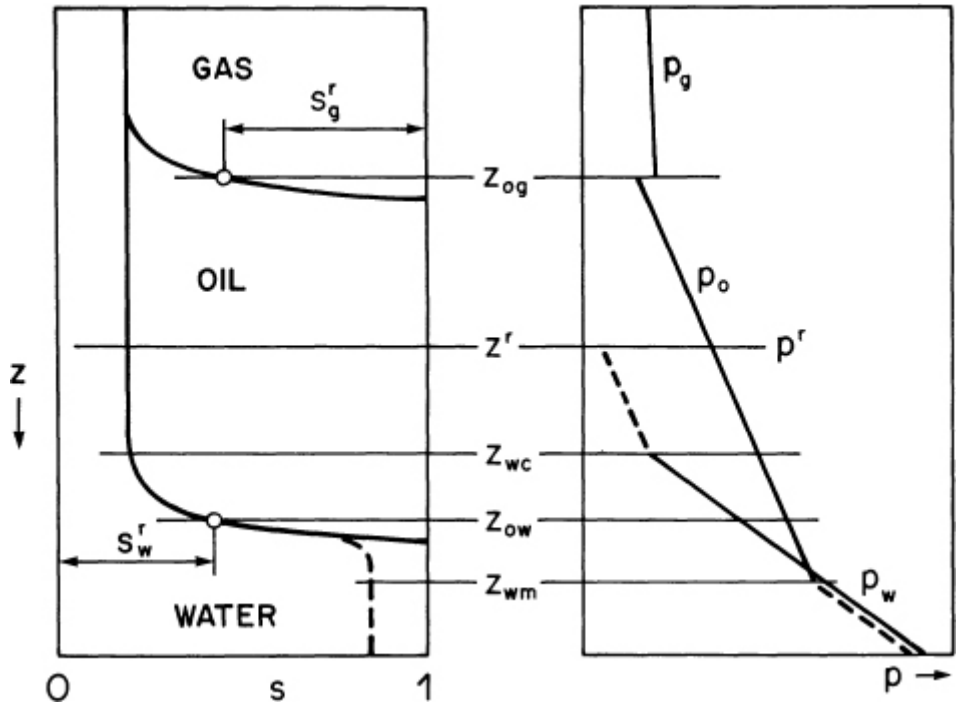


FIG. 9.7. Initial saturation and pressure distribution in the reservoir.

In the finite-difference model, it is customary to calculate $P_{ck} = P_c(z_k)$ and then define $S_{wk} = S_w(P_{ck})$. This does not account for averaging over the thickness of a block. In the extreme case of $P_c \rightarrow 0$ this could lead to an error in the initial volume in place of up to $(V_p/2)(S_{wmax} - S_{wc})$. This is illustrated in Fig. 9.8, where the actual average saturation for the block is shown by the dotted line. In most cases this error is not serious because of large uncertainties in reservoir definition and volumes in place. When required, the averaging should be done by defining pseudo-functions for each layer within the layer thickness and these functions should also be used in the simulation of flow, at least during the initial period. In this way the simulation of horizontal flow will also be consistent with the averaged saturations. The concept of pseudo-functions will be discussed in Chapter 12.

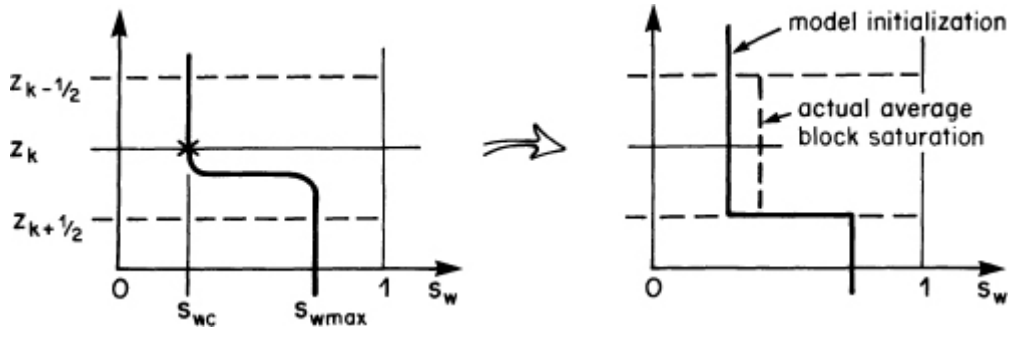


FIG. 9.8. Discretisation of initial conditions.

9.6 SIMULATION OF AQUIFERS

For the production of oil from many reservoirs, the source of energy is an adjoining aquifer. Aquifers can be always simulated by their inclusion in the computational grid. Because there is only single-phase flow in the aquifer, a coarser grid can be used, as shown in Fig. 9.9.

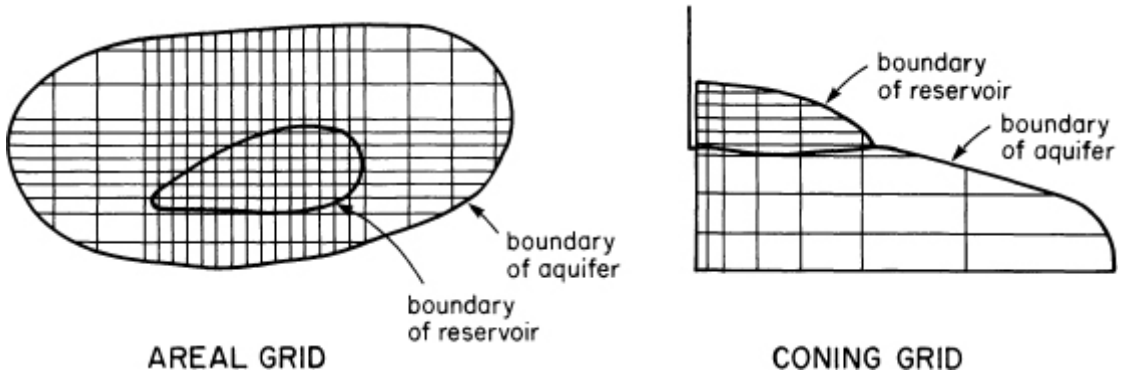


FIG. 9.9. Examples of grid covering reservoir and aquifer.

Such a treatment increases considerably the number of grid blocks required even when a coarse grid is used in the aquifer. The increased running time and storage are only rarely justified, because the aquifer is usually not as well defined as the reservoir.

Since the aquifer causes the influx of water to the reservoir, its influence can be simulated by restricting the grid to the reservoir and defining suitably the water influx in the blocks which are on the reservoir aquifer boundary. (It is also possible to extend the grid so that the boundary blocks actually lie in the aquifer; Coats (1968).)

Three methods of calculating aquifer influx are described below:

(a) '*Pot*' *Aquifer*. If the aquifer is relatively small and has closed boundaries, it will be in approximate equilibrium with the reservoir at all times. Therefore, flow will occur only when pressure at the reservoir–aquifer boundary changes, i.e.,

$$Q_A = -cV_{pA}\,\partial\bar{p}/\partial t \tag{9.46}$$

where Q_A is the rate of influx in units of reservoir volume/day, c is the total compressibility (water and rock) in the aquifer ($c = c_w + c_R$), V_{pA} is the pore volume of the aquifer, and $\bar{p}$ is the average pressure at the aquifer/reservoir boundary (assumed to be equal to the average aquifer pressure $\bar{p}_A$).

In finite-difference form, (9.46) becomes

$$Q_{\mathrm{A}ij}^{n+1} = -cV_{p\mathrm{A}ij}\frac{(p^{n+1}-p^{n})_{ij}}{\Delta t} \tag{9.47}$$

where $\bar{p}$ has been replaced by the block pressure p_{tj} and $V_{p\mathrm{A}ij}$ is the part of $V_{p\mathrm{A}}$ that 'belongs' to the reservoir block ij. For example, $V_{p\mathrm{A}ij}$ may be defined for the areal case as

$$V_{p\mathrm{A}ij} = V_{p\mathrm{A}}\frac{A_{ij}}{\sum A_{ij}} = \alpha_{ij}V_{p\mathrm{A}} \tag{9.48}$$

where A_{ij} is the area for aquifer influx for block (ij) and the summation is over all blocks with aquifer influx. In a heterogeneous reservoir, it may be better to distribute the influx according to transmissibilities.

(b) *Steady-State Infinite Aquifer*. This model (Schilthuis, 1936; Katz *et al*., 1963) assumes that the pressure at the external boundary of the aquifer does not change. Then the rate of influx is calculated as

$$Q_{\mathrm{A}} = C_{\mathrm{A}}(p_{\mathrm{A}}^{(i)} - \bar{p}) \tag{9.49}$$

where $p_{\mathrm{A}}^{(i)}$ is the pressure at the external boundary of the aquifer and it is assumed to be constant.

In finite-difference form, we define in analogy with (9.48) $C_{\mathrm{A}ij} = \mathrm{C_A}\alpha_{ij}$ where α_{ij} is a coefficient which distributes the influx among the blocks, and let $p = \frac{1}{2}(p^{n+1} + p^{n})_{ij}$ for the time interval from t^n to t^{n+1}. Then

$$Q_{\mathrm{A}ij}^{n+1} = [C_{\mathrm{A}ij}(p_{\mathrm{A}}^{(i)} - p_{ij}^{n}) - \tfrac{1}{2}C_{\mathrm{A}ij}(p^{n+1} - p^{n})_{ij}] \tag{9.50}$$

(c) *Unsteady-State Aquifer*. The two methods described above represent extreme cases. For the pot aquifer, the influx does not depend on the pressure level (the average aquifer pressure $\bar{p}_{\mathrm{A}}$ follows the pressure $\bar{p}$). For the steady-state aquifer, $\bar{p}_{\mathrm{A}}$ does not change $p_{\mathrm{A}} = p_{\mathrm{A}}^{(i)}$. In reality, the rate of influx is between these extremes and it can be obtained by solving the transient flow equation for the aquifer. This approach has been developed by Van Everdingen and Hurst (1949), and Hurst (1958) for analytical calculation, and it allows the calculation of $Q_{\mathrm{A}}(t)$ in some simple cases. Analytical solutions can be obtained for two basic cases of constant pressure drop ($p_{\mathrm{A}}^{(i)} - p$) or

constant rate Q_A (usually referred to as the constant terminal pressure and constant terminal rate cases). It is customary to express the results using 'influence functions' $QI(t)$ and $PI(t)$ as follows:

$$W(t) = \int_0^t Q_A(t) = QI(t)(p_A^{(i)} - \bar{p}) \tag{9.51}$$

for the case of constant pressure $\bar{p}$, and

$$p_A^{(i)} - \bar{p}(t) = PI(t)Q_A \tag{9.52}$$

for the case of constant rate Q_A. The functions QI and PI depend on aquifer geometry and properties. For example, an approximate solution of eqn. (9.52) for the case of a cylindrical aquifer with external radius r_e was given in Section 7.7 (Chapter 7). Influence functions for many other cases have been tabulated (Katz *et al.*, 1963; Katz and Coats, 1968). Since both $\bar{p}$ and Q_A are variable for a real aquifer, the correct solution must be obtained using the principle of superposition in conjunction with the material balance on the reservoir (Van Everdingen and Hurst, 1949). A simplified method, suitable for computer application, was proposed by Carter and Tracy (1960). Their method utilises the terminal rate influence function $PI(t)$ and leads to the expression

$$Q_{Aij}^{n+1} = \alpha_{ij}[a(t) + b(t)(p^{n+1} - p^n)_{ij}] \tag{9.53}$$

where $a(t)$ and $b(t)$ are functions of $PI(t)$ and cumulative influx $W(t)$ (see Exercise 9.1 for the definition of a and b).

A simple approach, applicable to any representation of a finite aquifer, was proposed by Fetkovich (1971). In his method, the rate influx during the time step n is calculated from the rate equation

$$\Delta t^n Q_{Aij}^{n+1} = \Delta t^n \alpha_{ij} J_A^n (\bar{p}_A^n - p_{ij}^{n+1}) \tag{9.54}$$

where p_A^n is the average aquifer pressure at the time t_A^n and J^n is the aquifer productivity index. After the time step is completed, the cumulative influx W^{n+1} is calculated, and p_A^{n+1} is updated using the material balance equation on the aquifer. Fetkovich has shown that one can use the pseudosteady-state or steady-state equations derived in Chapter 7, Section 7.7 to define J_A during a time step, even when the aquifer behaviour is transient. Thus, his method is completely analogous to the treatment of

individual wells in Chapter 7, and by a suitable choice of the rate function (9.54) can encompass all three types of aquifer behaviour discussed above.

9.7 SIMULATION OF AREAL AND CROSS-SECTIONAL PROBLEMS

In this section, we will discuss some special techniques, most of which are common to both areal and cross-sectional problems.

9.7.1 Use of Curvilinear Grid

Curvilinear grid has been treated in detail in Section 7.10.2 (Chapter 7). The treatment applies equally to multiphase flow. Sonier and Chaumet (1974) reported use of orthogonal curvilinear grid for areal simulation. They suggest that steady-state single-phase (potential) flow distribution with sources and sinks in well locations could be used to develop equipotential lines and streamlines, and then these lines could be used to define an orthogonal grid. This results in a grid with increased definition around wells, which in turn may require implicit treatment of transmissibilities.

A grid of this type is advantageous in studies of areal displacement in pattern flooding, when the grid can be easily constructed. Such grid will give accurate predictions of breakthrough time and well pressures with fewer grid blocks than a Cartesian grid with conventional irregular grid refined around wells. According to Robertson and Woo (1976), an additional benefit of curvilinear grid is that it reduces the grid orientation effect (see Section 9.7.3).

For simulation of an entire field with a large number of arbitrarily spaced wells, the construction of the grid is more difficult. Because the number of blocks that can be used is usually limited by the computer limitations, curvilinear grid is rarely used and the details of the flow in the vicinity of wells are obtained by the methods discussed in the following section.

Hirasaki and O'Dell (1970) discuss the more general case of a three-dimensional reservoir with variable dip and thickness. They show that large errors can occur, if the transmissibilities are calculated from block sizes projected onto a horizontal plane and thicknesses measured in the vertical direction. Correct solution can be always obtained using curvilinear grid and the general difference operator including the cross-derivative terms.

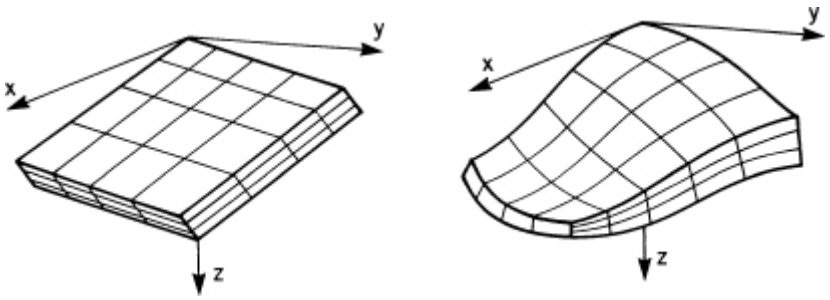


FIG. 9.10. Grid design for dipping reservoirs.

In practice, good results are obtained with rotated Cartesian coordinates orientated along the bedding plane of the reservoir if the angle of dip is small or approximately constant as shown in Fig. 9.10 (left). For cases with variable dip angle and thickness as shown in Fig. 9.10 (right), an approximately orthogonal grid may be constructed by projecting an areal Cartesian grid constructed in the average bedding plane and then constructing a vertical grid perpendicular to the actual bedding plane. The vertical grid constructed in this manner can be used for cross-sectional models or it provides the direction in which thickness of the blocks is determined in areal models. In all cases one can use eqns. (7.111) to calculate the transmissibilities provided all areal dimensions are measured along the surfaces and Δz perpendicular to them, rather than in horizontal and vertical directions.

Hirasaki and O'Dell (1970) also show the errors associated with using skewed co-ordinates in the areal plane (Fig. 9.11). Such a grid provides a method of conveniently accommodating well alignment, faults, and fractures. They show that neglecting the cross-derivative terms causes large errors for $\theta \geq 15°$. Consequently, this grid should be used with conventional difference equations only for small angles ($\theta < 5°$).

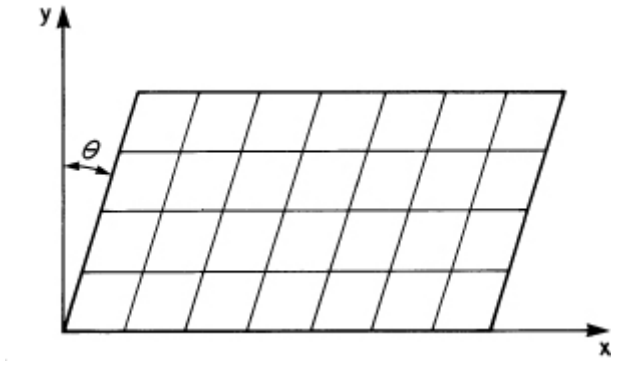


FIG. 9.11. Skewed co-ordinate system.

9.7.2 Treatment of Individual Wells

Because the block size is large compared to the well radius, an areal or cross-sectional model cannot predict saturations and pressures at the well without special treatment.

9.7.2.1 *Prediction of Well Pressure*

In single-phase flow problems the flowing well pressure p_{wf} may be predicted from the block pressure p_{ij} by invoking the assumption of radial flow within the area of the block. Appropriate equations to do this were developed in Chapter 7, Section 7.7.

The same analysis applies equally for multiphase flow if Q is the total rate in reservoir units and λ is a representative total mobility

$$\bar{\lambda}_{\mathrm{T}} = k\left[\frac{k_{\mathrm{ro}}}{\mu_{\mathrm{o}}} + \frac{k_{\mathrm{rw}}}{\mu_{\mathrm{w}}} + \frac{k_{\mathrm{rg}}}{\mu_{\mathrm{g}}}\right]$$

The rates for individual phases are then distributed as discussed in Section 9.4. Equations (7.60), (7.63), (7.64) or (7.72) are useful when it is necessary to simulate a well producing against a fixed bottom hole pressure p_{w} (pressure boundary condition). The unknown rate Q_l^{n+1} is expressed according to these equations as

$$Q_l^{n+1} = C_l(p_{\mathrm{wf}} - p_{lij}^{n+1}) \tag{9.55}$$

which can be easily implemented by including $C_l\, p_{wf}$ in the vector $\boldsymbol{Q}$ and modifying the corresponding diagonal element of matrix **T.**

The main problem in applying formula (9.55) is in obtaining the correct value for the mobility λ_{T}, which must account for the saturation changes in the vicinity of the well and in general can not be determined from the average saturations of the block. In the following section, we will introduce methods for predicting saturations at the well (or distribution of rates) from the block saturations. Some of them can also be used to obtain the correct well pressure, and can thus be viewed as methods for the determination of λ_{T}.

9.7.2.2 *Handling of Production in Multiphase Flow*

The relative production of each phase (WOR, GOR) depends on the saturations at the well. Because the saturation S_{lij} is a volume average over the grid block, the calculated WOR and GOR are usually too low when based on mobilities calculated from S_{lij}. Such calculations lead to optimistic predictions of oil recovery rate. This problem is common to 2-D and 3-D simulations.

There are several methods which can be used to obtain realistic predictions. The common idea is to obtain relationships for coning behaviour of the well as a function of average block saturation $S_{av} = S_{ij}$ (and possibly some other parameters). The methods then differ by the techniques used to predict the coning behaviour. The result can be used in the reservoir model in two ways:

(a) By defining pseudo-relative permeability curves that are different from the curves for flow inside the reservoir. These curves are then used for the distribution of production. The shift of pseudo-relative permeability curves from the rock curves depends upon the displacement process. An example is shown in Fig. 9.12. For bottom-water drive the pseudo k_{rw} curve will be above the reservoir curve (curve 1, Fig. 9.12). However, for an edge-water drive, the pseudo-curve will be below the rock curve (curve 2), because the water will invade the blocks on the periphery (Stright, 1973). This concept of pseudo-functions is discussed in Chapter 12.

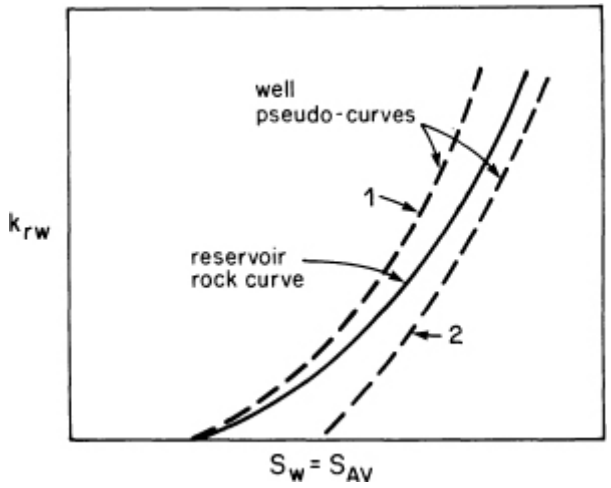


FIG. 9.12. Pseudo-relative permeabilities for a well.

(b) By developing correlations of WOR (GOR) as a function of S_{av}. Although each well should have a unique correlation, it is often possible to develop general correlation with parameters such as formation thickness etc. (Stright, 1973; Blades and Stright, 1975).

Obviously, the two methods of expressing the coning behaviour are equivalent. We will now discuss briefly three types of methods for obtaining these correlations, which also represent three levels of sophistication.

(a) *Analytical Coning Correlations*

The classical correlation of Sobocinski and Cornelius (1965), which only predicts breakthrough time, has been extended by Bournazel and Jeanson (1971), Telkov (1971) and Cottin (1971) to predict WOR. However, these correlations are generally *not reliable* unless there is sufficient history to calibrate them against field performance. A method of this type was also proposed by Chappelear and Hirasaki (1976). Their method is in the form of an empirical equation which is quadratic in water cut and agrees well with results of some coning model studies.

(b) *Correlations Based on Coning Simulation*

Because even the best analytical correlations make a number of simplifying assumptions, the correlations for practical problems are usually developed from 2-D coning simulations. The total pore volume of the coning model should be equal to the pore volume of the corresponding areal block. The influx into the coning model can be approximately determined from an areal simulation with reservoir k_r curves or estimated pseudo-curves for the well. These rates will change when the proper pseudo-curves are used, and therefore the correlations should be iteratively updated until both models predict the same WOR and GOR behaviour. In most cases, the sensitivity of correlations to the influx rate is small and iteration is not necessary.

(c) *Coupling of a Well* (*Coning*) *Model with the Reservoir Model*

This is the most rigorous, but also the most expensive approach. The work of Akbar *et al*. (1974) describes use of a 1-D radial simulator coupled with a 2-D areal model. While this approach gives a reasonable pressure distribution, it is not valid for predicting coning, because the vertical dimension is neglected. Mrosovsky and Ridings (1974) developed a technique for coupling 2-D radial models with a 3-D (or 2-D) reservoir model. This procedure predicts both pressures and saturations at the well. The technique involves iteration between the two models during a time step in order to obtain correct influx in the coning model. Such approaches can be prohibitively expensive.

9.7.3 Grid Orientation Phenomenon

Consider simulation of displacement in a confined five-spot pattern. Because of symmetry, the grid can be oriented in two ways, shown in Fig. 9.13, which are called diagonal and parallel grid. Because the smallest element of symmetry is one-half of the diagonal grid, one would naturally use it for best computer efficiency. However, it has been found that, under certain conditions, the two grid orientations will give quite different answers, and this has been termed the *grid orientation phenomenon*. If we simulate injection in a single well, the saturation contours will look as schematically shown in Fig. 9.14(a) and (b). Both these solutions are *incorrect*, because the front

should be essentially radial until it moves sufficiently close to the producing wells. For early times the two saturation profiles obtained numerically are almost identical when rotated with the coordinate system. It is obvious that the predicted breakthrough will be *early* for the parallel grid and *late* for the diagonal grid. The important fact, first reported by Coats *et al*. (1974) and confirmed by several others, is that *the finite-difference solution converges to two different results for the two orientations of the grid as the grid is refined*. This indicates that the phenomenon *cannot* be a result of truncation errors alone (a similar example was discussed in Chapter 5, Section 5.5.1).

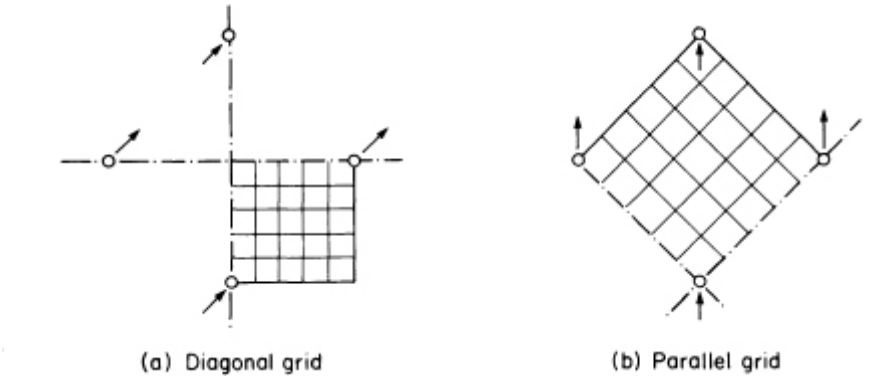


FIG. 9.13. Definition of the diagonal and parallel grid.

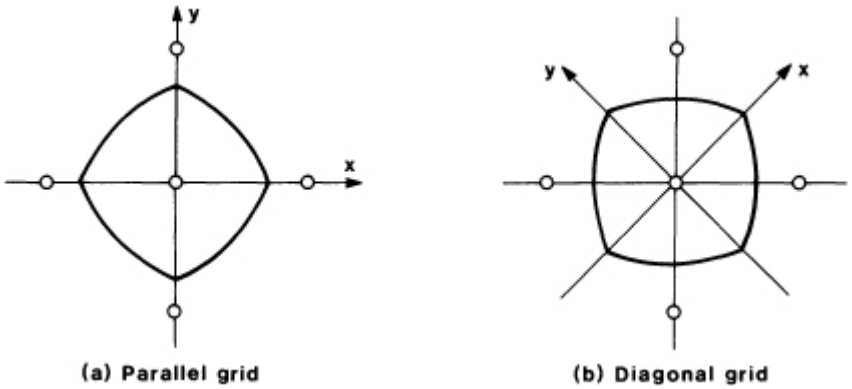


FIG. 9.14. Calculated saturation contours around injection well. Grid lines are parallel to co-ordinate directions.

The characteristic 'diamond' shape of the saturation fronts was first pointed out by Todd *et al*. (1972). These authors found relatively mild orientation effect in immiscible displacement with mobility ratio between 1 and 10 (unfavourable) and demonstrated that the two-point upstream weighting of relative permeability reduces the effect, but the solution still converges to two different results. Work of both Todd *et al*. (1972) and Spivak *et al*.(1977) indicated that the orientation effect is relatively mild for waterflood

applications. However, Coats *et al*. (1974) have shown that the effect can be very large for simulating steam displacement with a very high unfavourable mobility ratio. The problem is even more severe for simulating miscible displacement, where the results are unacceptable even for moderate mobility ratios (Settari *et al*., 1977).

9.7.3.1 *Factors Influencing Orientation Effect*

In general, the orientation effect increases with increasing height of the saturation front and it is largest for a piston-like (or miscible) displacement (Fig. 9.15). The shape of the front depends on k_r curves and mobility ratio.

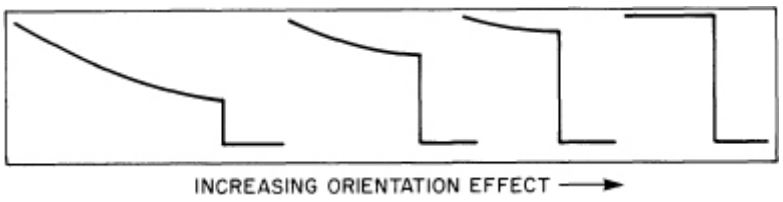


FIG. 9.15. Dependence of orientation effect on the fractional flow characteristics.

On the other hand, the orientation effect is decreased by dispersion-like terms (capillary pressure and gravity in immiscible flow and physical dispersion in miscible displacement), which tend to smooth the front (Fig. 9.16).

In the solution of miscible problems, the orientation error increases with increasing *unfavourable* mobility ratio and decreases with increasing dispersion. The effect is almost absent for unit or favourable mobility ratio.

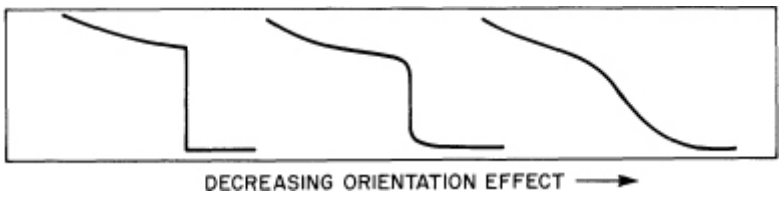


FIG. 9.16. Dependence of orientation effect on the dispersion terms.

The extent of the orientation effect in cross-sectional problems is difficult to evaluate because it would have to be obtained by comparison with analytic solutions. However, we believe that the orientation effect is actually beneficial to standard finite-difference approximation. One of the main concerns in cross-sectional problems is the correct simulation of gravity underrun or override. Because of numerical dispersion (truncation errors), it is sometimes necessary to use a large number of blocks in the vertical

direction to obtain required accuracy (Fig. 9.17). The grid orientation effect results in preferential flow *along* the grid lines, which in this case counteracts the vertical mixing caused by numerical dispersion.

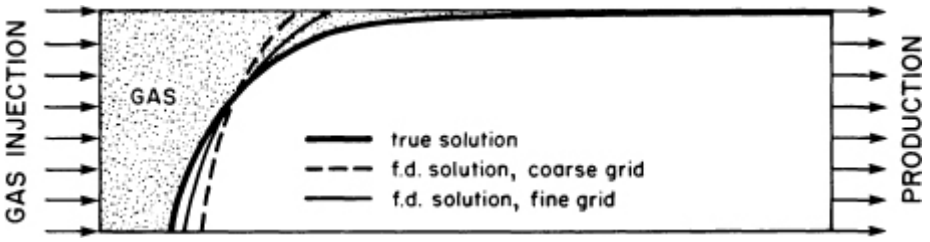


FIG. 9.17. Effect of numerical dispersion in a cross-sectional gas injection problem.

9.7.3.2 *Techniques for Eliminating the Orientation Effect*

The origin of the orientation effect can be shown by the following heuristic argument:

Consider water injection in an oil reservoir and assume that we use a method explicit in transmissibilities. Then the water saturation at any grid block can increase above the irreducible value only when one of the transmissibilities for this block becomes non-zero. At the first time step, only the saturation at the injection block can change. At the second time step, the water saturation becomes mobile at the two adjacent points, etc., as shown in Fig. 9.18.

The front of the mobile water saturation propagates as a straight line *diagonally* in relation to the co-ordinate system. Such a shape is actually obtained when the orientation effect is very strong (see Coats *et al*. (1974)). The reason for this is that water mobility at the point (i,j) can result in mobility at $(i + 1, j)$ and $(i,j + 1)$ but *not* at $(i + 1,j + 1)$ at the next time step. This is the consequence of using the five-point finite-difference formula. It appears that the problem cannot be satisfactorily solved unless the diagonal (corner) points are included in the formulation. This is supported by results of recent research discussed below.

Holloway *et al*. (1975) proposed a method in which correction to flow terms is calculated, which accounts for diagonal flow. This coupled with two-point upstream weighting resulted in reducing the orientation effect. Their method does not appear to be the complete solution. Most of the improvement resulted from the two-point upstream weighting of Todd *et al*. (1972). However, this is only a second-order improvement, and although it is noticeable when the orientation effect is mild, it fails for very severe conditions, as shown for miscible displacement by Settari *et al*. (1977). This is also confirmed by Holloway *et al*. (1975), who were not able to eliminate the grid orientation effect for a gas injection problem. Solutions free of orientation effect can be

obtained by either variational or nine-point finite-difference approximations. Variational approximations have been shown to be effective for miscible displacement by Settari *et al*. (1977) and for immiscible displacement by Spivak *et al*. (1977).

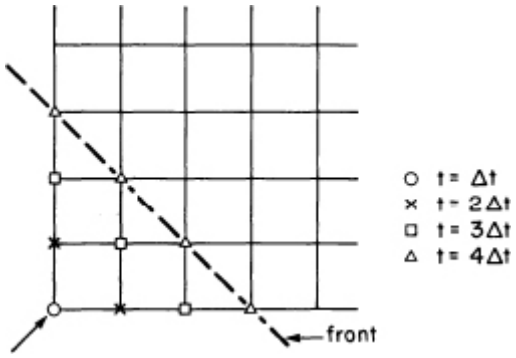


FIG. 9.18. Propagation of mobility with explicit transmissibilities.

A finite-difference method using nine-point approximation was developed by Yanosik and McCracken (1976). It should be noted that this formulation is similar to the variational approximation using Chapeau basis functions which was found to be also free of orientation effect (Settari *et al*., 1977).

When it is necessary to simulate unstable displacements with a conventional five-point approximation, the use of parallel grid gives more realistic results for both steam (Coats *et al*., 1974) and miscible (Settari *et al*., 1977) displacement. An alternative was suggested by Robertson and Woo (1976) who claim that the use of curvilinear grid (see Chapter 7, Section 7.10) eliminates orientation effects for immiscible displacement. However, they did not substantiate their claim by showing convergence of the numerical solution to the true solution.

9.8 SIMULATION OF SINGLE-WELL PROBLEMS

The main problem in solving single-well problems is the stability of the finite-difference equations. Because of the radial grid design of the model, the pore volumes of the blocks usually vary by several orders of magnitude between the smallest blocks at the well and the largest blocks at the external radius. Since the stability of the equations with explicit treatment of transmissibilities (Section 9.3.2) will be controlled by the smallest block size, this method will usually require use of impractically small time steps. This was experimentally first shown by Welge and Weber (1964). As demonstrated by Blair and Weinaug (1969), implicit treatment of transmissibilities (discussed in Chapter 5, Section 5.5.2) is necessary for this type of a problem. In fact, all the methods for implicit treatment of transmissibilities discussed in Chapter 5 have

been developed for solution of coning problems. In addition to these methods, the sequential method discussed in Section 5.6 (Chapter 5) can also be used.

9.8.1 Treatment of the Production Terms (the Well Model)

Treatment of the production terms is particularly important. Spivak and Coats (1970) have shown that implicit treatment of production terms alone results in several-fold increase in stability when the transmissibilities are treated explicitly.

Transmissibilities in the production terms may be treated implicitly as described in Section 5.7.2 (Chapter 5). In addition when the production is from several layers, the treatment should also satisfy the correct pressure drop in the wellbore (compatibility condition, cf. Section 9.4.2). The equations for the production terms are also called a 'well model' (Chappelear and Rogers, 1974).

The simplest well model is obtained when the rates are distributed according to mobilities alone (eqn. 9.40). Nolen and Berry (1972) investigated the errors caused by this simplification compared to the correct production allocation according to eqn. (9.39). For the coning problem of Blair and Weinaug (1969), they found the errors to be small unless the well was completed through a zone of very low permeability. They also reported that explicit evaluation of $\Delta\Phi$ in (9.39) caused stability problems, which were eliminated by including a simplified well model consisting of an additional column of blocks representing the wellbore with very high vertical transmissibility. Chappelear and Rogers (1974) presented a well model, based on eqns. (9.41) and (9.43), with the rate equation (9.43) treated implicitly as

$$Q_{lk}^{n+1} = (B_l \mathrm{T} Q_l)_k^n (p_k^{\mathrm{w}} - p_{ek}^n) + (B_l \mathrm{T} Q_l)_s' (p_k^{\mathrm{w}} - p_{ek}^{n+1}) \Delta_t S + (B_l \mathrm{T} Q_l)_k^n \Delta_t p_{ek}$$

Both p_k^{w} and the nonlinear term in this equation are iteratively updated to satisfy the prescribed rate.

Sonier *et al*. (1973) proposed a well model which accounts for the compatibility condition again through eqn. (9.41), but it also simulates the outlet effect (cf. Chapter 5, Section 5.7). However, their treatment of the outlet condition is not an integral part of the finite-difference scheme and requires iteration.

A well model consistent with the discretisation of the equations, was presented by Settari and Aziz (1974a). This model handles rigorously both wellbore flow and outlet effect and it is described below. The use of the point-distributed grid with grid points at the well radius provides a natural means for embedding the well model in the finite-difference equations. In what follows, we will suppress the i-subscript of the grid points (i, k) at the wellbore and consider for simplicity only two-phase flow.

9.8.1.1 *Well Model for Centre Well* (*Point-distributed Grid*)

Consider first the distribution of rates from a block in layer k. Before breakthrough, i.e., if $S_{wk} < S_{wo}$, the water production is zero, $Q_{wk} = 0$. After breakthrough, the condition is $S_{wk} = S_{wo}$, or, if the equations are solved in terms of pressures,

$$p_{ok} - p_{wk} = P_{ck} = P_c(S_{wo}) \tag{9.56}$$

The equation for the water phase for the point k after breakthrough is replaced by eqn. (9.56). After the solution has been obtained, the water production can be found by substituting the solution in the original finite-difference equation for this point. According to the definition of the residual by eqn. (5.18), tis can be written as

$$Q_{wk}^{n+1} = R_{wk}^{n+1} \tag{9.57}$$

Since

$$Q_{ok}^{n+1} = R_{ok}^{n+1} \tag{9.58}$$

the solution will satisfy, in the case of total oil production specified,

$$Q_{ok}^{n+1} = \left(\frac{R_{ok}}{\sum_k R_{ok}} \right)^{n+1} Q_{To} \tag{9.59}$$

$$Q_{wk}^{n+1} = \begin{cases} 0 & \text{if} \quad S_{wk} < S_{wo} \\ Q_{ok}^{n+1} \left(\dfrac{R_{wk}}{R_{ok}} \right)^{n+1} & \text{if} \quad S_{wk} = S_{wo} \end{cases} \tag{9.60}$$

These two equations give the rate distribution instead of eqn. (9.39).

Let us now consider production from several layers. The pressure in the wellbore p^w must obey the equations of multiphase flow in pipes. Flow in vertical pipes is discussed, among others, by Govier and Aziz (1972). Neglecting kinetic energy effects, the total pressure drop may be expressed as

$$\mathrm{d}p^w = H_g \, \mathrm{d}z + \mathrm{d}p_f \tag{9.61}$$

where $H_g \mathrm{d}z$ is the hydrostatic head term and $\mathrm{d}p_f$ is the frictional pressure drop. The frictional term can be expressed as

$$\mathrm{d}p_{\mathrm{f}} = Q_{\mathrm{o}}^{\mathrm{w}} F_{\mathrm{f}}$$

where Q_o^{w} is the oil rate flowing in the wellbore and F_{f} is a friction function which can be calculated by one of the known methods. Equation (9.61) can now be discretised as

$$\Delta p_{k+1/2}^{\mathrm{w}} = p_{k+1}^{\mathrm{w}} - p_k^{\mathrm{w}} = H_{\mathrm{g}\,k+1/2}(z_{k+1} - z_k) + F_{\mathrm{f}\,k+1/2} Q_{\mathrm{o}\,k+1/2}^{\mathrm{w}} \tag{9.62}$$

This relation replaces the approximation (9.41). In the point-distributed grid, the pressure at the grid points at the well is assumed to be the same as the pressure in the wellbore, i.e. $p_k = p_k^{\mathrm{w}}$. Using this, we can solve (9.62) for Q_o^{w} :

$$Q_{\mathrm{o}\,k+1/2}^{\mathrm{w}} = \frac{1}{F_{\mathrm{f}\,k+1/2}} [p_{k+1} - p_k - H_{\mathrm{g}\,k+1/2}(z_{k+1} - z_k)] \tag{9.63}$$

The material balance equation in the wellbore can now be written as (Fig. 9.19)

$$Q_{\mathrm{o}\,k+1/2}^{\mathrm{w}} - Q_{\mathrm{o}\,k-1/2}^{\mathrm{w}} + Q_{\mathrm{o}k}^{n+1} = 0 \tag{9.64}$$

where $Q_{o_k}^{n+1}$ is the production rate from layer k.

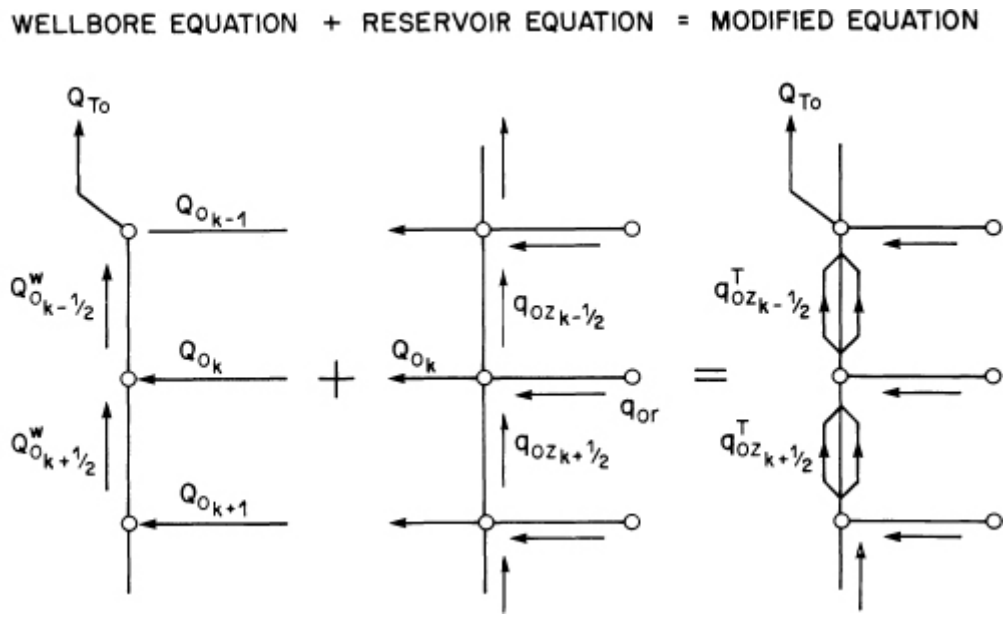


FIG. 9.19. Combination of wellbore equations with the reservoir equation.

The finite difference equation (9.58) for the production block k can be written as

$$q_{or} + q_{\mathrm{oz}\,k+1/2} - q_{\mathrm{oz}\,k-1/2} - q_{\mathrm{oacc}} = Q_{\mathrm{o}k}^{n+1} \tag{9.65}$$

where the interblock flow rates q_{or} and q_{oz} are defined in Fig. 9.19 and q_{oacc} is the accumulation term. Using (9.64) we can eliminate $Q_{o_k}^{n+1}$ from (9.65). As shown on Fig. 9.19, the resulting equation can be written as

$$q_{or} + q^{T}_{oz\,k+1/2} + q^{T}_{oz\,k-1/2} - q_{oacc} = 0 \tag{9.66}$$

where

$$q^{T}_{oz\,k+1/2} = q_{oz\,k+1/2} + Q^{w}_{o\,k+1/2}$$

Because Q_o^w as expressed by (9.63) are of the same form as the vertical flow terms, eqn. (9.66) which includes the well model can be obtained from (9.65) simply by omitting the production term and modifying the transmissibility and gravity in the vertical flow terms as follows:

(a) Replace TZ_o by $TZ_o + \dfrac{1}{F_f} = TZ_o + TW$

(b) Replace γ_o by $\gamma_o + \dfrac{TW}{TZ_o + TW} H_g$

Only the top production block will have a production term, which will be the *total oil production* Q_{To}, as shown in Fig. 9.19. The layer productions can be calculated from (9.57) and (9.58) after the solution has been obtained. The terms TW and H_g depend on the unknown rates and theoretically require iteration. The dependence is, however, weak and the iteration is not required in practice.

The model just described has the following properties:

1. The only production term occurring in the equations is the total production Q_{To}.
2. Both outlet effect and the wellbore pressure drop are satisfied rigorously.
3. The well model does not alter the structure of equations.
4. The well terms are automatically treated implicitly in the same way as interblock transmissibilities, and the total rate is always satisfied.

The model is extended for three-phase flow by specifying gas production as

$$Q_{gk}^{n+1} = \begin{cases} 0 & \text{if} \quad S_g < S_{go} \\ Q_{ok}^{n+1}\left(\dfrac{R_{gk}}{R_{ok}}\right)^{n+1} & \text{if} \quad S_g = S_{go} \end{cases}$$

where S_{go} is the gas saturation which reduces the gas-oil capillary pressure to minimum.

When bottomhole pressure is specified instead of rate, the oil equation for the top producing block K will be replaced by the equation

$$p_{oK} = p_{wf}$$

In the case of total liquid production, this formulation will require iteration, because Q_{To} is not fixed.

9.8.1.2 *Well Model for Block-centred Grid or Off-centre Well*

Consider now the case of either block-centred grid or a well completed in an interior column of blocks. In both cases the well pressure will be different from the block pressure as indicated by eqn. (9.55). The conventional method of production allocation which neglects the outlet effect should be used because the block size is too large. The compatibility condition can be easily implemented when the well produces against a specified bottomhole pressure. If the pressure p_k^w at the top production block is known, p_k^w for all layers can be computed from (9.62) using the rates from the previous time step. The rates are then calculated independently for each layer. If we use notation of eqn. (9.55), the implicitly treated rate will be

$$\begin{aligned} Q_{lk}^{n+1} &\simeq C_{lk}^n(p_k^w - p_{lk}^n) - C_{lk}^n \Delta_t p_{lk} + (C_{lk})_S'(p_k^w - p_{lk}^n)\,\Delta_t S_{wk} \\ &= Q_{lk}^n + (Q_{lk})_p' \Delta_t p_{lk} + (Q_{lk})_S' \Delta_t S_{wk} \end{aligned} \tag{9.67}$$

where C_{lk} is either a transmissibility between the block centre and the wellbore for a coning problem with block-centred grid, or a productivity coefficient in the case of a well located inside a block.

When a constant rate is specified, the same method may be used if the unknown pressure p_k^w is iteratively updated to give the prescribed rate. With a suitable algorithm, this procedure is stable and usually requires only two iterations. An alternative method is to treat p_k^w as an unknown in (9.67) and add the rate equation to the system. This approach has been used for singlephase gas well simulation by Coats *et al*. (1971) and it is also discussed by Trimble and McDonald (1976). The additional equation is obtained by summing (9.67):

$$p_K^{\mathrm{w}} \sum C_{\mathrm{ok}} + \sum (Q_{\mathrm{ok}})'_p \, \Delta_t p_{\mathrm{ok}} + \sum (Q_{\mathrm{ok}})'_S \, \Delta_t S_{\mathrm{wk}} = Q_{\mathrm{To}} - \sum_{k=K+1}^{k_{\max}} \left[C_{\mathrm{ok}} \left(\sum_{m=K}^{k-1} \Delta p_{k+1/2}^{\mathrm{w}} - p_{\mathrm{ok}}^{n} \right) \right] \tag{9.68}$$

where $\Delta p^{\mathrm{w}}_{k+1/2}$ on the right side can be calculated from (9.62). Although we could solve the augmented matrix problem, it is more convenient to use the above equation to eliminate p^{w}_k as an unknown in each of the equations containing Q^{n+1}_{lk}. Because (9.68) contains $\Delta_t p$ and $\Delta_t S_{\mathrm{w}}$ for *all* producing blocks, this operation will result in a fill of the matrix such that the equation for any production block will have a non-zero coefficient for all unknowns for all blocks $k = K, \ldots, k_{\max}$. If $k_{\max} - K > 1$, the altered structure of the matrix may increase the band-width for certain orderings of unknowns and it requires modifications of algorithms for iterative methods. For example, for any line method with lines in the z-direction, the line containing the well cannot be solved by the Thomas algorithm, because the band-width of the sub-matrix is $2(k_{\max} - K) + 1$.

9.8.2 Comparison of Stability and Efficiency of Various Treatments of Transmissibilities

Several authors studied the methods for implicit treatment of coefficients for the two-phase problem used by Blair and Weinaug (1969). Complete data are found in Settari and Aziz (1974a). Letkeman and Ridings (1970), Nolen and Berry (1972) and Settari and Aziz (1974a) among others obtained results for this problem using different versions of the SS method. Letkeman and Ridings used p, S formulation with linearised implicit transmissibilities, Nolen and Berry used the same formulation with linearised and semi-implicit treatment and Settari and Aziz used p_{o}, p_{w} formulation with Newton's iteration on transmissibility matrix and the well model described in Section 9.8.1.1. All authors found the models stable for steps up to 100 days. Time step sensitivity of the results is shown in Figs. 9.20 and 9.21. The results in Fig. 9.21 were obtained with one Newton's iteration and are, therefore, equivalent to the linearised treatment. The low time step sensitivity of the model of Settari and Aziz is attributed to the well treatment described earlier in Section 9.8.1.1. The large differences in the results obtained by the three works with essentially the same linearised method are noteworthy. The model of Letkeman and Ridings has a very large time step sensitivity. The linearised model of Nolen and Berry gives saturation oscillations for $\Delta t = 100$ days while the model of Settari and Aziz gives smooth solution. This shows that the performance of a simulator

can be strongly influenced by ‘small’ differences in implementing the method (see Section 9.8.3) and the well model.

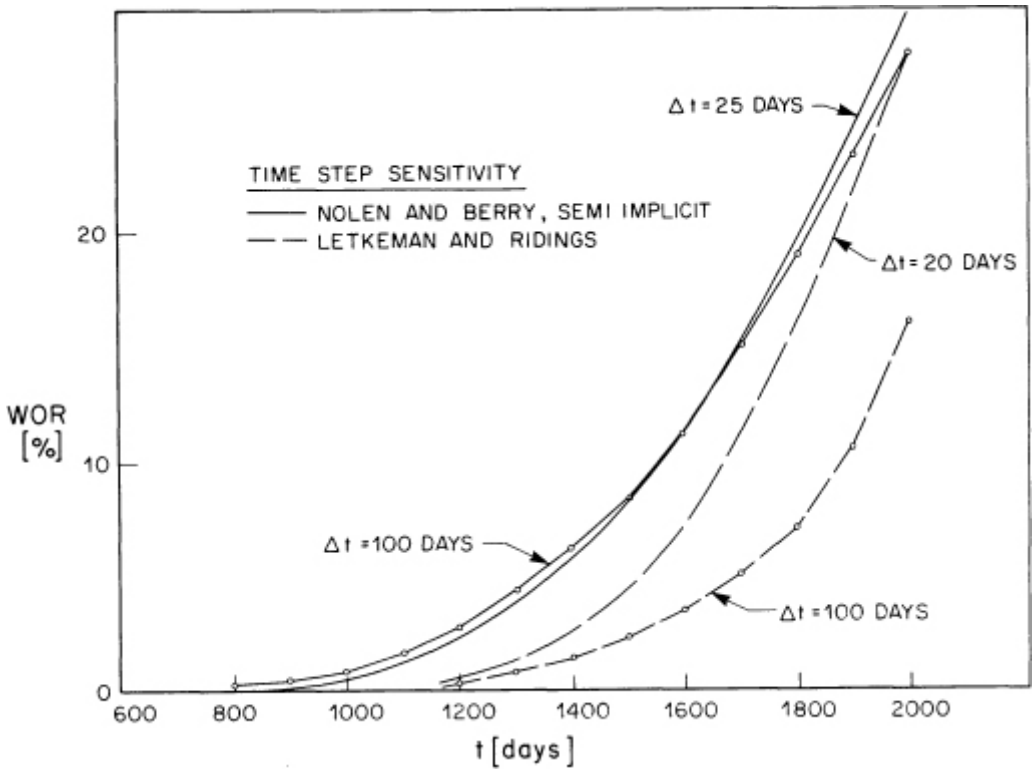


FIG. 9.20. Time step sensitivity of the models of Letkeman and Ridings and Nolen and Berry for the problem of Blair and Weinaug.

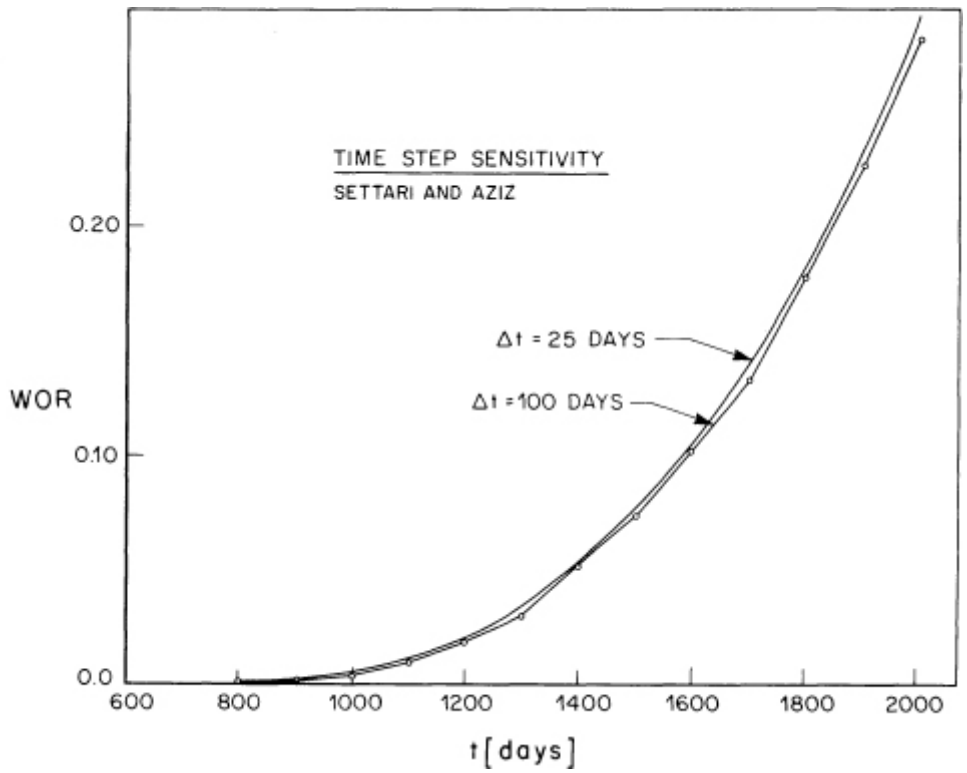


FIG. 9.21. Time step sensitivity of the model of Settari and Aziz for the problem of Blair and Weinaug.

Feasibility of Newton's iteration. Settari (1973) investigated the feasibility of Newton's method as compared to the linearised method. Although Newton's method allows use of practically unlimited time steps, the number of iterations required for convergence of one time step increases as the time step is increased and it approximately equals the number of smaller time steps one would take with the linearised method to reach the same time. (For example, Newton's method with 150-day time step would require three iterations, while the linearised method would need three 50-day time steps.) Because the time truncation errors of the fully implicit solution can be very large (Chapter 5, Section 5.5.2), a better solution (for the same amount of computer work) is usually obtained with the linearised method and smaller time steps than with Newton's method with large time step. The truncation errors can be controlled by selecting the time step size such that the maximum saturation change over the time step will not exceed a given value (usually 5–10%).

Comparison of SS and SEQ models. There are no comparisons available in the literature between the two models. Here results obtained with some commercial (INTERCOMP) models are presented. Figure 9.22 shows comparison of three models for the problem of Blair and Weinaug. The SS model is based on a *p, S* formulation with block-centred grid and conventional allocation of production. The SEQ Model I is a

modification of the SS model with transmissibilities for the saturation equation calculated according to eqns. (5.127). The line labelled SEQ Model II (Coats, 1976b) was obtained with a model described by Coats (1976), which uses eqn. (5.128) to calculate transmissibilities and incorporates the well model described in Section 9.8.1.2. Both sequential models give slightly lower WOR than the SS model. Figure 9.23 shows the time step sensitivity of SEQ Model I. Note that while WOR is smooth for time steps up to 75 days, the run with 100 days shows signs of some stability problems.

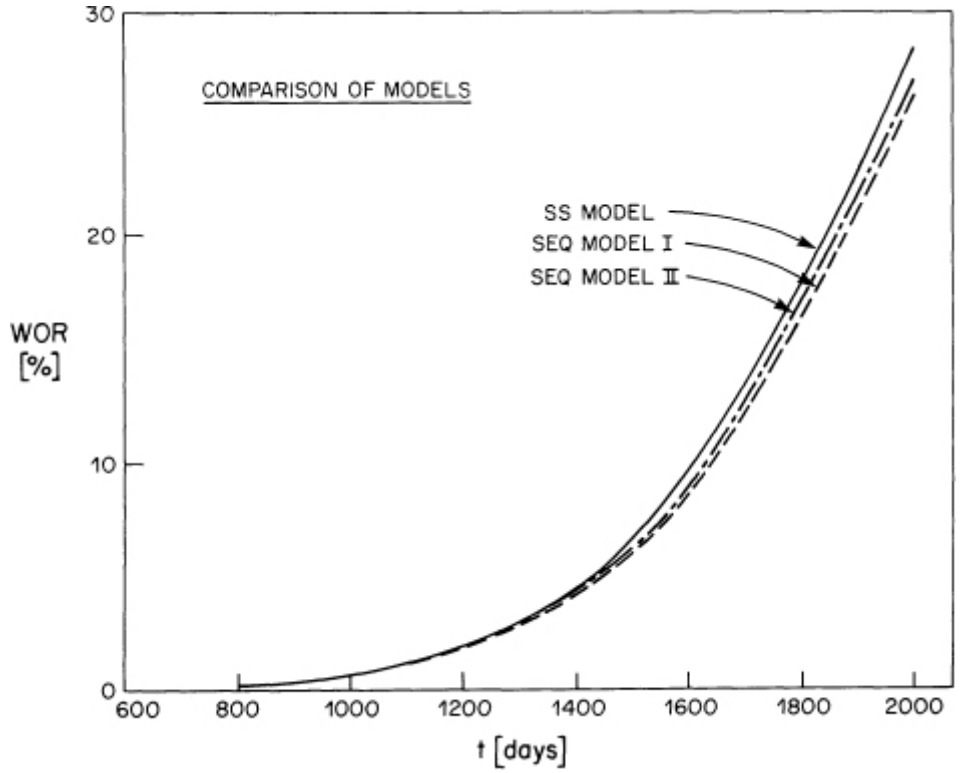


FIG. 9.22. Comparison of SS and SEQ models for problem of Blair and Weinaug.

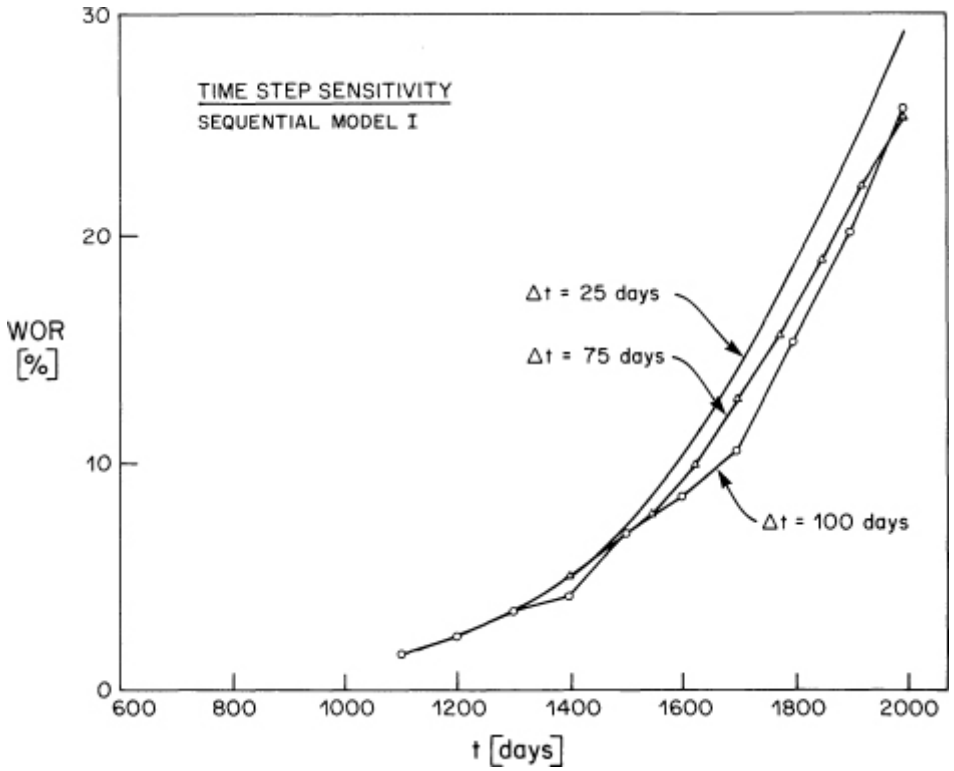


FIG. 9.23. Time-step sensitivity of SEQ Model I.

Ko (1977) has also presented results for this problem with a sequential model. His model is similar to the model of Settari and Aziz (1974a) and includes the well model of Section 9.8.1.1, modified for the sequential case. The results of his runs are smooth for Δt up to 100 days, but the time step sensitivity is larger than that shown in Fig. 9.23. Ko also found that in two-dimensional applications stability improved by dropping the nonlinear term that involves the product of saturation and capillary pressure derivatives.

9.8.3 Practical Considerations

Treatment of slopes in transmissibility coefficients. Some discussion of this problem is in Section 5.5 (Chapter 5). It is usually sufficient to use tangent slopes obtained from the k_r tables which will be piecewise constant if linear interpolation is used. When chord slopes are used, they should be taken in the direction of expected saturation change. This is particularly important when the saturation reaches the value at which one of the phases is immobile (in an o–w system $S_w = S_{wc}$ or $S_w = 1 - S_{oirr}$). Consider the case of water displacing oil. If $S_w = S_{wc}$, the chord of k_{rw} should be taken as non-zero, while when S_w reaches the maximum value, the slope of k_{ro} should be taken as zero (Fig. 9.24). Setting the slope to zero at S_{oirr} prevents overshoot because S_o cannot change. If the slope of k_{rw} is set to zero at S_{wc}, the water transmissibilities would

be treated, *explicitly* at that point with corresponding possible loss of stability. A similar problem arises when the saturations are close to limiting values. Coats (1976) suggested a method of computing slopes which prevents overshoot and Chappelear and Rogers (1974) describe the treatment of slopes for gas flow near S_{gc}.

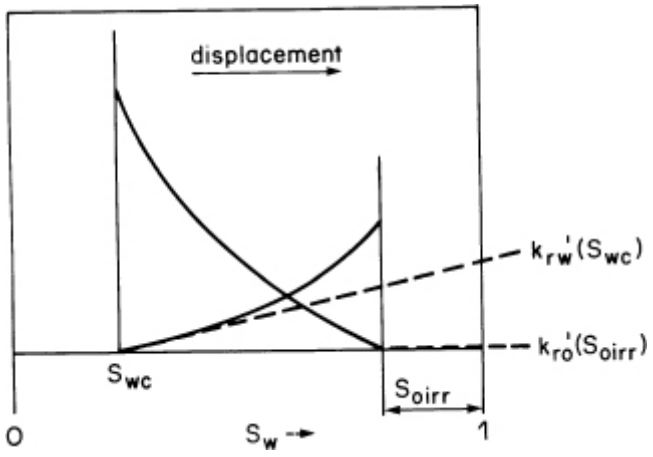


FIG. 9.24. Handling slopes of k_r curves at endpoints.

Handling of overshoot. This problem is related to time step selection and the treatment of slopes discussed above. It can be eliminated by decreasing the time step, but this may not be practical. Chappelear and Rogers (1974) discuss time step selection with respect to overshoot. Small overshoot can be tolerated; alternatively, the saturation may be reset to the correct value, which will incur a material balance error.

Other considerations. Several other practical considerations are discussed by Trimble and McDonald (1976). Their emphasis was on increased reliability which is achieved by implicit treatment of variables which are normally treated explicitly (such as pressure-dependent fluid properties) and use of Newton's iteration. It must be recognised that this reliability is gained at the expense of increased computing time per time step and the 'optimum' implicitness of the model depends on the problem solved.

9.9 CONCLUDING REMARKS

Many ideas developed in this chapter are common to several types of 2-D problems and also extend directly to 3-D problems and other types of models not discussed in this book. The division of the material in Sections 9.7 and 9.8 was chosen only to reflect typical situations. For example, the well model treated in Section 9.8.1 is equally important for cross-sectional and 3-D simulations as it is for the single-well case. The same remark applies to many topics of Chapter 12.

The current trend in simulation is towards general purpose models that can handle problems of various types. Spillette *et al*. (1973) claim that such a simulator can be based on sequential formulation. However, in our view, for problems of varying difficulty maximum efficiency cannot be achieved by a single method.

EXERCISE

Exercise 9.1

Derive eqn. (9.53) *for the Carter–Tracy method of water influx calculation.*

Solution Outline

For an infinite aquifer surrounding a cylindrical reservoir of a radius r_e, the dimensionless influence function can be written as

$$\Delta p = p_A^{(i)} - \bar{p}(t_D) = \frac{Q_A}{B} PI(t_D) \tag{A}$$

where

$$t_D = t\frac{6{\cdot}328k}{\phi c r_e^2} \qquad B = 1{\cdot}1191\phi chr_e^2$$

PI = terminal rate case influence function
c = total compressibility [1/psia]
k = permeability [darcy]
h = aquifer thickness [ft]

Carter and Tracy (1960) show that the cumulative influx can be approximately expressed as

$$W(t^{n+1}) - W(t^n) = \left[\frac{B\,\Delta p^{n+1} - W(t^n)PI'(t_D^{n+1})}{PI(t_D^{n+1}) - t_D^n PI'(t_D^{n+1})}\right](t_D^{n+1} - t_D^n) \tag{B}$$

This can be written in the form of eqn. (9.53) with

$$a = \frac{B\,\Delta p^n - W^n PI'^{n+1}}{PI^{n+1} - t_D^n PI'^{n+1}}\frac{\Delta t_D}{\Delta t} \tag{C}$$

$$b = \frac{-B}{PI^{n+1} - t_D^n PI'^{n+1}} \frac{\Delta t_D}{\Delta t} \tag{D}$$

CHAPTER 10

SOLUTION OF BLOCK PENTADIAGONAL EQUATIONS

10.1 INTRODUCTION

In Chapter 6 we showed how methods of Chapter 4 for tridiagonal matrix equations could be extended to solve block tridiagonal equations. In this chapter a parallel treatment is presented for the extension of methods of Chapter 8 for pentadiagonal equations to solve block pentadiagonal equations. Both direct and iterative methods will be considered.

The most general form of the system of L partial differential equations to be solved may be written as

$$\frac{\partial}{\partial x}\left[\lambda X_{lk}\frac{\partial U_k}{\partial x}\right]+\frac{\partial}{\partial y}\left[\lambda Y_{lk}\frac{\partial U_k}{\partial y}\right]=\beta_{lk}q_k+P_{lk}\frac{\partial U_k}{\partial t} \tag{10.1}$$

$$l,k=1,2,3,\ldots L$$

where a repeated subscript in any term indicates summation over all values of that subscript, e.g., for $L = 3$,

$$\frac{\partial}{\partial x}\left[\lambda X_{lk}\frac{\partial U_k}{\partial x}\right]=\frac{\partial}{\partial x}\left[\lambda X_{l1}\frac{\partial U_1}{\partial x}\right] \tag{10.2}$$

$$+\frac{\partial}{\partial x}\left[\lambda X_{l2}\frac{\partial U_2}{\partial x}\right]+\frac{\partial}{\partial x}\left[\lambda X_{l3}\frac{\partial U_3}{\partial x}\right] \qquad l=1,2,3$$

For any given problem many of the coefficients λX_{lk}, λY_{lk}, β_{lk} or P_{lk} may be zero. This is easily seen by comparing eqn. (10.1) with, for example, eqn. (9.3). It should also be noted that for two-phase flow indices l, k = 1, 2 only. The additional terms due to gravitational forces are left out here as they serve only to complicate the notation. In analogy with eqn. (8.3), the difference equations for a single grid point (i,j) may be written as:

$$\mathbf{c}_{ij}u^{n}_{i-1,j} + \mathbf{g}_{ij}u^{n}_{i,j-1} + \mathbf{a}_{ij}u^{n}_{ij} + \mathbf{f}_{ij}u^{n}_{i,j+1} + \mathbf{b}_{ij}u^{n}_{i+1,j} = \boldsymbol{d}_{ij} \tag{10.3}$$

The dimensionality of Q^{n+1}_{lk} will depend upon the number of partial differential equations to be solved simultaneously which is equal to L, the number of phases that must be handled simultaneously. All of the coefficients will be ($L \times L$) sub-matrices. In black-oil simulations L will be at most 3 for the SS method and 2 for the SEQ method. For example, for the SS method, the sub-matrices will be 3×3 for three-phase flow and 2×2 for two-phase flow.

The difference equation for this system of equations may be written in matrix form as

$$\mathbf{A}\boldsymbol{u} = \boldsymbol{d} \tag{10.4}$$

For example, when the natural ordering is used as shown in Fig. 8.1, the matrix **A** will take the form shown in Fig. 8.2. However, now each × in Fig. 8.2 represents either a (2 ×2) or a (3 ×3) sub-matrix depending upon the number of phases to be considered.

The vector $\boldsymbol{u}$ for the natural ordering with elements ordered in the x-direction first is defined by

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_{1,1} \\ \boldsymbol{u}_{2,1} \\ \vdots \\ \boldsymbol{u}_{1,2} \\ \boldsymbol{u}_{2,2} \\ \vdots \\ \boldsymbol{u}_{1,J} \end{bmatrix} \tag{10.5}$$

and the elements of $\boldsymbol{u}_{ij}$ contain values of the dependent variable at a grid point (i,j) for all phases, i.e., for $L = 3$,

$$\boldsymbol{u}_{ij} = \begin{bmatrix} u_{1\,ij} \\ u_{2\,ij} \\ u_{3\,ij} \end{bmatrix} \tag{10.6}$$

The vector $\boldsymbol{d}$ in eqn. (10.4) is defined in a similar fashion.

10.2 DIRECT METHODS

The form of eqn. (10.4) is the same as the form of eqn. (8.4), except for the fact that now all elements of the matrix **A** are (2 × 2) or (3 × 3) sub-matrices rather than scalars. This means that all direct methods of Chapter 8, Section 8.2 may be applied directly to eqn. (10.3) by simply replacing scalar operations in the elimination process by matrix operations. Since submatrices may contain a significant number of zeros in definite locations the solution algorithms should be developed to take advantage of this property. As indicated in Chapter 6 there are several choices available to us in the decomposition process and some of these are better as far as the round-off error is concerned. Actual comparison of work required for different ordering schemes is not available. Until such comparisons become available, the work estimates presented in Chapter 8 may be used. These estimates are approximately valid for multiphase flow if we replace I by IL and J by JL, and approach exact values for large I and J (Exercise 10.1). This implies in particular that asymptotic work ratios for D4 to standard ordering (Fig. 8.9) are also valid for multiphase problems.

10.3 ITERATIVE METHODS

The general discussion of iterative methods presented in Chapter 8, Section 8.3.3, applies directly to the solution of block-pentadiagonal equations. There are, however, more possibilities open to us now than for the solution of eqn. (8.5). For example, point-iterative methods discussed before may be generalised by replacing the scalar operations by operations with submatrices. A detailed discussion of this problem is, however, of academic interest only, since point-iterative methods turn out to be impractical for difficult reservoir simulation problems. Extension of block-iterative methods to handle eqn. (10.3) is of practical importance. In particular block SOR (BSOR), ADI and SIP are used extensively.

10.3.1 BSOR Method

Let us consider a simple example corresponding to the example of eqn. (8.52) for two-phase flow. The procedure described in Section 8.3.4 (Chapter 8) can be applied directly by simply noting that for each line an equation of the form

$$\mathbf{A}_i \boldsymbol{u}_i = \boldsymbol{d}_i \tag{10.7}$$

must be solved where $\mathbf{A}_i$ are now block-tridiagonal matrices requiring the methods discussed in Chapter 6. This method is also described by Björdammen and Coats (1969).

Thus, in principle, the extension of BSOR to multiphase problems is straightforward. The real difficulty arises in the estimation of ω_b. In the absence of any published results, the following guidelines are provided based upon intuitive arguments and experience:

1. When the simultaneous solution method is used with pressure of each phase being the dependent variables, then ω_b should be predicted by the same methods as for single-phase flow. However, the actual values of this iteration parameter may differ markedly from the predicted optimum values for single-phase problems (Chapter 8). For two-phase flow problems Björdammen and Coats (1969) find a value of $\omega = 1$ to be optimum (i.e., Gauss–Seidel method).

2. When simultaneous equations for two saturations are solved, the value of ω_b will be closer to 1 due to the hyperbolic character of the equations. The convergence rate is sensitive to the direction of sweep. As pointed out in Section 5.6.3 (Chapter 5), with $P_c = 0$ and flow in one direction only, the saturation equations can be ordered so that the coefficient matrix is triangular. Such a system can be solved directly without iterations by forward or backward substitutions. This substitution corresponds to one Gauss-Seidel sweep in the proper direction, which suggests that the ω_b should be close to 1 even in the general case.

3. When the simultaneous equations are for one pressure and one saturation, then ω_b should probably be between the two values computed for cases (1) and (2) above. It is also possible to use different values of ω in the equations for different phases. The ω_b for the saturation equation is expected to be very close to 1, while its range may be 1 to 2 for the pressure equation. No results are available in the literature for this approach.

In almost all cases some numerical experiments are required to arrive at an ω which is close to the optimum.

Woo and Emanuel (1976) have indicated good success with the LSOR method when applied in conjunction with a one-dimensional correction (1DC). However, their paper does not provide sufficient detail to be of much value.

10.3.2 Iterative ADI Method

Iterative ADI methods were discussed in Section 8.3.6 (Chapter 8). Elements of **V** and **H** matrices in, for example, eqn. (8.73a) now become (2 × 2) or (3 × 3) sub-matrices depending upon the number of equations being solved simultaneously. For every line in each sweep a block tridiagonal matrix equation of the form (10.7) must be solved.

The choice of an optimum set of iteration parameters and the choice of the number of iterations in a cycle is again most difficult. Douglas *et al*. (1959) have applied a

generalised form of the Peaceman and Rachford (1955) method for the solution of two simultaneous partial differential equations.

The Douglas *e t al*. (1959) method is obtained by generalising eqn. (8.77). The form of the equation is exactly the same and now all matrices have ($L \times L$) sub-matrices as elements. With the ordering of $\boldsymbol{u}$ given by eqn. (10.5) the ADI equations may be written as

$$\begin{aligned}(\mathbf{H} + \boldsymbol{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{u}^* &= (r^{(v+1)}\mathbf{D} - \mathbf{V})\boldsymbol{u}^{(v)} + \boldsymbol{d} \\ (\mathbf{V} + \boldsymbol{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{u}^{(v+1)} &= (r^{(v+1)}\mathbf{D} - \mathbf{H})\boldsymbol{u}^* + \boldsymbol{d}\end{aligned} \tag{10.8}$$

10.3.3 The SIP Method

Weinstein *et al*. (1970) have shown how the SIP method for a single partial differential equation proposed by Stone (1968) could be applied to a system of partial differential equations. As for other iteration procedures the extension of SIP discussed in Section 8.3.7 (Chapter 8) is straightforward. Elements in eqn. (8.80) are now (2×2) or (3×3) submatrices and division must be replaced by matrix inversion. Weinstein *et al*. provide details of the algorithm, and show how advantage may be taken of the zeros present in sub-matrices.

The iteration parameters are selected by the procedure discussed in Section 8.3.7 (Chapter 8). Equations (8.88) and (8.89) are used by giving due regard to the fact that λX_{lk} must be substituted for λX, etc. They recommend the use of 4–10 parameters in a cycle. If such a sequence causes divergence, $(1 - \alpha_{max})$ in eqn. (8.88) should be multiplied by a factor of 2–20. On the other hand, if the convergence is slow $(1 - \alpha_{max})$ should be divided by a factor of 2–20. These parameters are suitable for incompressible flow problems. For compressible problems an additional parameter equal to unity is added to the sequence. The use of the same parameter twice, as for single-phase flow equations (Chapter 8), has not been found to be desirable for multiphase problems. Thus one parameter is used only once in a cycle. Changing the number and sequence of parameters may have a significant effect on the convergence properties of the method. Similar observations are also reported by Suarez and Farouq Ali (1976) based on their tests with two-phase homogeneous and heterogeneous problems.

10.3.4 Comparison of Iterative Methods

Actual test results from the application of all of the important iterative methods to a standard set of test problems are not available.

A comparison of ADI and SOR methods is presented by Björdammen and Coats (1969). For oil–water pressure maintenance type problems (areal models) they found iterative ADI required only 45–75% of the work required by LSOR ($\omega = 1$ was used). However, LSOR was found to be slightly better than the ADI in the simulation of a gas–oil cross-sectional problem.

Weinstein *et al*. (1970) have presented comparisons of iterative ADI and SIP for three two-phase problems and very limited results for three three-phase problems. The three two-phase problems were: (1) gas–oil incompressible cross-section, (2) dissolved gas drive cross-section, and (3) water–oil radial coning. The three-phase problems considered were: (1) pressure maintenance, (2) laboratory model of a dissolved gas drive system, and (3) field dissolved gas drive. Based on their tests, the authors concluded that SIP required significantly less computer effort than iterative ADI for these difficult problems. As a matter of fact, for problems with high ratio of transmissibilities in the two directions, ADI did not converge in many cases while SIP did. It must be remembered that the SIP algorithm requires more computations per iteration than ADI. However, for difficult problems SIP more than makes up for this by requiring fewer iterations. These conclusions are consistent with the results reported in Chapter 8 for single equations (see for example, Fig. 8.20). Until more detailed results are available the material in Section 8.3.9 (Chapter 8) may provide some guidance.

10.4 COMPARISON OF ITERATIVE AND DIRECT METHODS

In the absence of published results comparing direct and iterative methods for coupled equations, we can only remark that many ideas discussed in Chapter 8, Section 8.4 should also be applicable to multiphase flow.

10.5 CONCLUDING REMARKS

For relatively easy reservoir simulation problems the simultaneous solution of flow equations is usually not necessary and material of this chapter is not relevant. For difficult problems where equations must be solved simultaneously reasonable success has been achieved with SIP and LSOR with 1DC. Detailed results of the experience with iterative methods are yet to be published.

Direct elimination with D4 ordering appears to be superior to all iterative methods for most practical problems provided sufficient computer core storage is available.

More work is required on comparative efficiency of various iterative methods for practical reservoir simulation problems.

EXERCISE

Exercise 10.1

Derive the work requirement for a block-five diagonal matrix equation with $L \times L$ blocks.

Solution Outline

The band-width of the $LN \times LN$ matrix **A** is $(2M + L)L + L - 1$ (or $2M + 1$ blocks) and eqn. (8.11) gives:

$$
\begin{aligned}
W_L &= \sum_{i=1}^{N-M} [(ML + L)^2 + (ML + L - 1)] \\
&\quad + \sum_{i=1}^{N-M} [(ML + L - 1)^2 + (ML + L - 2) + \cdots] \\
&\quad + \sum_{i=1}^{N-M} [(ML + 1)^2 + ML] + \sum_{i=1}^{ML} (i^2 + i - 1) \\
&= L(N - M)[(ML + 1)^2 + ML] \\
&\quad + (N - M)\{2ML[(L - 1) + (L - 2) \cdots + 1] \\
&\quad + [L^2 + (L - 1)^2 + \cdots + 1] - (L - 1)\} \\
&\quad + (N - M)[(L - 1) + (L - 2) + \cdots + 1] \\
&\quad + \frac{ML(ML + 1)(2ML + 1)}{6} + \frac{ML(ML - 1)}{3} \\
&= \frac{ML(ML + 1)(2ML + 1)}{6} + \frac{ML(ML - 1)}{2} \\
&\quad + L(N - M)[(ML + 1)^2 + ML] + (N - M) \\
&\quad \times \left\{ 2ML \frac{L(L - 1)}{2} + \frac{L(L + 1)(2L + 1)}{6} + \frac{L(L - 1)}{2} - (L - 1) \right\} \\
&= W + \Delta W
\end{aligned}
$$

where W is the same as derived in Exercise 8.1 (eqn. F), if we substitute ML for M, NL for N, and ΔW is a lower order term, i.e., $W_L \rightarrow W$ as N and M increase with fixed L.

CHAPTER 11

THREE-DIMENSIONAL PROBLEMS AND SOLUTION TECHNIQUES

11.1 INTRODUCTION

So far in this book we have shown how single-phase and multiphase flow models can be extended from one dimension (Chapters 3 and 5) to two dimensions (Chapters 7 and 9). As we will see in this chapter, extension of these models to three dimensions does not introduce any new concepts. The relationship between single-phase one-dimensional models and multiphase one-dimensional models was shown in going from Chapter 3 to Chapter 5. Similarly, multiphase two-dimensional problems were discussed in Chapter 9 by extending the ideas presented in Chapters 5 and 7. The matrix equations resulting from these models have also been discussed by starting with the simplest tridiagonal matrix equations resulting from single-phase one-dimensional flow in Chapter 4 to block tridiagonal matrix equations resulting from multiphase one-dimensional flow in Chapter 6, pentadiagonal matrix equations resulting from single-phase two-dimensional flow in Chapter 8 and block pentadiagonal matrix equations resulting from multiphase two-dimensional flow in Chapter 10. The material in Chapters 3–10 has therefore established what needs to be done in going from single phase to three-phase and from one dimension to several dimensions. For this reason it is not necessary to discuss the three-dimensional problems in the kind of detail presented for one- and two-dimensional problems.

In this chapter we will show how ideas presented so far can be easily extended to handle three-dimensional problems. This does not imply that three-dimensional problems are easy to solve. On the contrary, the practical solution of three-dimensional problems can be difficult and expensive. Some important aspects of three-dimensional simulation problems will be discussed in this chapter. Many three-dimensional problems can be adequately approximated by two-dimensional models through the concepts of pseudo-functions discussed in the next chapter. Some general aspects of three-dimensional problems are considered by Aziz (1968).

11.2 SINGLE PHASE FLOW

11.2.1 Basic Equation and Discretisation

The general partial differential equation for single-phase three-dimensional flow may be written in Cartesian co-ordinates as:

$$\frac{\partial}{\partial x}\left[\lambda X\left(\frac{\partial p}{\partial x}-\gamma\frac{\partial h}{\partial x}\right)\right]+\frac{\partial}{\partial y}\left[\lambda Y\left(\frac{\partial p}{\partial y}-\gamma\frac{\partial h}{\partial y}\right)\right] + \frac{\partial}{\partial z}\left[\lambda Z\left(\frac{\partial p}{\partial z}-\gamma\frac{\partial h}{\partial z}\right)\right]=\beta\frac{\partial p}{\partial t}+q \tag{11.1}$$

By an appropriate choice of λX, λY, λZ, β, γ and q the above equation can represent various models of gas or liquid flow as discussed in Chapter 2.

The flow equation (11.1) may also be written in cylindrical co-ordinates when detailed three-dimensional flow around a well is to be investigated (Exercise 11.1). In order to keep the discussion brief, only equations in Cartesian co-ordinates are considered here.

By analogy with eqn. (7.12) a finite-difference approximation for eqn. (11.1) is:

$$[\Delta \mathrm{T}\,\Delta(p-\gamma h)]_{ijk}=\frac{V_{ijk}\beta_{ijk}}{\Delta t}\Delta_t p_{ijk}+Q_{ijk} \tag{11.2}$$

$$i=1,2,\ldots,I$$
$$j=1,2,\ldots,J$$
$$k=1,2,\ldots,K$$

where

$$[\Delta \mathrm{T}\,\Delta(p-\gamma h)]_{ijk}=[\Delta_x \mathrm{T}X\,\Delta_x(p-\gamma h)+\Delta_y \mathrm{T}Y\,\Delta_y(p-\gamma h) + \Delta_z \mathrm{T}Z\,\Delta_z(p-\gamma h)]_{ijk} \tag{11.3}$$

$$V_{ijk}=\Delta x_i\,\Delta y_j\,\Delta z_k$$

is the volume of block (i, j, k) and

$$Q_{ijk}=V_{ijk}q_{ijk}$$

The first term on the right side of eqn. (11.3) may be written in expanded form as:

$$[\Delta_x \mathrm{T}X\,\Delta_x(p-\gamma h)]_{ijk}=\mathrm{T}X_{(i+1/2),j,k}[p_{i+1}-p_i-\gamma_{i+1/2}(h_{i+1}-h_i)]_{jk} + \mathrm{T}X_{(i-1/2),j,k}[p_{i-1}-p_i-\gamma_{i-1/2}(h_{i-1}-h_i)]_{jk}$$

Definitions of the second and third terms on the right side of eqn. (11.3) may be written immediately by analogy with the above equation. Coefficients TX, TY and TZ may be defined by following the procedure used for two-dimensional problems in Chapter 9 (Section 9.3.1). The coefficients for the cylindrical case are considered in Exercise 11.1.

When all of the terms in eqn. (11.2) are expanded and the coefficients of p_{ijk} are collected the resulting equation can be written in the following form if subscripts i, j, and k are all suppressed:

$$zp_{k-1} + gp_{j-1} + cp_{i-1} + ap + bp_{i+1} + fp_{j+1} + sp_{k+1} = d \tag{11.4}$$

This difference equation can be written for each grid point and the resulting equations may be collected and expressed in matrix form as

$$\mathbf{A}\boldsymbol{p} = \boldsymbol{d} \tag{11.5}$$

There are seven unknowns in a difference equation for an interior point, hence the matrix contains seven diagonals as shown in Fig. 11.1. For this case the points (i, j, k) of the grid are ordered in sequence such that i is swept through first (i = 1, 2, …, I), j second (j = 1, 2, …, J) and k last (k = 1,2, …, K). The elements of vector for this ordering are:

$$\boldsymbol{p} = \begin{bmatrix} p_{1,1,1} \\ p_{2,1,1} \\ \vdots \\ p_{I,1,1} \\ p_{1,2,1} \\ p_{2,2,1} \\ \vdots \\ p_{1,J,K} \\ \vdots \\ p_{I-1,J,K} \\ p_{I,J,K} \end{bmatrix} \tag{11.6}$$

It is possible to solve eqn. (11.5) by both direct and iterative methods. These methods are direct extensions of the methods presented in Chapter 8. Some of the more important techniques are considered next.

11.2.2 Special Methods for 3-D Problems

In Section 7.9 (Chapter 7) we considered some special methods for two-dimensional problems known as ADI and ADE methods. Both types of methods can be readily extended to three-dimensional problems. However, the application of ADE methods to 3-D reservoir simulation problems is of little practical importance and they will not be considered. Relatively simple 3-D problems may be solved by the non-iterative ADI methods discussed below.

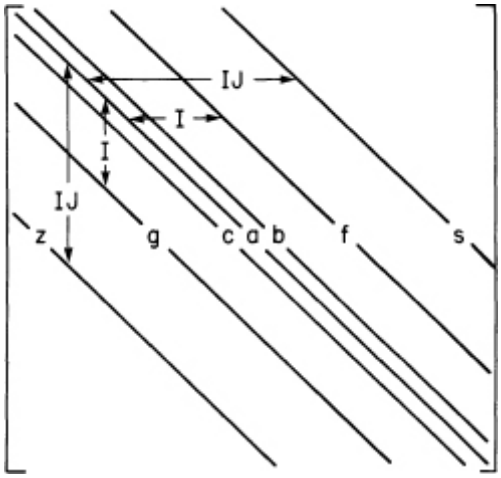


FIG. 11.1. The form of matrix **A** in eqn. (11.5).

11.2.2.1 *ADI Methods*

As in Chapter 7, consider the backward-difference approximation of eqn. (11.1) without the gravity terms. The resulting equation may be written as:

$$\Delta_x \mathrm{T} X \Delta_x p^{n+1} + \Delta_y \mathrm{T} Y \Delta_y p^{n+1} + \Delta_z \mathrm{T} Z \Delta_z p^{n+1} = \varphi(p^{n+1} - p^n) + Q \qquad (11.7)$$

In order to keep the notation as simple as possible subscripts *i,j,k* have been suppressed wherever appropriate in this chapter.

The direct extension of the Peaceman and Rachford (1955) method for this problem turns out to be only conditionally stable even for the linear problem (Douglas, 1961). Some methods that are unconditionally stable for the linear problem are given below for eqn. (11.7).

Douglas and Rachford (1956) *method*. The method is defined by the following equations:

$$\Delta_x T_X \Delta_x p^* + \Delta_y T_Y \Delta_y p^n + \Delta_z T_Z \Delta_z p^n = \varphi(p^* - p^n) + Q \tag{11.8}$$
$$\Delta_y T_Y \Delta_y p^{**} = \Delta_y T_Y \Delta_y p^n + \varphi(p^{**} - p^*)$$
$$\Delta_z T_Z \Delta_z p^{n+1} = \Delta_z T_Z \Delta_z p^n + \varphi(p^{n+1} - p^{**})$$

This method is a perturbation of the backward-difference approximation.

Brian (1961) *method*. Brian used an extrapolation approach to develop a method which is a perturbation of the Crank–Nicolson method:

$$\Delta_x T_X \Delta_x p^* + \Delta_y T_Y \Delta_y p^n + \Delta_z T_Z \Delta_z p^n = 2\varphi(p^* - p^n) + Q \tag{11.9}$$
$$\Delta_y T_Y \Delta_y p^{**} = \Delta_y T_Y \Delta_y p^n + 2\varphi(p^{**} - p^*)$$
$$\Delta_z T_Z \Delta_z p^{n+1/2} = \Delta_z T_Z \Delta_z p^n + 2\varphi(p^{n+1/2} - p^{**})$$
$$p^{n+1} = p^n + 2(p^{n+1/2} - p^n)$$

Douglas (1962) *method*. Douglas developed an ADI method directly from the Crank–Nicolson approximation. A form of this method which is convenient for computations is given below:

$$\tfrac{1}{2}[\Delta_x T_X \Delta_x (p^* + p^n)] + \Delta_y T_Y \Delta_y p^n + \Delta_z T_Z \Delta_z p^n = \varphi(p^* - p^n) + Q \tag{11.10}$$
$$\tfrac{1}{2}[\Delta_y T_Y \Delta_y (p^{**} - p^n)] = \varphi(p^{**} - p^*)$$
$$\tfrac{1}{2}[\Delta_z T_Z \Delta_z (p^{n+1} - p^n)] = \varphi(p^{n+1} - p^{**})$$

This method is another perturbation of the Crank–Nicolson method.

The evaluation of nonlinear terms should be at the $n + 1$ level for the Douglas–Rachford method and n + sup(1)/sub(2) level for the other two methods.

Because of rather limited practical use of these methods for reservoir simulation problems, they will not be discussed any further.

11.2.3 Direct Methods of Solution

The efficient application of direct elimination with standard ordering shown in Fig. 11.1 and alternate point ordering requires that the points be ordered in the shortest direction first. Therefore if

$$I \geq J \geq K$$

the points must be ordered first in the k direction followed by j and i, to minimise work. If these rules are observed, the maximum band-width for standard ordering will be JK and the work and storage estimates for large I, J and K will be (Price and Coats, 1974):

$$W_1 = IJ^3K^3 \tag{11.11}$$

$$S_1 = IJ^2K^2 \tag{11.12}$$

For 2-cyclic matrices alternating point ordering yields

$$W_4 = \frac{IJ^3K^3}{2} \tag{11.13}$$

$$S_4 = \frac{IJ^2K^2}{2} \tag{11.14}$$

For example, the ordering given by eqn. (11.6) is optimal if $I \leq J \leq K$.

Other schemes discussed in Chapter 8 could also be extended to three-dimensional problems. For example, the alternating diagonal lines scheme D4 now becomes an alternating diagonal planes scheme. This scheme has been discussed by Price and Coats (1974) and here we summarise their results:

Let m represent a diagonal plane where all grid points on the plane satisfy

$$i + j + k = m \qquad m = 3, 4, \ldots, M$$

and

$$M = I + J + K$$

If M is even then the planes should be chosen in the order

$$3, 5, 7, \ldots, M-1, 4, 6, 8, \ldots, M$$

If M is odd then the order should be

$$3, 5, 7, \ldots, M, 4, 6, \ldots, M-1$$

The points in a plane m should be numbered in order of decreasing k and for each constant value of k in order of decreasing j and increasing i, provided $I \geq J \geq K$. The work and storage estimates for three different cases are summarised in Table 11.1. This table also provides a comparison of D4 ordering and standard ordering.

The values of f for D4 ordering vary between 0•171 and 0•5.

In order to present the results of Table 11.1 graphically, it is necessary to introduce some additional assumptions. For the second case in the table we assume that $J \simeq K$. Then the work ratio may be written as

$$f = 0{\cdot}5 - \frac{14}{40r}$$

where $r = I/J$ (since $I - J \geq K$ for this case, $r \geq 2$). Now we can plot the work ratios for three-dimensional problems as we did for the two-dimensional case in Fig. 8.8. However, for three-dimensional problems each of the three cases of Table 11.1 must be considered separately. The results are shown in Fig. 11.2. Discussion in this section applies to equations in Cartesian co-ordinates. Similar results, with some important differences, can be developed for the cylindrical co-ordinate case (Exercise 11.2).

TABLE 11.1

WORK ESTIMATES FOR D4 ORDERING AND COMPARISON WITH STANDARD ORDERING, BASED ON PRICE AND COATS (1974)

Case	*Approximation for D4 work*	*Work ratio of D4 to standard ordering, f*
$I = J = K$	$\frac{23}{90}I^6 + \frac{6}{35}I^7$	$0{\cdot}171 + \frac{0{\cdot}582}{I}$
$I - J \geq K$	$\frac{IJ^3K^3}{2} - \frac{J^4K^3}{4} - \frac{J^2K^5}{8} + \frac{JK^6}{40}$	$\frac{1}{2} - \frac{1}{4}\frac{J}{I} - \frac{1}{8}\frac{K^2}{IJ} + \frac{K^3}{40IJ^2}$
$I = J, J = IK$	$\left(\frac{I^4}{4} - \frac{I^2}{8} + \frac{I}{20} - \frac{1}{280}\right)K^7$	$\frac{1}{4} - \frac{1}{8I^2} + \frac{1}{20I^3} - \frac{1}{280I^4}$

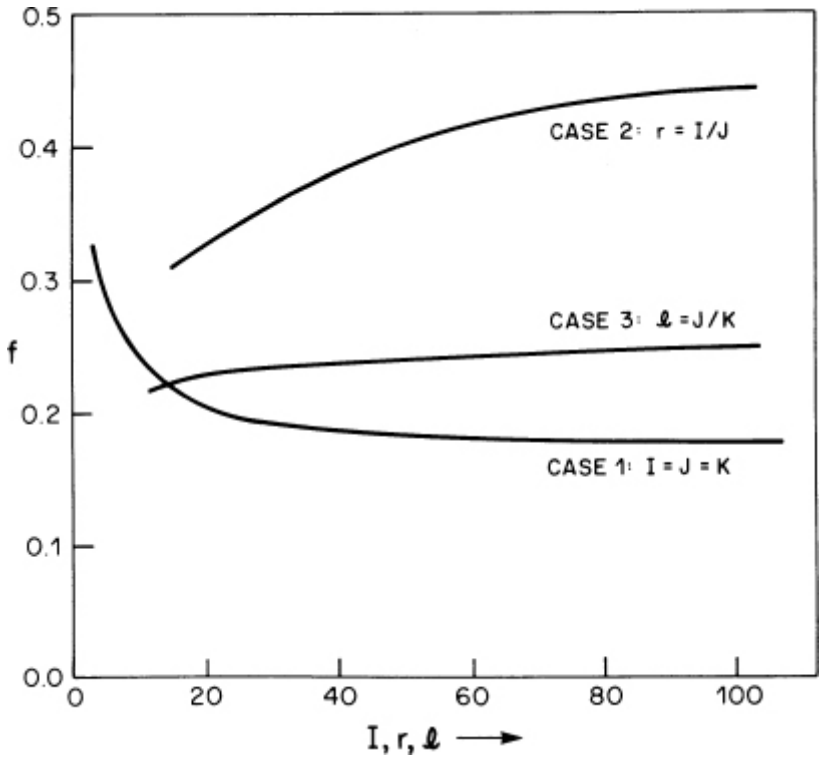


FIG. 11.2. Work ratio *f* for different types of three-dimensional problems.

11.2.4 Iterative Methods

Three classes of iterative methods have achieved various levels of success in the solution of three-dimensional problems. These are: (1) BSOR (or LSOR) with or without correction, (2) iterative ADI, and (3) SIP. A brief discussion of these methods follows.

11.2.4.1 *BSOR and Related Methods*

Several choices are open in the use of BSOR for three-dimensional problems. We could consider the direct application of LSOR. The lines must be chosen to minimise ω_b, the optimum value of ω. The smallest value of ω_b usually results if the lines are chosen in the direction of the largest transmissibilities as for two-dimensional problems (see Section 8.3.9, Chapter 8). This is almost always the vertical direction.

BSOR could also be used by considering unknowns on a single plane at one time. Let us collect all values of p_{ijk} for a given value of k in a vector $\boldsymbol{p}_k$. Then the vector $\boldsymbol{p}$ given by eqn. (11.6) may be written as

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_k \\ \vdots \\ \boldsymbol{p}_K \end{bmatrix} \tag{11.15}$$

where for natural ordering

$$\boldsymbol{p}_k = \begin{bmatrix} p_{1,1,k} \\ p_{2,1,k} \\ \vdots \\ p_{1,2,k} \\ \vdots \\ p_{I,J,k} \end{bmatrix} \tag{11.16}$$

Equation (11.5) may now be written as

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & & & & & \\ \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B}_2 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \mathbf{C}_K & \mathbf{A}_K \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{p}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{d}_K \end{bmatrix} \tag{11.17}$$

This is similar in form to eqn. (8.52) except now $\mathbf{A}_k$ are pentadiagonal submatrices. The form of $\mathbf{A}_k$ will depend upon the ordering of unknowns on a plane as in the case of two-dimensional problems (Chapter 8, Sections 8.1 and 8.2). For example if natural ordering is used, $\mathbf{A}_k$ will take the form shown in Figs. 8.12. The equations for the computation of $(\nu + 1)$ level of iteration from (ν) level by the BSOR method, that solves for unknowns on one plane simultaneously, may be written as

$$\mathbf{A}_k \boldsymbol{p}_k^* = -\mathbf{C}_k \boldsymbol{p}_{k-1}^{(\nu+1)} - \mathbf{B}_k \boldsymbol{p}_{k+1}^{(\nu)} + \boldsymbol{d}_k \tag{11.18a}$$

$$\boldsymbol{p}_k^{(\nu+1)} = \omega \boldsymbol{p}_k^* + (1 - \omega)\boldsymbol{p}_k^{(\nu)} \tag{11.18b}$$

$$k = 1, 2, \ldots, K$$

provided we set $\boldsymbol{p}_0 = \boldsymbol{p}_{K+1} = \mathbf{0}$. This equation is obtained in direct analogy with eqn. (8.54). Because the above equations are solved repetitively, it is extremely important to solve the matrix problem in eqn. (11.18a) efficiently. Any of the fast direct methods considered in Section 8.2 (Chapter 8) may be applied here. If, in addition, the matrices obtained after the forward pass of elimination are stored, only the backward substitution is necessary in the second and successive iterations. This approach is possible with standard elimination as well as with ordering schemes and it is essential for the efficient use of BSOR. The optimum value of ω may be computed by the methods discussed in Chapter 8 (Section 8.3.3).

BSOR with vertical planes using D4 ordering for eqn. (11.18a) has been used for simulation problems by Wattenbarger and Thurnau (1976). They have also applied this method with two-dimensional correction (2DC) and found that the use of 2DC was necessary to obtain good convergence for heterogeneous problems.

11.2.4.2 *Iterative ADI Methods*

As in the case of two-dimensional problems (Section 8.3.6, Chapter 8) there are many versions of the iterative ADI methods that can be used for three-dimensional problems. The matrix equation (11.5) resulting from a seven-point difference formula (11.2) may be written as

$$(\mathbf{X} + \mathbf{Y} + \mathbf{Z} + \mathbf{\Sigma})\boldsymbol{p} = \boldsymbol{d} \tag{11.19}$$

where the elements of $\mathbf{X}$ result from $\Delta_x T_X \Delta_x p$, the elements of $\mathbf{Y}$ result from $\Delta_y T_Y \Delta_y p$, the elements of $\mathbf{Z}$ result from $\Delta_z T_Z \Delta_z p$, and the elements of $\mathbf{\Sigma}$ result from the time derivative. By a suitable ordering of $\boldsymbol{p}$ any of $\mathbf{X}$, $\mathbf{Y}$, or $\mathbf{Z}$ can be transformed into a block-diagonal matrix with each block being a tridiagonal $(I \times I)$, $(J \times J)$, or $(K \times K)$ matrix.

We can now generalise any of the iterative ADI methods discussed in Chapter 8. For example, a method corresponding to Varga's modification of the Peaceman–Rachford method (eqn. 8.77) may be written as:

$$\begin{aligned}
(\mathbf{X} + \mathbf{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{*} &= (r^{(v+1)}\mathbf{D} - \mathbf{Y} - \mathbf{Z})\boldsymbol{p}^{(v)} + \boldsymbol{d} \\
(\mathbf{Y} + \mathbf{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{**} &= (r^{(v+1)}\mathbf{D} - \mathbf{Z})\boldsymbol{p}^{(v)} - \mathbf{X}\boldsymbol{p}^{*} + \boldsymbol{d} \\
(\mathbf{Z} + \mathbf{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{(v+1)} &= r^{(v+1)}\mathbf{D}\boldsymbol{p}^{(v)} - \mathbf{X}\boldsymbol{p}^{*} - \mathbf{Y}\boldsymbol{p}^{**}
\end{aligned} \tag{11.20}$$

As in the case of two-dimensional problems $\mathbf{D}$ is a diagonal matrix containing the sum of the diagonal elements of $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$.

Douglas (1962) has proposed another method (Varga, 1962, p. 244) that can be modified for parabolic systems and expressed as

$$(\mathbf{X} + \boldsymbol{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{*} = (r^{(v+1)}\mathbf{D} - \mathbf{X} - 2\mathbf{Y} - 2\mathbf{Z})\boldsymbol{p}^{(v)} + 2\boldsymbol{d} \tag{11.21}$$

$$(\mathbf{Y} + \boldsymbol{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{**} = (r^{(v+1)}\mathbf{D} - \mathbf{X} - \mathbf{Y} - 2\mathbf{Z})\boldsymbol{p}^{(v)} - \mathbf{X}\boldsymbol{p}^{*} + 2\boldsymbol{d}$$

$$(\mathbf{Z} + \boldsymbol{\Sigma} + r^{(v+1)}\mathbf{D})\boldsymbol{p}^{(v+1)} = (r^{(v+1)}\mathbf{D} - \mathbf{X} - \mathbf{Y} - \mathbf{Z})\boldsymbol{p}^{(v)} - \mathbf{X}\boldsymbol{p}^{*} - \mathbf{Y}\boldsymbol{p}^{**} + 2\boldsymbol{d}$$

The above method reduces to the Douglas (1962) method when $\boldsymbol{\Sigma} = \mathbf{0}$ (elliptic equations) and $\mathbf{D} = \mathbf{I}$.

There is very little reported in the literature on the selection of iteration parameters for these methods. A reasonable approach is to extend eqn. (8.77) for the estimation of $r_{\min}$ and then calculate parameters in a geometric sequence between $r_{\min}$ and $r_{\max} = 1$ to 2. The equation for $r_{\min}$ is (Weinstein *et al*., 1969):

$r_{\min}$ = minimum over the grid of

$$\left[\frac{\pi^2}{2I^2(1+\rho_1)}, \quad \frac{\pi^2}{2J^2(1+\rho_2)}, \quad \frac{\pi^2}{2K^2(1+\rho_3)}\right] \tag{11.22}$$

Where

$$\rho_1 = \frac{\lambda Y}{\lambda X}\left(\frac{\Delta x}{\Delta y}\right)^2 + \frac{\lambda Z}{\lambda X}\left(\frac{\Delta x}{\Delta z}\right)^2$$

$$\rho_2 = \frac{\lambda X}{\lambda Y}\left(\frac{\Delta y}{\Delta x}\right)^2 + \frac{\lambda Z}{\lambda Y}\left(\frac{\Delta y}{\Delta z}\right)^2$$

$$\rho_3 = \frac{\lambda X}{\lambda Z}\left(\frac{\Delta z}{\Delta x}\right)^2 + \frac{\lambda Y}{\lambda Z}\left(\frac{\Delta z}{\Delta y}\right)^2$$

The values of $\rho_p = 0$ and ∞ ($p = 1, 2, 3$) should not be considered in the evaluation of the minimum.

11.2.4.3 *The SIP Method*

This method has already been discussed for single-phase two-dimensional flow in Chapter 8 (Section 8.3.7) and for multiphase two-dimensional flow in Chapter 10 (Section 10.3.3). The extension of SIP to three-dimensional problems was proposed by Weinstein *et al*. (1969). As in the case of two-dimensional problems **A** is modified by

N such that **A** + **N** is factorable into **L** and **U** matrices which are sparse and of the form shown in Figs. 11.3 and 11.4. Equation (11.5) can be written in residual form as

$$(\mathbf{A} + \mathbf{N})\boldsymbol{\delta}^{(\nu+1)} \equiv (\mathbf{LU})\boldsymbol{\delta}^{(\nu+1)} = -\boldsymbol{R}^{(\nu)} \tag{11.23}$$

where

$$\boldsymbol{R}^{(\nu)} = \mathbf{A}\boldsymbol{p}^{(\nu)} - \boldsymbol{d} \tag{11.24a}$$

$$\boldsymbol{\delta}^{(\nu+1)} = \boldsymbol{p}^{(\nu+1)} - \boldsymbol{p}^{(\nu)} \tag{11.24b}$$

The expressions for the elements of **L** and **U** are developed in a manner similar to the procedure used in Section 8.3.7 (Chapter 8). The final algorithm as reported by Weinstein *et al*. (1969) is:

$$\begin{aligned}
\mathrm{z}_{ijk} &= z_{ijk}[1 + \alpha(\mathrm{b}_{i,j,k-1} + \mathrm{f}_{i,j,k-1})^{-1}] \\
\mathrm{g}_{ijk} &= g_{ijk}[1 + \alpha(\mathrm{s}_{i,j-1,k} + \mathrm{b}_{i,j-1,k})^{-1}] \\
\mathrm{c}_{ijk} &= c_{ijk}[1 + \alpha(\mathrm{s}_{i-1,j,k} + \mathrm{f}_{i-1,j,k})^{-1}] \\
A_{ijk} &= \mathrm{z}_{ijk}\mathrm{b}_{i,j,k-1} \\
C_{ijk} &= \mathrm{g}_{ijk}\mathrm{b}_{i,j-1,k} \\
G_{ijk} &= \mathrm{c}_{ijk}\mathrm{f}_{i-1,j,k} \\
W_{ijk} &= \mathrm{c}_{ijk}\mathrm{s}_{i-1,j,k} \\
T_{ijk} &= \mathrm{z}_{ijk}\mathrm{f}_{i,j,k-1} \\
U_{ijk} &= \mathrm{g}_{ijk}\mathrm{s}_{i,j-1,k} \\
\mathrm{a}_{ijk} &= a_{ijk} + \alpha[A_{ijk} + C_{ijk} + G_{ijk} + W_{ijk} + T_{ijk} + U_{ijk}] \\
&\quad - \mathrm{c}_{ijk}\mathrm{b}_{i-1,j,k} - \mathrm{g}_{ijk}\mathrm{f}_{i,j-1,k} - \mathrm{z}_{ijk}\mathrm{s}_{i,j,k-1} \\
\mathrm{b}_{ijk} &= \mathrm{a}_{ijk}^{-1}[b_{ijk} - \alpha(A_{ijk} + C_{ijk})] \\
\mathrm{f}_{ijk} &= \mathrm{a}_{ijk}^{-1}[f_{ijk} - \alpha(T_{ijk} + G_{ijk})] \\
\mathrm{s}_{ijk} &= \mathrm{a}_{ijk}^{-1}[s_{ijk} - \alpha(W_{ijk} + U_{ijk})]
\end{aligned} \tag{11.25}$$

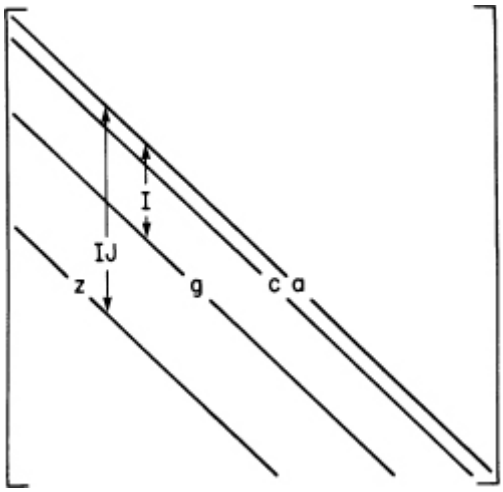


FIG. 11.3. MATRIX **L**.

After the elements of **L** and **U** are computed, vector $\boldsymbol{v}$ is obtained from

$$\mathbf{L}\boldsymbol{v} = -\boldsymbol{R}^{(\nu)} \tag{11.26}$$

by forward substitution by solving

$$v_{ijk} = \mathrm{a}_{ijk}^{-1}[\mathrm{R}_{ijk}^{(\nu)} - \mathrm{z}_{ijk}^{-1} v_{i,j,k-1} - \mathrm{g}_{ijk} v_{i,j-1,k} - \mathrm{c}_{ijk} v_{i-1,j,k}] \tag{11.27}$$

The vector $\boldsymbol{\delta}$ is obtained from

$$\mathbf{U}\boldsymbol{\delta} = \boldsymbol{v} \tag{11.28}$$

by backward substitution by solving

$$\delta_{ijk}^{(\nu+1)} = v_{ijk} - \mathrm{b}_{ijk}\,\delta_{i+1,j,k}^{(\nu+1)} - \mathrm{f}_{ijk}\,\delta_{i,j+1,k}^{(\nu+1)} - \mathrm{s}_{ijk}\,\delta_{i,j,k+1}^{(\nu+1)} \tag{11.29}$$

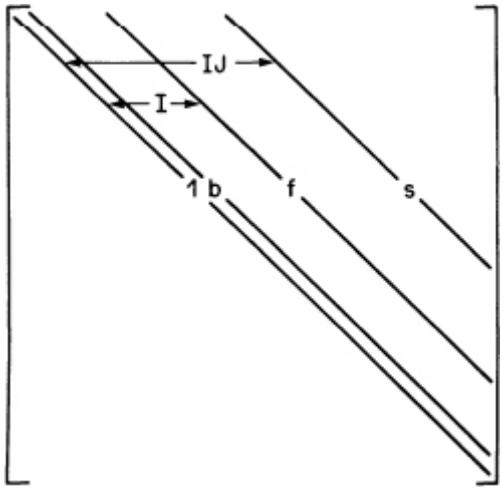


FIG. 11.4. Matrix **U**.

Equations (11.25), (11.27) and (11.29) constitute the 3-D SIP algorithm provided we set

$$z_{ij1} = s_{ijK} = 0 \qquad i = 1, 2, \ldots, I \qquad j = 1, 2, \ldots, J$$
$$g_{i1j} = f_{iJk} = 0 \qquad i = 1, 2, \ldots, I \qquad k = 1, 2, \ldots, K$$
$$c_{1jk} = b_{Ijk} = 0 \qquad j = 1, 2, \ldots, J \qquad k = 1, 2, \ldots, K$$

The algorithm can be further generalised by using different values of iteration parameters in each direction. The generalised algorithm has been presented by Weinstein *et al*. (1969), but no results have been reported with this form of SIP.

As in the case of 2-D SIP, Weinstein *et al*. recommended that the above ordering of unknowns be used for odd-numbered iterations and for even-numbered iterations this sequence be altered to the following:

$$i = 1, 2, \ldots, I$$
$$j = J, J-1, \ldots, 1$$
$$k = K, K-1, \ldots, 1$$

The changing of sequence improves the rate of convergence. Use of other independent orderings (two are possible in this case) is not necessary according to Weinstein *et al*. (1969).

The iteration parameters are estimated by setting the right side of eqn. (11.22) equal to $(1 - \alpha_{max})$ and then using eqn. (8.89) to calculate α_m. Generally the number of parameters per cycle (M) is between 4 and 10. If the sequence calculated above causes divergence, $(1 - \alpha_{max})$ calculated should be multiplied by a factor of 2–10 and if iterations converge but slowly, it should be divided by a factor of 2–10.

11.2.5 Comparison of Methods

In Sections 9.2.2, to 9.2.4 we have discussed three classes of methods: (1) non-iterative ADI, (2) direct, and (3) iterative. At first glance, non-iterative ADI methods may appear attractive for reservoir simulation because of their simplicity and relatively small amount of computer work required per time step. This, however, is not always the case. Actual tests with the method indicate that unreasonably small time steps may have to be used (Briggs and Dixon, 1968).

When the matrix eqn. (11.5) is to be solved, one may choose either a direct or an iterative method. For sufficiently large problems a suitably selected iterative method

will require less work. The cross-over point depends upon the choice of methods being considered and the convergence criterion for the iterative method.

Price and Coats (1974) present results which can be of value in deciding which method should be used. We will present a summary of their results here. Some of the discussion also applies to two-dimensional problems discussed in Chapters 8 and 10.

The work of an iterative method may be expressed as

$$W_{it} = cN_iIJK \tag{11.30}$$

where c is the number of multiplications and divisions per iteration per grid block and N_i is the number of iterations required for convergence. The work of direct methods may be approximated by the expression

$$W = fI(JK)^3 \tag{11.31}$$

where the coefficient $f = 1$ for the natural ordering and it lies between 0•17 and 0•5 for D4 ordering.

The work ratio may be approximated by

$$\frac{W}{W_{it}} = \frac{f(JK)^2}{cN_i} = \frac{fw^2}{cN_i} \tag{11.32}$$

where $w = JK$ is the nominal band-width of the matrix ($w = J$ for two-dimensional problems). The critical band-width, w_c, at which $W = W_{it}$ is

$$w_c = (cN_i/f)^{1/2} \tag{11.33}$$

For example, this value of w_c for a two-dimensional problem is shown as the cross-over point in Fig. 8.26. This cross-over point obviously depends upon the methods used and the acceptable tolerance for the iterative methods. Direct methods with standard ordering cannot usually compete with iterative methods for the solution of three-dimensional problems. However, the work of Price and Coats shows that with D4 ordering direct methods require less work than iterative methods for many practical problems.

It is possible to get a rough estimate of w_c from eqn. (11.33) by computing f from the formulae in Table 11.1 or from Fig. 11.2. The value of constant c may be estimated from Table 11.2 where for comparison purposes the values of this constant for two-dimensional problems are also provided. The number of iterations required for convergence is a strong function of the problem being solved, the method used and the acceptable tolerance.

Price and Coats (1974) have compared iterative ADI, LSOR, SIP and direct elimination with D4 ordering for three practical three-dimensional problems. They found the D4 scheme to be very competitive with iterative methods, and faster than iterative methods for full systems (no inactive blocks) with nominal band-widths of up to 38. For most practical simulation problems there are usually a large number of inactive blocks. For such cases, the D4 scheme was found to be faster for nominal band-widths as large as 72. Direct methods become even more efficient for problems where it is difficult to obtain good iteration parameters, provided sufficient computer core storage is available.

TABLE 11.2

NUMBER OF MULTIPLICATIONS AND DIVISIONS PER ITERATION PER GRID POINT FOR VARIOUS ITERATIVE METHODS (FROM PRICE AND COATS, 1974)

Iterative method	*Value of c*	
	Two-dimensional problems	*Three-dimensional problems*
SIP	24	37
ADI	19	28
LSOR	9	11

11.3 MULTIPHASE FLOW

11.3.1 Basic Solution Methods and Their Work Requirements

The solution methods presented in Chapter 5 (IMPES, SS and SEQ) are extended to three dimensions in exactly the same way as we have done in Chapter 9 for 2-D problems. The only difference is in the additional flow terms in the third spatial dimension.

The work requirements per time step increase in 3-D exactly as in 2-D when going from IMPES to SS method. If we assume that eqn. (9.28) gives the work estimate, then for $I \geq J \geq K$ we can obtain the 3-D estimates from those listed in Tables 9.1 and 9.2, if we multiply B and N by K and W_{TS} by K^3. Therefore, the work ratios relative to the work for IMPES are the same for 2-D and 3-D problems.

It follows immediately that the ratio of work for 3-D and 2-D problems having the same I and J is

$$W_{3D}/W_{2D} = K^3 \tag{11.34}$$

i.e., the work increases with the third power of the number of layers. Similarly, we can deduce the storage requirements by multiplying values in Tables 9.3 and 9.4 by K^2. Therefore, for a direct method,

$$S_{3D}/S_{2D} = K^2 \tag{11.35}$$

The above discussion is valid only when the same method for solving the matrix equations is used for all cases. In practice, this is often not the case. For example, in the case of the SEQ method for three-phase flow, the pressure equation may be solved by a direct method and the saturation equation by an iterative method. This may be more effective than solving both equations either directly or iteratively. The work ratios of W_{SEQ}/W_{IMPES} will then be different from those given in Tables 9.1 and 9.2. As another example, a problem which will be solved by LSOR in two dimensions may be solved more efficiently by BSOR in 3-D. The improved rate of convergence will then lead to the ratio of work different from eqn. (11.34). Wattenbarger and Thurnau (1976) reported that if vertical planes are used for BSOR, then the ratio of work for a 3-D and 2-D areal problem solved by the IMPES method is

$$W_{3D}/W_{2D} = CK \tag{11.36}$$

where C $\simeq$ (2•12 to 2•5) N_{3D}/N_{2D} and N is the number of iterations necessary to solve the problem. The ratio N_{3D}/N_{2D} will be close to 1 for homogeneous problems and increase for heterogeneous problems.

11.3.2 Methods for Solving the Matrix Equations

11.3.2.1 *Solution Methods*

The only difference between the matrix problems for 3-D as opposed to 2-D problems is that the matrices are block-seven diagonal as compared to block-five diagonal. All methods given in Section 11.2 are extended to handle multiphase flow problems in exactly the same fashion as methods of Chapter 8 were extended in Chapter 10. For example, the details of eqn. (11.20) for two-phase, three-dimensional SS equations are given by Coats *et al*. (1967). The extension to three-phase flow is obtained by simply replacing the elements of $\boldsymbol{p}$ in eqn. (11.6) and $\boldsymbol{d}$ by sub-vectors of the form (see eqn. (10.6)):

$$\mathbf{p}_{ijk} = \begin{bmatrix} p_{1ijk} \\ p_{2ijk} \\ p_{3ijk} \end{bmatrix} \tag{11.37}$$

and replacing all elements of all of the matrices by 3 × 3 sub-matrices. This procedure is completely analogous to the procedure used in Chapters 5, 9 and 10.

The details of the SIP algorithm for multiphase three-dimensional problems are available in the literature (cf. Weinstein *et al*., 1969; Suarez and Farouq Ali, 1976).

11.3.2.2 *Comparison of Methods*

The comparisons presented in Section 11.2.5 are valid for multiphase problems solved by the IMPES method and for two-phase problems solved by the SEQ methods.

The application of SS method to two- or three-phase problems and the application of SEQ method to three-phase problems yield systems of equations with block-structured matrices. For these problems, the literature does not give systematic comparisons of methods which could guide the user. One must use whatever information is available for two-dimensional problems (Chapters 8 and 10) and single-phase three-dimensional problems (Section 11.2.5) in order to make a selection.

11.4 CONCLUDING REMARKS

Three-dimensional problems can be very expensive to run in terms of computer and human effort required. For this reason one must always attempt to simplify the problem so that a two-dimensional study can be conducted. Some methods to do this are discussed in Chapter 12. Another approach that reduces the effort required for three-dimensional problems was presented by Watts and Huang (1976). They proposed a method to solve three-dimensional problems, which is related to the Additive Correction Methods discussed in Section 8.3.5 (Chapter 8).

EXERCISES

Exercise 11.1

(a) *Derive the equivalent of eqn*. (11.1) *in 3-D cylindrical co-ordinates.*

(b) *Derive the form of finite-difference transmissibilities.*

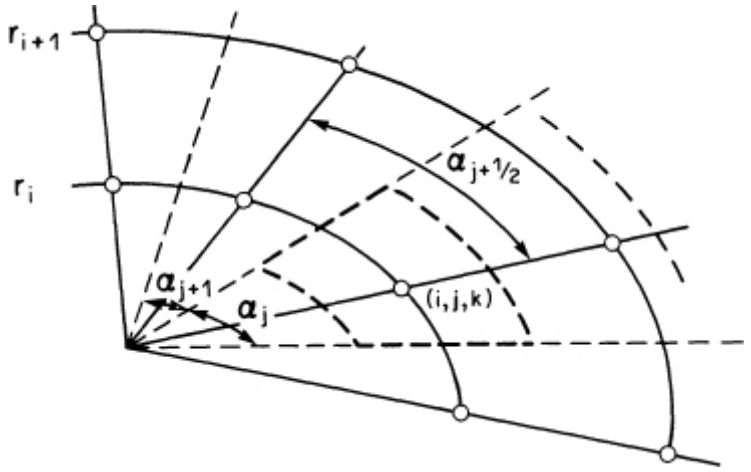
Solution Outline

(a) Using the same approach as in Exercise 7.1, we obtain,

$$\frac{1}{r}\left[r\lambda R\left(\frac{\partial p}{\partial r}-\gamma\frac{\partial h}{\partial r}\right)\right]+\frac{1}{r^2}\frac{\partial}{\partial\theta}\left[\lambda\mathrm{T}\left(\frac{\partial p}{\partial\theta}-\gamma\frac{\partial h}{\partial\theta}\right)\right]+\frac{\partial}{\partial z}\left[\lambda Z\left(\frac{\partial p}{\partial z}-\gamma\frac{\partial h}{\partial z}\right)\right]=\beta\frac{\partial p}{\partial t}+q \tag{A}$$

where λR, $\lambda\mathrm{T}$ and λZ are the transmissibilities in the r, θ and z direction. Equation (A) assumes that these are the principal directions of tensor λ.

(b) Denote the angle of segment j in radians by α_j, below. Then by generalising the development in Chapter 3, Section 3.6, we get



$$\mathrm{T}R_{(i+1/2),j,k}=\alpha_j r^L_{i+1/2}\frac{\Delta z_k}{(r_{i+1}-r_i)}\lambda R_{(i+1/2),j,k} \tag{B}$$

$$\mathrm{T}T_{i,(j+1/2),k}=\frac{(r_{i+1/2}-r_{i-1/2})\,\Delta z_k}{\alpha_{j+1/2}r_i}\lambda\mathrm{T}_{i,(j+1/2),k} \tag{C}$$

$$\mathrm{T}Z_{i,j,k+1/2}=\frac{\alpha_j(r^2_{i+1,2}-r^2_{i-1/2})}{2\,\Delta z_{k+1/2}}\lambda Z_{i,j,k+1/2} \tag{D}$$

where $r_{i+1/2}$ and $r^L_{i+1/2}$ are defined by eqns. (3.156) and (3.157) and

$$\alpha_j=\tfrac{1}{2}(\alpha_{j+1/2}+\alpha_{j-1/2}) \tag{E}$$

$$\Delta z_k=\tfrac{1}{2}(\Delta z_{k+1/2}+\Delta z_{k-1/2}) \tag{F}$$

The volume of the block is

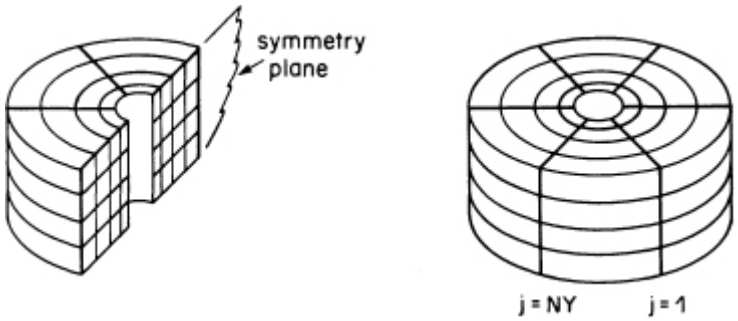
$$V_{ijk} = \frac{\alpha_j}{2} \Delta z_k (r^2_{i+1/2} - r^2_{i-1/2}) \tag{G}$$

Exercise 11.2

Investigate the use of direct methods for equations in cylindrical 3-*D coordinates. Consider:* (*a*) *standard ordering and* (*b*) D4 *ordering.*

Solution Outline

Denote by *NR, NY, NZ* the number of points in *r,* θ and *z* directions. When the geometry of the problem is not closed in the θ direction (see below, left), there is no difference between the cylindrical and Cartesian coordinate case. Consider therefore the case as shown in the figure (right).



(a) The coupling between $j = 1$ and $j = NY$ will not increase the bandwidth only if the *j* direction is ordered as *second*. Therefore the points should be numbered in the order

$$i, j, k \qquad \text{if} \quad NR \leq NZ$$

$$k, j, i \qquad \text{if} \quad NZ \leq NR$$

(b) For D4 ordering it is necessary that *NY* be *even*, otherwise submatrices $\mathbf{A_1}$, and $\mathbf{A_4}$ of eqn. (8.21) will no longer be diagonal. Find now the optimum rules for ordering within the diagonal planes.

CHAPTER 12

SPECIAL TOPICS

12.1 INTRODUCTION

This chapter discusses methods designed for handling various special problems, which have not been discussed elsewhere, but often arise in practical simulations. These include the use of pseudo-functions, simulation of reservoirs that do not behave as black-oil systems by black-oil models, and other special techniques. A common feature of many of these techniques is the attempt to approximately handle complex problems by reducing them to simpler problems through suitable assumptions. For example, the 'vertical equilibrium' (VE) technique reduces a 3-D problem to 2-D under appropriate conditions.

Techniques of this type are employed where:

(a) assumptions of the approximate method are valid and its use results in savings of computer cost,
(b) cost of simulation using a more rigorous model is prohibitive and the simplified model can still yield some useful information, or
(c) more rigorous models are not available.

12.2 PSEUDO-FUNCTIONS

Simulation of three-dimensional flow in reservoirs, or resolution in sufficient detail of the flow near individual wells in two- or three-dimensional models, may require a large number of blocks. Such simulations can be very costly and are often beyond the capacity of the available computer. The concept of pseudo-functions is a practical technique that allows one to approximately account for: (a) flow in the vertical or dip-normal direction in two-dimensional areal models, or (b) radial flow near wells in two- or three-dimensional models of reservoirs.

The simplification suggested is accomplished by replacing rock relative permeability and capillary pressure relationships with some pseudorelationships that account in an approximate fashion for the detail omitted in the reservoir model. In order

to derive pseudo-relative permeability and pseudo-capillary pressure relationships, a simpler analytical or numerical model of the system is used. In this section the concept will be illustrated by presenting the Coats *et al*. (1971a) use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoirs. This will be followed by a brief discussion of its extensions and of other types of pseudofunctions.

12.2.1 Coats *et al*. (1971a) Vertical Equilibrium Model

In a reservoir of large areal extent relative to thickness it is often reasonable to assume that equilibrium is established in the vertical direction instantaneously (viscous forces are negligible) and flow calculations need to be performed only in the two areal dimensions. This is known as the vertical equilibrium (VE) assumption. The simplest model of VE is obtained if the capillary forces are also neglected. This assumption is particularly suited to gas–water reservoirs and it results in complete gravity segregation. It does not, however, imply that the fluid–fluid interface would be horizontal throughout the reservoir, since no equilibrium assumption is made in the horizontal direction.

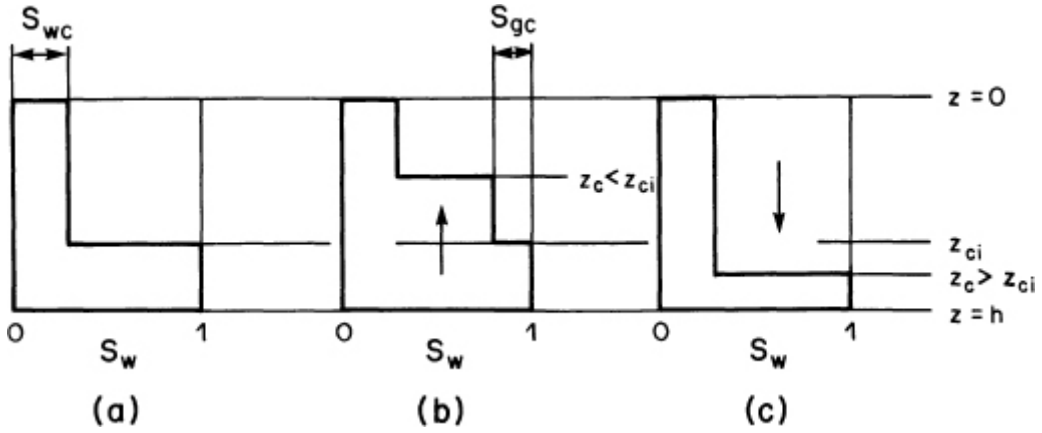


FIG. 12.1. Vertical fluid distribution due to gravity equilibrium.

Let us consider a vertical slice through a single block of a horizontal homogeneous two-phase gas–water reservoir. If the capillary pressure transition zone is small (say less than 10 % of reservoir thickness), we can assume that in gravity equilibrium the fluids will be completely segregated as shown in Fig. 12.1.

To simulate this distribution in a 2-D areal model, average properties $\bar{S}_W$, $\bar{P}_c$ and $\bar{k}_{rl}$ are defined such that correct flows and mass of fluids are obtained. Therefore, the depth-averaged saturation $\bar{S}_w$ is defined by

$$\bar{\phi}\bar{S}_w h = \int_0^h \phi(z) S_w(z)\,dz \tag{12.1}$$

where $\bar{\phi}$ is the average porosity used in the areal model. For constant ϕ, the initial saturation $\bar{S}_{wi}$ is (see Fig. 12.1a)

$$\bar{S}_{w_i} = \frac{S_{wc} z_{ci} + h - z_{ci}}{h} \tag{12.2}$$

If the gas–water contact (GWC) rises to z_c ($z_c < z_{ci}$) the average saturation is (see Fig. 12.1b)

$$\bar{S}_w = \frac{S_{wc} z_c + (1 - S_{gc})}{} \tag{12.3}$$

If instead the gas–water contact drops below its initial location ($z_c > z_{ci}$) the depth-averaged saturation is given by (see Fig. 12.1c)

$$\bar{S}_w = \frac{(S_{wc} z_c + h - z_c)}{h} \tag{12.4}$$

If $z = 0$ is selected as the reference plane for the two-dimensional areal calculations, then pressures at this plane are used to calculate flow between blocks in the horizontal plane. Since we have already assumed zero capillary pressure, water and gas pressures must be equal at the gas–water contact (z_c):

$$p_{wc} = p_{gc} \tag{12.5}$$

At any position z the pressures in the two phases are:

$$p_w(z) = p_{wc} + \gamma_w (z - z_c) \tag{12.6a}$$

$$p_g(z) = p_{gc} + \gamma_g (z - z_c) \tag{12.6b}$$

The difference in the two pressures at the reference plane is given by

$$p_g(0) - p_w(0) \equiv \bar{P}_c = \Delta\gamma z_c \tag{12.7}$$

where $\Delta\gamma = \gamma_w - \gamma_g$. Coats *et al*. (1971a) refer to the pressure difference in the above equation as 'pseudo-capillary pressure', even though it has nothing whatsoever to do

with capillary forces. Substituting for z_c from eqn. (12.7) into eqn. (12.3) and (12.4) yields:

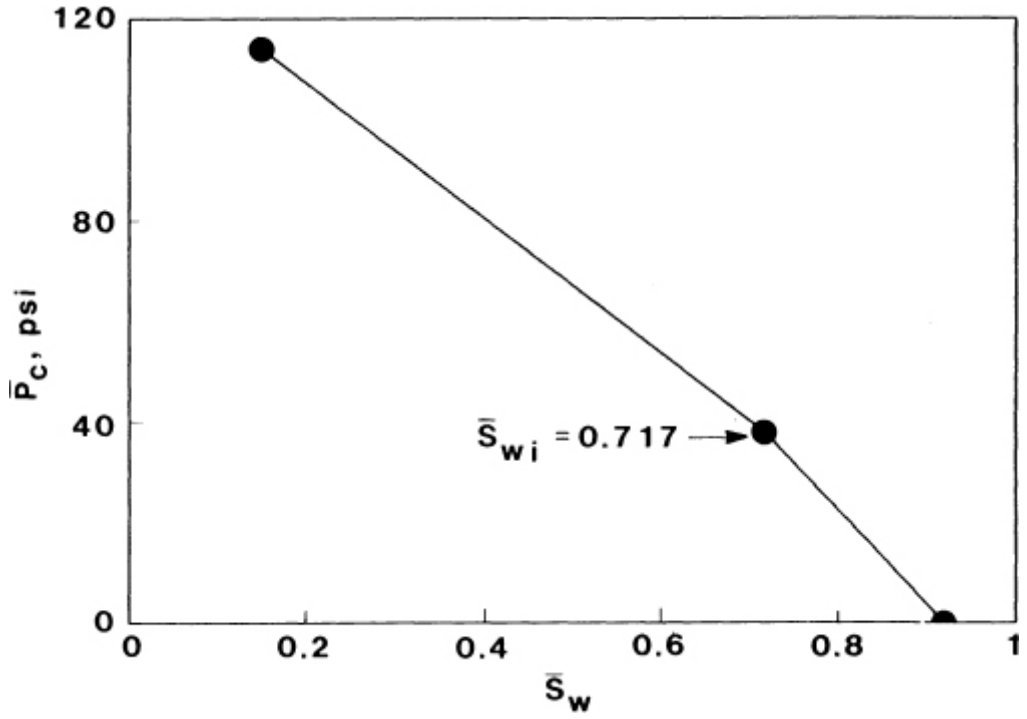


FIG. 12.2. An example of pseudo-capillary pressure (from Coats *et al.*, 1971).

$$\bar{S}_w = -\frac{(1 - S_{wc} - S_{gc})\bar{P}_c}{h\,\Delta\gamma} + \frac{h - S_{gc} z_{ci}}{h} \qquad z_c < z_{ci} \tag{12.8}$$

and

$$\bar{S}_w = 1 - \frac{(1 - S_{wc})\bar{P}_c}{h\,\Delta\gamma} \qquad z_c > z_{ci} \tag{12.9}$$

An example of the pseudo-capillary pressure calculated by the above relations for h = 300 ft, S_{wc} = 0•15, S_{gc} = 0•25, z_{ci} = 100 ft, $\Delta\gamma$ = 0*38 psi/ft is shown in Fig. 12.2.

Pseudo-relative permeabilities are derived from the requirement that the horizontal flow between the blocks calculated using the pseudo-relative permeability be equal to the actual flow integrated over the thickness h, i.e.,

$$h\bar{k}\bar{k}_{rl}(\bar{S}_w) = \int_0^h k(z) k_{rl}(S_w)\,\mathrm{d}z \tag{12.10}$$

where $\bar{k}$ and $\bar{k}_{rl}$ are the horizontal permeability and relative permeability for use in 2-D VE calculations. For constant k, this gives

$$\bar{k}_{rw} = \frac{k_{rwrg}(z_{ci} - z_c) + (h - z_{ci})}{h} \qquad z_c < z_{ci} \tag{12.11}$$

$$\bar{k}_{rw} = \frac{h - z_c}{h} \qquad z_c > z_{ci} \tag{12.12}$$

where k_{rwrg} is k_{rw} at the residual gas saturation S_{gc}.

Using eqns. (12.3) and (12.4) z_c can be eliminated from the above relations to obtain

$$\bar{k}_{rw} = \frac{k_{rwrg}\bar{S}_w}{(1 - S_{gc} - S_{wc})} + \frac{k_{rwrg}[z_{ci}(1 - S_{gc} - S_{wc}) - h + S_{gc}z_{ci}]}{h(1 - S_{gc} - S_{wc})} + \frac{(h - z_{ci})(1 - S_{gc} - S_{wc})}{h(1 - S_{gc} - S_{wc})} \qquad z_c < z_{ci} \tag{12.13}$$

and

$$\bar{k}_{rw} = \frac{\bar{S}_w - S_{wc}}{1 - S_{wc}} \qquad z_c > z_{ci} \tag{12.14}$$

Similarly

$$\bar{k}_{rg} = \frac{k_{rgcw}}{1 - S_{gc} - S_{wc}}\left[1 - \frac{S_{gc}z_{ci}}{h} - \bar{S}_w\right] \qquad z < z_{ci} \tag{12.15}$$

and

$$\bar{k}_{rg} = \frac{k_{rgcw}}{1 - S_{wc}}(1 - \bar{S}_w) \qquad z > z_{ci} \tag{12.16}$$

where k_{rgcw} is k_{rg} at the critical water saturation S_{wc}.

Figure 12.3 shows an example of pseudo-relative permeability curves along with rock curves for the case h = 300 ft, z_{ci} = 100 ft, S_{gc} = 0•25, S_{wc} = 0•15, k_{rwrg} = 0•4 and k_{rgcw} = 0•9. Note that *the pseudo-functions depend only on the end points of the rock curves* (*not on their shape*) *and the initial position of the contact* z_{ci}. Therefore they may be different for each grid block in reservoirs of large vertical relief.

Coats *et al*. (1971a) show how this approach could be used to accommodate reservoir structure and stratification. The practical applicability of this approach depends upon the rapidity with which perturbations from equilibrium in the vertical direction dissipate as compared with the rate of movement in the horizontal directions. Coats *et al*. (1971a) have developed a dimensionless group, the value of which is proportional to the degree of validity of the VE assumption. No critical value of this criterion is available at this time.

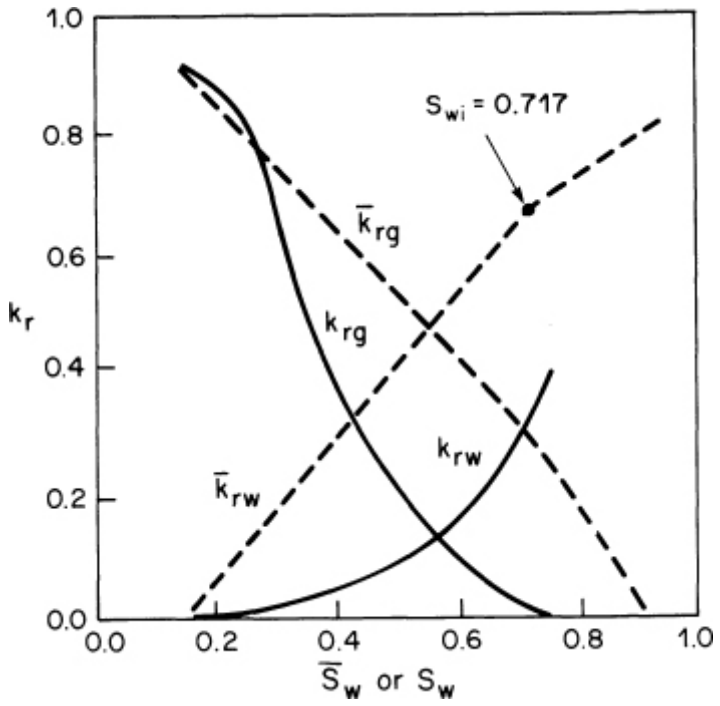


FIG. 12.3. An example of pseudo-relative permeability and corresponding rock curves (from Coats *et al*., 1971).

12.2.2 Other Pseudo-functions

(*a*) *Generalisation of VE concept*. Equations (12.1) and (12.10) can be used to develop the VE functions when the vertical fluid distribution is given by the equilibrium of capillary and gravity forces. Coats *et al*. (1967) presented such a method for the case when the capillary transition zone is not negligible. They also presented a criterion for determining when their approach provides a good representation of fluid displacement in reservoirs. This approach is particularly valuable for the simulation of water–oil flow in reservoirs with good vertical communication.

Martin (1968) has provided a sound theoretical basis for the VE pseudofunctions by showing how three-dimensional flow equations can be reduced to two-dimensional equations. This is accomplished through partial integration of the equations. Martin's

relationships include capillary-gravity equilibrium in the vertical direction and, for the two-phase case with a negligible capillary transition zone, they reduce to the equations presented in the previous section.

The generalised VE functions in homogeneous formations are functions of only the initial fluid contacts and the rock P_c and k_r curves. In a heterogeneous media, these functions also depend on the stratification ($\phi(z)$, $k(z)$) and may be different for each areal block.

(*b*) *Pseudo-functions for displacement in stratified reservoirs controlled by viscous forces*. Hearn (1971) has developed a procedure for calculating pseudo-relative permeability curves for use in the simulation of waterflooding in stratified reservoirs. In this approach viscous rather than gravity and capillary forces determine fluid distribution in the vertical direction. The model is based on the work of Hiatt (1958) and Warren and Cosgrove (1964), and it is valid only when the vertical viscous pressure gradients are negligible compared to the horizontal gradients. Hearn's model assumes incompressible flow with piston-like displacement taking place independently in each layer. Hawthorne (1974) has shown how Hearn's model can be modified to include the influence of capillary pressure.

(*c*) *Dynamic pseudo-functions*. The models discussed so far are not applicable for reservoirs which do not reach equilibrium conditions because of poor vertical communication, large vertical flows, changes in flow rates, etc. Although even in the general case, in principle, the reservoir can be described in 2-D, the exact determination of appropriate pseudo-functions would require the complete 3-D solution. This is not a desirable situation since solving the problem in 3-D is what we are trying to avoid through the use of pseudo-functions. In such a situation the pseudo-functions for a given block would vary with time and they would be different for each set of simulated conditions. As an approximation to this, Jacks *et al*. (1973) have developed a procedure for calculating these 'dynamic' pseudo-functions from 2-D cross-sectional simulations. These functions are applicable over a wide range of flow conditions and initial fluid saturations and may be space and time dependent. The dynamic pseudo-functions are derived from detailed vertical cross-section simulation studies of the reservoir under conditions to be expected during the areal model simulations.

The results of cross-section studies are processed to produce depth-averaged fluid saturations and pseudo-relative permeability values. Kyte and Berry (1975) have improved the Jacks *et al*. (1973) procedure to allow the calculation of dynamic pseudo-capillary pressure and to compensate for differences between grid sizes used in the areal and cross-sectional models.

(*d*) *Pseudo-functions for individual wells*. As discussed in Chapter 9, Section 9.7.3, realistic prediction of well performance in an areal model requires modification of k_{rl} functions, which may also be called pseudofunctions. In addition to the methods discussed in Chapter 9, Emmanuel and Cook (1974) showed how a cross-sectional model can be used to generate pseudo-curves. Woods and Khurana (1977) have extended this procedure by the partial integration of the flow equations to include water coning in three-dimensional reservoir models. Their approach does not assume vertical equilibrium; it accounts for viscous forces and geometry of flow which dominates well performance. The pseudo-functions are derived from the results of simulations with a two-dimensional coning model. The development is restricted to two-phase flow.

12.3 STREAM TUBE AND RELATED MODELS

Stream tube models predate the advent of simulation technology. They can be used for hand calculation of displacements in 2-D patterns. The reservoir is divided into 'stream tubes' based on a known or estimated pressure field and it is assumed that *there is no cross-flow between the tubes*. Then the displacement along each tube can be calculated separately and thus the 2-D problem is reduced to a number of 1-D problems. The calculation is rigorous only if the stream tube boundaries are identical with streamlines and the streamlines do not change with time.

Models of this type have been proposed by Muskat (1937), Fay and Prats (1951), Hurst (1953), Higgins and Leighton (1962, 1962a), Hauber (1964), Patton *et al*. (1971) and Le Blanc (1971).

Early models were designed for hand calculations but the model of Higgins and Leighton and later models are computer oriented. The most general model is that of Le Blanc.

When the pressure distribution is described by the steady-state (Laplace type) equation, the stream tube approach is rigorous. This is true of singlephase incompressible flow. Through the use of potential or pseudo-pressure as discussed in Chapter 2, stream tube models can also handle gravitational effects and variations of density and viscosity with pressure. The procedure can also be used rigorously for miscible displacement with unit mobility ratio. In more complex situations, stream tubes are still determined from the potential solution, and because they no longer correspond to streamlines, the model only approximately solves the problem.

The distribution of streamlines can be determined by analytical or numerical methods. Analytical solutions are available for a variety of well patterns (Muskat, 1937, Morel-Seytoux, 1966). Numerical techniques for solving elliptic equations are discussed in numerous texts.

Once the stream tubes have been defined, the saturation or concentration distribution along streamlines can be calculated again by analytical or numerical methods.

Application of stream tube models is restricted to two-dimensional patterns and to problems that approximately satisfy the simplifying assumptions mentioned above. Patton *et al*. (1971) present an interesting application of this technique to the prediction of polymer flood performance. A detailed treatment of the topic is beyond the scope of this book.

12.4 SIMULATION OF VARIABLE BUBBLE-POINT PROBLEMS

The bubble-point pressure p_b of an undersaturated oil is the pressure at which the first bubble of gas evolves as the pressure is decreased (Fig. 12.4). This is also the pressure at which all of the gas in a gas-liquid system is dissolved (or goes into solution) as the pressure is increased. Clearly, for $p > p_b$ the solution gas–oil ratio R_s must be constant. However, if sufficient gas is continuously supplied as the pressure is increased, the oil phase will remain saturated along the *swelling curves* shown by dashed lines in Fig. 12.4.

In the three-phase flow equations developed in Chapter 5, it was assumed that unique R_s and b_o curves can be used everywhere in the reservoir. This is true for example if the pressure decreases everywhere during the life of the reservoir. There are, however, several instances when this assumption is not satisfied.

Consider an originally undersaturated reservoir with initial conditions given by point A in Fig. 12.5. If the pressure is reduced below the original bubble-point pressure p_{b1}, free gas will evolve. Suppose the reservoir pressure is then increased by water injection starting at point B. Because of vertical gas migration, a block at the bottom of the reservoir will have a lower gas saturation and therefore the bubble point of the mixture in this block p_{b2} will be lower than the original bubble point p_{b1}, and re-pressurisation may lead to conditions indicated by point C. In contrast, a block towards the top of the reservoir with a larger gas saturation may follow the swelling curve above the original bubble point to point D with $P_{b3} > P_{b1}$.

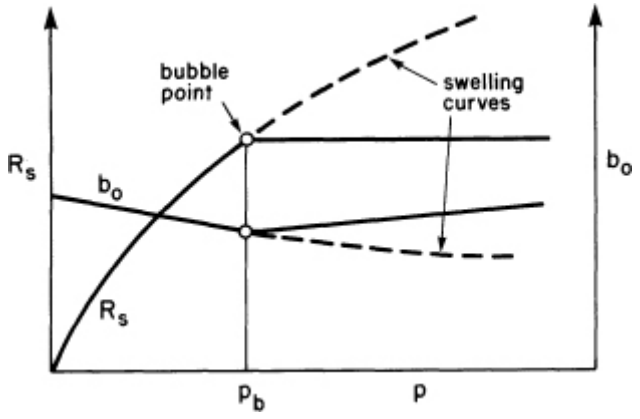


FIG. 12.4. Bubble-point pressure and swelling curves.

Gas injection into undersaturated reservoirs will also result in different bubble points in different parts of the reservoir. Finally, even in virgin reservoirs the bubble point increases with depth and this variation cannot be ignored for reservoirs of large thickness.

Simulation of variable bubble point requires special handling of the gas equation. Steffensen and Sheffield (1973), Kazemi (1975) and Thomas *et al*. (1976) presented methods of handling the problem by treating the bubble point as a variable in the accumulation terms only. Stright *et al*. (1977) presented a fully implicit treatment which is described below.

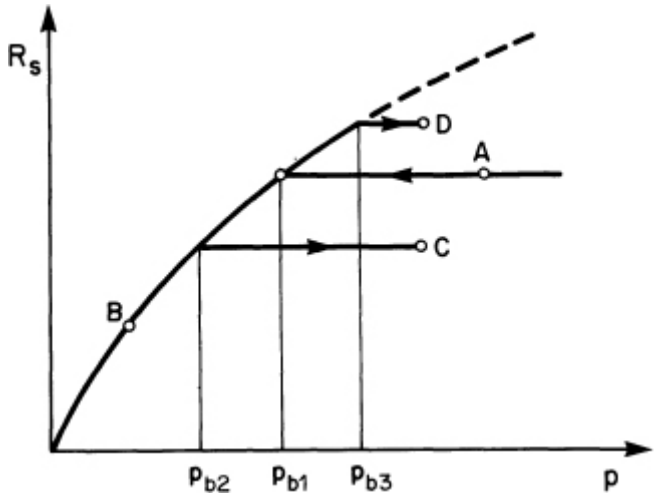


FIG. 12.5. Changing bubble point with reversal of pressure changes.

It is convenient to use *saturation pressure,* p_s, as a variable instead of bubble point. Saturation pressure of a computational cell is defined as the block pressure if there is free gas in the block, and it is the bubble point of the oil phase if the cell is undersaturated. Then the following holds:

(a) $p \geq p_s$
(b) if $p > p_s$, then $S_g = 0$
(c) if $S_g > 0$, then $p = p_s$

These relationships show that p_s and S_g are mutually exclusive variables; that is, they cannot both change independently. Oil properties are defined by p and p_s through the swelling curves and the oil compressibility above the bubble point:

$$R_s = f(p_s) \qquad b_o = f(p, p_s)$$

Consider now the conservative finite-difference expansion of the mass transfer term $\partial/\partial t(S_o b_o R_s)$ in the gas equation

$$\Delta_t(S_o b_o R_s) = (b_o R_s)^{n+1}(S_o^{n+1} - S_o^n) + S_o^n[(b_o R_s)^{n+1} - (b_o R_s)^n] \tag{12.17}$$

$$(b_o R_s)^k = b_o(p^k, p_s^k) R_s(p_s^k) \tag{12.18}$$

For simplicity, rock compressibility is assumed to be zero. The second term in the expansion is a function of both pressure change and saturation pressure change:

$$\Delta_t(b_o R_s) = f(\Delta_t p, \Delta_t p_s) \tag{12.19}$$

Since the expansion of this term depends on the unknown changes in p and p_s, the solution technique must be iterative. Stright *et al.* (1977) ignore $\Delta_t p_s$ for the first iteration by expanding eqn. (12.17) as

$$\Delta_t(S_o b_o R_s) = (b_o R_s)^n \Delta_t S_o + S_o^n b_o' R_s^n \Delta_t p \tag{12.20}$$

where

$$b_o' = [b_o(p^n + \Delta p, p_s^n) - b_o(p^n, p_s^n)]/\Delta p \tag{12.21}$$

The results of the first iteration are denoted by $p^{(1)}, S_o^{1)}, S_g^{(1)}$. There are now several possibilities that can result in a change in saturation pressure:

(a) $p^{(1)} < p_s^n$, in which case $p_s^{n+1} = p^{n+1}$ and S_g is increased by gas evolution,

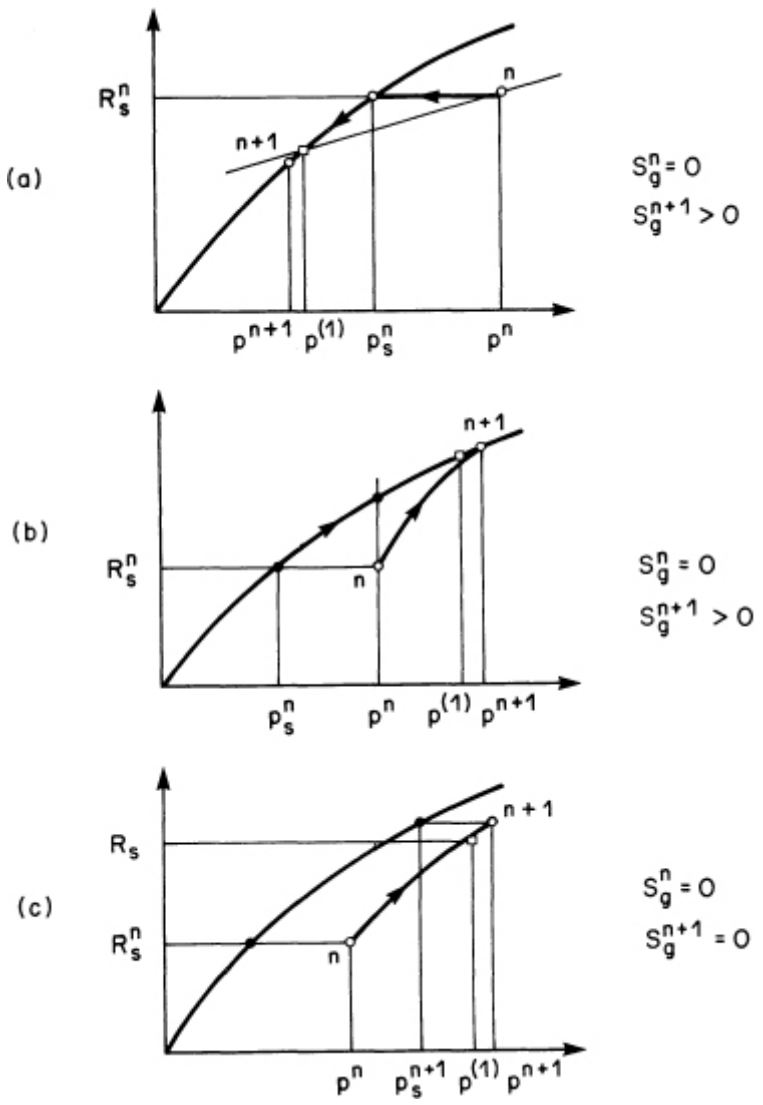


FIG. 12.6. Logic for modelling variable saturation pressure (from Stright *et al.*, 1977).

(b) $p^{(1)} \geq p_s^n$ and $S_g^{(1)} > 0$ in which case some (or all) available gas may be dissolved and p_s may increase up to p^{n+1}.

(c) $S_g^{(1)} < 0$, in which case p_s must be reduced to obtain $S_g^{n+1} = 0$.

The last situation can result from mixing when flow is in the direction of increasing R_s. The remaining possibilities represent cases when saturation pressure does not change.

Case (*a*) This is the standard treatment for reservoirs produced under declining pressure. Stright *et al.* (1977) give two equivalent expansions which approximate the chord slope between *n* and *n* + 1 (see Fig. 12.6a).

Case (*b*) The cell may or may not become saturated depending on $S_g^{(1)}$ and $p^{(1)}$. The pressure *p**, at which the free gas as predicted by $S_g^{(1)}$ will dissolve (i.e., the bubble-

point pressure), can be determined from the following calculation. At this pressure, $S_g^* = 0,\ S_o^* = 1 - S_{\mathrm{w}}^{(1)}$ and the conservation condition for the oil phase gives

$$b_{\mathrm{o}}^* S_{\mathrm{o}}^* = b_{\mathrm{o}}^{(1)} S_{\mathrm{o}}^{(1)} \tag{12.22}$$

Similarly, the total gas in the cell after the first iteration is:

$$b_{\mathrm{o}}^{(1)} R_{\mathrm{s}}^n S_{\mathrm{o}}^{(1)} + b_{\mathrm{g}}^{(1)} S_{\mathrm{g}}^{(1)} \tag{12.123}$$

and if all of the gas is dissolved, then it must be equal to $b_o^* R_s^* S_o^*$. Utilising eqn. (12.22) above, we get

$$R_{\mathrm{s}}^* = (b_{\mathrm{o}}^{(1)} R_{\mathrm{s}}^n S_{\mathrm{o}}^{(1)} + b_{\mathrm{g}}^{(1)} S_{\mathrm{g}}^{(1)}) / b_{\mathrm{o}}^{(1)} S_{\mathrm{o}}^{(1)} \tag{12.24}$$

which determines the predicted bubble point p^*. There are now two possibilities:

1. If $p^{(1)} < p^*$, then only part of the gas can be dissolved and $p_s^{n+1} = p^{n+1}$. The pressure $p^{(1)}$ then serves as a good approximation to p_s^{n+1}, and the saturation $S_g^{(1)}$ is recomputed to account for dissolved gas:

$$\begin{aligned} S_{\mathrm{g}}^{(1)} &\simeq S_{\mathrm{g}}^{(1)} - \delta S_{\mathrm{g}} \\ &= S_{\mathrm{g}}^{(1)} - [R_{\mathrm{s}}(p^{(1)}) b_{\mathrm{o}}(p^{(1)}, p^{(1)}) - R_{\mathrm{s}}(p^n) b_{\mathrm{o}}(p^n, p_{\mathrm{s}}^n)] S_{\mathrm{o}}^n / b_{\mathrm{g}}^{(1)} \end{aligned} \tag{12.25}$$

   Expansion of eqn. (12.17) is then written as:

$$\begin{aligned} \Delta_t(S_{\mathrm{o}} b_{\mathrm{o}} R_{\mathrm{s}}) = b_{\mathrm{o}}^{(1)} R_{\mathrm{s}}^{(1)} (S_{\mathrm{o}}^{n+1} - S_{\mathrm{o}}^n) + S_{\mathrm{o}}^n [\Delta(b_{\mathrm{o}} R_{\mathrm{s}})_{p_{\mathrm{s}}} \\ + (b_{\mathrm{o}} R_{\mathrm{s}})'_p (p^{n+1} - p^n)] \end{aligned} \tag{12.26}$$

   where according to Fig. 12.6b

$$\begin{aligned} \Delta(b_{\mathrm{o}} R_{\mathrm{s}})_{p_{\mathrm{s}}} &= [b_{\mathrm{o}}(p^n, p^n) R_{\mathrm{s}}(p^n) - b_{\mathrm{o}}(p^n, p_{\mathrm{s}}^n) R_{\mathrm{s}}(p_{\mathrm{s}}^n)] \\ (b_{\mathrm{o}} R_{\mathrm{s}})'_p &= [b_{\mathrm{o}}(p^{(1)}, p^{(1)}) R_{\mathrm{s}}(p^{(1)}) - b_{\mathrm{o}}(p^n, p^n) R_{\mathrm{s}}(p^n)] / (p^{(1)} - p^n) \end{aligned}$$

2. If $p^{(1)} > p^*$, all available gas will dissolve and saturation pressure will not reach block pressure, $p_s^{n+1} < p^{n+1}$. Then we set $S_g^{n+1} = 0$, $\delta S_{\mathrm{g}} = -S_{\mathrm{g}}^n$ and the saturation pressure change is determined by assuming $p_s^{n+1} \approx p^*$. Expansion (12.17) now takes the form:

$$\Delta_t(S_o b_o R_s) = b_o(p^{(1)}, p^*) R_s(p^*)(S_o^{n+1} - S_o^n) + S_o^n[\Delta(b_o R_s)_{p_s} + (b_o R_s)'_{p_s}(p_s^{n+1} - p^n) + (b_o R_s)'_p(p^{n+1} - p_s^{n+1})] \tag{12.27}$$

where according to Fig (12.6c)

$$\Delta(b_o R_s)_{p_s} = [b_o(p^n, p^n) R_s(p^n) - b_o(p^n, p_s^n) R_s(p_s^n)]$$

$$(b_o R_s)'_{p_s} = [b_o(p^*, p^*) R_s(p^*) - b_o(p^n, p^n) R_s(p^n)]/(p^* - p^n)$$

$$(b_o R_s)'_p = [b_o(p^{(1)}, p^*) - b_o(p^*, p^*)] R_s(p^*)/(p^{(1)} - p^*)$$

Case (*c*) This case can be handled by the algorithm of Case (b). Note that eqns. (12.22) to (12.24) will predict a decrease in bubble point when $S_g^{(1)}$ is negative.

For all cases, the equation for the second iteration can be written in a common form

$$\Delta_t(S_o b_o R_s) = (b_o R_s)^k \Delta_t S_o + S_o^n[(b_o R_s)'_p \Delta_t p + (b_o R_s)'_{p_s} \Delta_t p_s] + R \tag{12.28}$$

where R is an explicit term. The term R and the coefficients depend on the status of the cell predicted by the first iteration.

The method just described usually converges in two iterations. Treatment of the accumulation terms is sufficient for areal or 3-D models. For coning applications, it is necessary for stability to include implicit terms with respect to p_s in the flow terms involving R_s. These terms replace the implicit coefficients with respect to $\Delta_t S_g$ in the gas equation without changing the structure of the equations. The implicit production terms must also be modified in the same fashion.

In conclusion, it is worth noting that the same computational problem arises in the simulation of steam flooding and geothermal reservoirs. Here the two variables constrained by the phase relationship are steam saturation and block temperature.

12.5 SIMULATION OF SYSTEMS NOT DESCRIBED BY BLACK-OIL MODELS

Some processes taking place in the reservoir cannot be described by black-oil models because the basic assumptions upon which these models are based are not valid. However, they can often be described by more general equations for multicomponent systems including both convective and dispersive mass transfer. If we consider only oil

and gas phases, each consisting of N hydrocarbon components, the general equations are (Bird *et al.*, 1960):

$$-\nabla\cdot(\rho_o\omega_{oi}\boldsymbol{u}_o+\rho_g\omega_{gi}\boldsymbol{u}_g)+\nabla\cdot(\phi S_o\rho_o K_{oi}\nabla\omega_{oi}+\phi S_g\rho_g K_{gi}\nabla\omega_{gi}) \tag{12.29}$$
$$=\frac{\partial}{\partial t}(\phi S_o\rho_o\omega_{oi}+\phi S_g\rho_g\omega_{gi})+q_i \qquad i=1,\ldots,N$$

Here ω_{oi} and ω_{gi} are the mass fractions of component i in the oil and gas phases and K_{li} are the tensors of dispersion coefficients.

The supplementary equations are the familiar two-phase relations

$$\boldsymbol{u}_l=-\lambda_l(\nabla p_l-\gamma_l\nabla z) \qquad P_c=p_g-p_o \qquad S_o+S_g=1 \tag{12.30}$$

constraints on mass fractions

$$\sum_{i=1}^{N}\omega_{oi}=1 \qquad \sum_{i=1}^{N}\omega_{gi}=1 \tag{12.31}$$

and the equations describing mass transfer between the phases:

$$\frac{\omega_{gi}}{\omega_{oi}}\equiv \mathrm{K}_i=\mathrm{f}(p_l,\omega_i) \qquad i=1,\ldots,N \tag{12.32}$$

where ω_i is the fraction of component i in the two-phase mixture, and K_i are the phase equilibrium constants (Katz *et al.*, 1959), also called K-values or partition coefficients. Phase densities and viscosities depend in a complex way on the composition of each phase, and the phase compositions depend upon the overall composition and pressure (temperature is assumed to be constant throughout this discussion).

Equations (12.29) to (12.32) are a closed system of $2N + 6$ equations in $2N + 6$ unknowns ω_{oi}, ω_{gi}, $\boldsymbol{u}_o$, $\boldsymbol{u}_g$, p_o, p_g, S_o, S_g. It is easy to see how this formulation can be extended to three-phase flow.

Two simplifications of the general equations which lead to commonly used models are discussed next.

12.5.1 Simulation of Miscible Displacement

Miscible displacement is obtained when two or more components flow within a single phase. Examples are displacement of oil by a solvent, and the transport of salts,

polymers or contaminants in water. These processes are described by the *convection–diffusion equation* and have applications in many areas including miscible, polymer and chemical flooding of reservoirs and air and water pollution modelling.

Consider two components, oil and solvent, which are miscible in all proportions. If we assume that

(a) the fluid components and the rock are incompressible, and
(b) there is no volume change due to the mixing of components, then the general component balance can be simplified to (Peaceman and Rachford, 1962; Lantz, 1970; Settari *et al*., 1977):

$$-\nabla . \boldsymbol{u} = \nabla . \left[\frac{k}{\mu} (\nabla p - \gamma \nabla z) \right] = q \tag{12.33}$$

$$\nabla . (\phi K \nabla C) - \nabla . \boldsymbol{u} C = \phi \frac{\partial C}{\partial t} + C_q q \tag{12.34}$$

Here $\boldsymbol{u}$ is the single-phase Darcy velocity, $\boldsymbol{K}$ is the dispersion tensor which has contributions from molecular diffusion and from hydrodynamic dispersion, C is the *in situ* volumetric solvent concentration and C_q is the source/sink concentration (C_q is equal to input concentration for injection wells and $C_q = C$ for production wells). The mixture viscosity μ and density ρ are given by:

$$\gamma = \gamma_o (1 - C) + \gamma_s C \tag{12.35}$$

$$\mu = \mathrm{f}(\mu_o, \mu_s, C) \tag{12.36}$$

Equation (12.35) is a consequence of the assumption (a) above. The viscosity function (12.36) is usually expressed by a logarithmic or power-law mixing rule (Peaceman and Rachford, 1962; Settari *et al*., 1977). Similar equations can be derived for multicomponent systems. Their treatment is complicated by the interaction of dispersion coefficients (Whitaker, 1967; Bear, 1972; Sigmund, 1976).

12.5.1.1 *Solution of Convection–Diffusion Equations*

Direct solution of these equations presents two difficulties. The first, common to all equations of this type, is that the numerical solution of convective flow must be very accurate so that the numerical errors (called ‘numerical dispersion’) do not mask the effect of physical dispersion. The numerical methods tend to either give smooth solutions with large numerical dispersion or solutions which are accurate only for a

small grid spacing and oscillate for coarse grids. This problem has been studied extensively in the literature on numerical fluid dynamics (e.g., Van Leer, 1977; Forester, 1977). In the petroleum literature, these equations have been solved by finite difference as well as variational methods. A review of the subject is presented by Settari *et al*. (1977).

The second problem is more characteristic of reservoir applications. In many miscible processes, the mobility ratio $M = \mu_o/\mu_s$ is highly unfavourable (M = 10–100). This may cause grid orientation problems as discussed in Chapter 9, Section 9.7.3. With sufficiently high M, the flow will become physically unstable, resulting in viscous fingering of the displacing fluid through the oil. Such flows can be simulated, as demonstrated by Peaceman and Rachford (1962) and Settari *et al*. (1977). However, resolving the fine structure of the flow requires a fine grid, which may be both impractical and undesirable for large-scale field simulations.

12.5.1.2 *Simulation of Miscible Displacement Using Immiscible Simulators*

There is a strong motivation to develop methods which could simulate miscible displacement using conventional black-oil (β-type) simulators. Besides the widespread use of black-oil models and the difficulties associated with the direct solution of miscible equations, some fluid mixtures may be miscible or immiscible (two phase) depending on the concentration and pressure. Therefore the reservoir flow may change from immiscible to miscible and vice versa; hence it could not be described by eqns. (12.33) and (12.34).

The basis for the miscible–immiscible analogy was established by Lantz (1970). He showed that for a constant dispersion field, $\boldsymbol{K}$, miscible flow could be rigorously modelled by immiscible flow equations. If the immiscible phases are denoted by 'w' and 'n', then the analogous variables are:

$$\mu_w = \mu_o, \qquad \mu_n = \mu_s \qquad S_n = C$$

Lantz showed that bulk miscible flow can be represented by the bulk immiscible flow and the dispersion flux can be represented by the capillary flow.

For $\mu_s = \mu_o = \mu$ (i.e., $M = 1$), the analogy requires

$$k_{rw} = S_w \tag{12.37a}$$

$$k_{rn} = S_n \tag{12.37b}$$

and

$$P_c = \frac{\phi K \mu}{k} \ln\left(\frac{1 - S_w}{S_w}\right) \tag{12.38}$$

The relative permeabilities are proportional to saturation, because the two phases flow with the same velocity. When $M \neq 1$, k_{rl} and P_c become strong functions of viscosities which may, for large M, cause stability problems.

Todd and Longstaff (1972) have presented a more practical approach. They showed that an empirical model can give realistic predictions of miscible flood performance without having to resolve the fine structure of the flow. The relative permeabilities in their model are always defined by eqn. (12.37). However, the effective grid block viscosities and densities are defined by empirical mixing rules which reflect the effective fluid properties averaged over the block. The mixing depends on the rate of dispersion. If the rate of dispersion is high, fluids within a block can be considered completely mixed and their properties determined by mixing rules (12.35) and (12.36). On the other hand, if the dispersion is small, the mixed zone will be negligible compared to the block size and the effective properties of the oil and solvent should be close to those of pure components. Todd and Longstaff (1972) have suggested the following model for effective viscosities μ_{oe}, μ_{se} in the general case:

$$\mu_w = \mu_{oe} = \mu_o^{1-\omega} \mu^{\omega} \tag{12.39a}$$

$$\mu_n = \mu_{se} = \mu_s^{1-\omega} \mu^{\omega} \tag{12.39b}$$

where ω is a mixing parameter and μ is the fully mixed viscosity. Note that $\mu_{le} = \mu_l$ for $\omega = 0$ and $\mu_{le} = \mu$ for $\omega = 1$.

Effective densities are obtained as follows. Let $\mu = f(\mu_o, \mu_s, C = S_w)$ be the viscosity rule for complete mixing, and define effective saturations, S_{le}, through this rule so that the following relations are satisfied with μ_l determined by (12.39):

$$\mu_w = f(\mu_o, \mu_s, S_{we})$$
$$\mu_n = f(\mu_o, \mu_s, 1 - S_{we})$$

In other words, S_{le} are the saturations for which the complete mixing rule yields the partially-mixed viscosities.

The effective densities are now obtained from:

$$\gamma_w = \gamma_{oe} = \gamma_o S_{we} + \gamma_s (1 - S_{we}) \tag{12.40a}$$

$$\gamma_n = \gamma_{se} = \gamma_o(1 - S_{ne}) + \gamma_s S_{ne} \tag{12.40b}$$

This model fails for a power law mixing rule when $M = 1$. For this case, a rule analogous to (12.39) is proposed:

$$\gamma_w = (1 - \omega)\gamma_o + \omega\gamma \tag{12.41a}$$

$$\gamma_n = (1 - \omega)\gamma_s + \omega\gamma \tag{12.41b}$$

The major problem in the application of this scheme lies in the selection of an appropriate mixing parameter ω. Todd and Longstaff (1972) used $\omega = 2/3$ to match laboratory experiments, but they recommended $\omega = 1/3$ for field applications. Warner (1977) gives some analysis which suggests the use of $\omega = 0{\bullet}8$.

12.5.2 Simulation of Compositional Effects

An important special case of the general equations is obtained by ignoring the dispersive transport in a multicomponent system. This is called a *compositional model*. Usually, water is treated as a separate component present only in the water phase and no mass transfer is assumed between water and oil or gas phases (Van-Quy *et al*., 1972; Nolen, 1973).

A compositional model will reduce to a black-oil model if it is assumed that the hydrocarbons consist of only two components, oil and gas, with only gas being able to partition between the liquid and gas phases. The compositional properties can then be derived from black-oil properties and vice-versa (Exercise 12.1).

12.5.2.1 *Compositional Models*

The major difficulty in simulating compositional flow is an efficient prediction of the PVT behaviour of complex mixtures. To date the best predictive methods are based on modifications and extensions of the Redlich–Kwong equation of state (Wilson, 1969; Soave, 1972; Peng and Robinson, 1976) and they are computationally slow because the solution is usually obtained iteratively. Attempts to improve the efficiency of PVT treatment in simulators are discussed by Fussell and Yanosik (1976), Kazemi *et al*. (1977) and Fussell and Fussell (1977). Most compositional simulators are based on the extension of IMPES method. This is achieved by treating the component mass fractions explicitly in the flow terms, and it results in stability limitation. For most problems in Cartesian co-ordinates, this limitation is less restrictive than the limitation resulting from the explicit treatment of phase transmissibilities. However, when it is necessary to treat transmissibilities implicitly, the stability restriction with respect to component

mass fractions becomes a problem. Experience with fully implicit compositional models has not been reported in the literature to date.

12.5.2.2 *Simulation of Compositional Effects Using Black-Oil Models*

Several methods for approximate modelling of compositional effects in a black-oil simulator have been presented. The main advantage of this approach is that, when applicable, it takes much less computing time than a fully compositional treatment.

Spivak and Dixon (1973) presented a method for treating gas-condensate reservoirs by assuming that condensate and gas can be represented by two pseudo-components. The dry gas component was assumed to exist only in the gas phase and the 'condensate', which is liquid at standard conditions was allowed to exist in both the gas and the liquid phases at reservoir conditions. Mass transfer between phases was determined by the liquid content of the gas phase, defined as

$$r_s = \left(\frac{V_{\text{condensate}}}{V_{\text{dry gas}}} \right)_{\text{STC}} = \text{f}(p) \tag{12.42}$$

Thus, r_s is analogous to R_s and the formulation is identical to the formulation of a black-oil model if the rôles of gas and liquid hydrocarbon phases are interchanged.

Cook *et al*. (1974) present a method of accounting for compositional effects during gas injection in volatile oil or gas condensate reservoirs. In their model, the properties b_o, b_g R_s (or r_s), μ_o, μ_g, ρ_o and ρ_g are correlated with a parameter G_i, representing 'cumulative injection gas invading a cell'. They obtained the required correlations from 1-D compositional simulations of gas injection. Because the correlations were found to be independent of grid position, Cook *et al*. (1974) then applied these functions in multidimensional models.

Henry (1976) shows that assuming fluid properties to be functions of G_i only as proposed by Cook *et al*. (1974) is not valid when the K-values are functions of composition (Cook *et al*. model assumed K = f(p) only). Consequently, use of modified beta-type models is not recommended when large changes in composition are anticipated.

12.6 HISTORY-DEPENDENT SATURATION FUNCTIONS

Hysteresis of capillary pressure and relative permeabilities is particularly important for correct simulation in situations involving flow reversals, such as gas cycling and steam stimulation (cf. Cutler and Rees, 1970; Land, 1968; Evrenos and Comer, 1969; Dandona

and Morse, 1973). A physical description of the phenomenon followed by a numerical model is presented here.

12.6.1 Physical Model of Hysteresis

Construction of physical models from experimental hysteresis data is discussed by Evrenos and Comer (1969a) and Killough (1976). (Our exposition is based on the work of Killough with additional detail.)

Consider first capillary hysteresis in a w–n system. As shown in Fig. 12.7, three primary curves must be considered: (a) is the primary drainage curve, (b) is called the pendular imbibition curve, and (c) is the secondary drainage curve. These curves are obtained when displacement in either direction is carried out completely to the residual value of saturation. The curves are therefore called *bounding curves.*

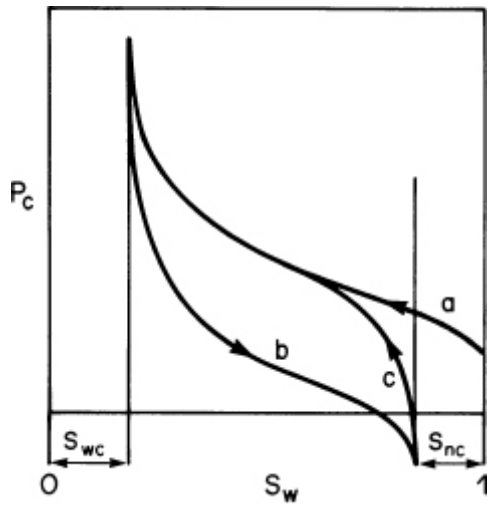


FIG. 12.7. Capillary hysteresis.

If the drainage process is reversed before S_w reaches S_{wc}, P_c will follow a curve of the type (d) shown in Fig. 12.8 (left). Similarly, if the primary imbibition process is reversed, the new P_c will follow a curve of the type (e) shown in Fig. 12.8 (left). Experimental observations indicate that the end points of these *scanning curves* are the residual saturations.

Another situation occurs if a scanning process, originating on the primary drainage curve at point A (Fig. 12.8, right), is again reversed. In such a situation a new scanning curve is generated which reaches the bounding curve at the starting point A. Similarly, if scanning starts on the imbibition curve, the drainage scanning curves will scan towards S_{wc} and imbibition curves towards point B (Fig. 12.8, left).

The behaviour described above is only valid when the system has previously undergone a complete drainage displacement and therefore all drainage cycles follow curve (c) of Fig. 12.7. However, if the *primary* drainage process is reversed, the imbibition scanning curve will scan towards a residual saturation $S_{nc}^{*} < S_{nc}$ (Fig. 12.9). This is because the residual non-wetting saturation depends on the *maximum* non-wetting saturation S_n^{max} attained previously in a drainage cycle, and it increases with S_n^{max} . Such behaviour is consistent with a model of relative permeability hysteresis to be discussed next.

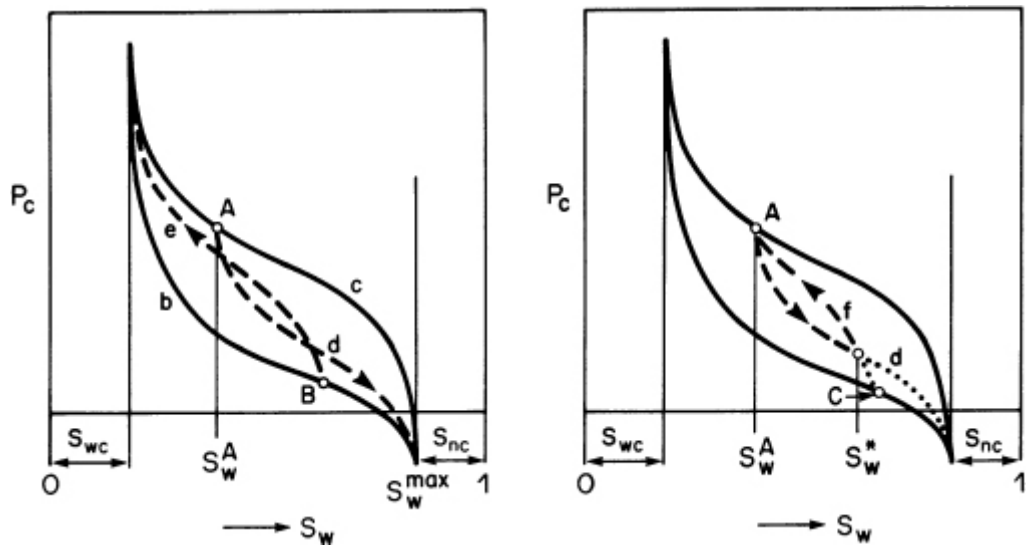


FIG. 12.8. Scanning P_c curves. Left, reversal starting on bounding curves; right, reversal starting on a scanning curve.

Typical k_{rn} hysteresis is shown in Fig. 12.10. The residual saturation S_{nc}^{*} in an imbibition cycle is caused by 'trapping off ' of the non-wetting phase by the advancing wetting phase. It has been determined experimentally that $S_{nc}^{*} = f(S_n^{max})$ and when another reversal from imbibition to drainage occurs, k_{rn} will follow the imbibition curve until S_n reaches the historical maximum non-wetting saturation, S_n^{max}. For non-wetting saturations greater than this maximum, k_{rn} follows the drainage curve. Thus, according to this model

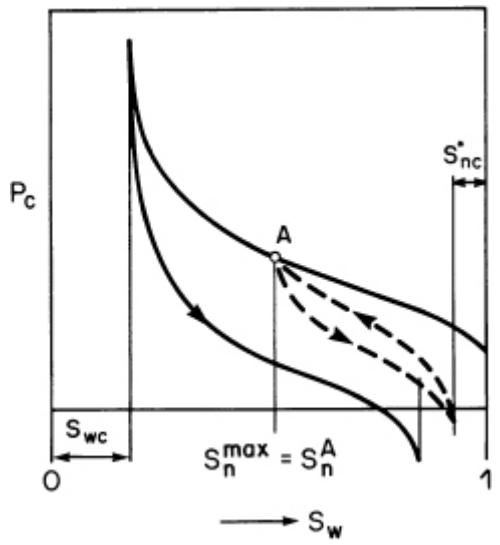


FIG. 12.9. Scanning P_c curve starting on the primary drainage curve.

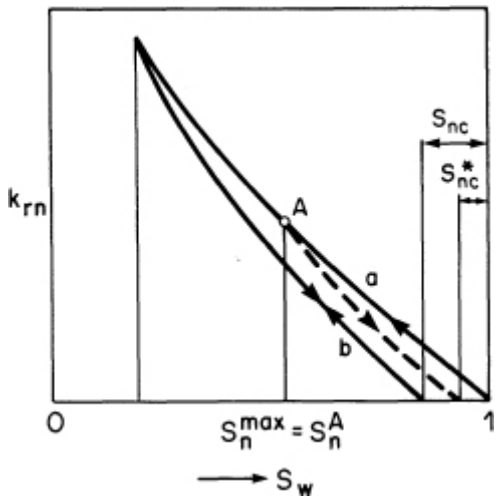


FIG. 12.10. Hysteresis model for k_{rn}.

$$S_{nc} = f(S_w^{max} = 1 - S_{wc})$$

A similar model can be formulated for k_{rw}, but its dependence on the residual non-wetting phase saturation is usually much smaller.

12.6.2 Numerical Treatment of Hysteresis

Numerical models attempt to qualitatively reproduce the behaviour described above from the knowledge of imbibition and drainage curves only. Killough (1976) proposes the following equations for the scanning P_c curves:

For curve (d) (Fig. 12.8):

$$P_c(S_w) = P_c^c(S_w) - F(S_w)[P_c^c(S_w) - P_c^b(S_w)] \tag{12.43}$$

where

$$F = \frac{\dfrac{1}{S_w - S_w^A + \varepsilon} - \dfrac{1}{\varepsilon}}{\dfrac{1}{S_w^{max} - S_w^A + \varepsilon} - \dfrac{1}{\varepsilon}} \tag{12.44}$$

$P_c^c\,(S_w)$ and $P_c^b\,(S_w)$ are the values of P_c at S_w from the bounding curves (c) and (b), S_w^A is the saturation at the point of reversal and ε is a parameter affecting the shape of the curve. Similarly, curve (e) is approximated by

$$P_c(S_w) = P_c^b(S_w) + F(S_w)[P_c^c(S_w) - P_c^b(S_w)] \tag{12.45}$$

Where

$$F = \frac{\dfrac{1}{S_w^B - S_w + \varepsilon} - \dfrac{1}{\varepsilon}}{\dfrac{1}{S_w^B - S_{wc} + \varepsilon} - \dfrac{1}{\varepsilon}} \tag{12.46}$$

The second reversal curve (curve (f) of Fig. 12.8, right) is determined by

$$P_c(S_w) = P_c^b(S_w) + F^*(S_w)[P_c^c(S_w) - P_c^b(S_w)] \tag{12.47}$$

where

$$F^* = \frac{\dfrac{1}{S_w^* - S_w^C + \varepsilon} - \dfrac{1}{\varepsilon}}{\dfrac{1}{S_w^A - S_w^C + \varepsilon} - \dfrac{1}{\varepsilon}} \tag{12.48}$$

The unknown saturation S_w^C at point C is determined from the condition that the two curves (d) and (f) meet at S_w^*. This gives $F^* = 1 - F$ which can be solved for S_w^C.

The case depicted on Fig. 12.9 is handled by the same equations, but the bounding curves P_c^{c} and P_c^{b} are transformed (normalised) to P_c^{c} and P_c^{b} so that they cover the interval $(S_{\mathrm{wc}}, 1 - S_{\mathrm{nc}}^{*})$. Note that the primary drainage curve is not used. The trapped saturation S_{nc}^{*} is calculated by the equation developed by Land (1968a):

$$S_{\mathrm{nc}}^{*} = \frac{S_{\mathrm{n}}^{\mathrm{A}}}{1 + \left(\frac{1}{S_{\mathrm{nc}}} - \frac{1}{(1 - S_{\mathrm{wc}})}\right) S_{\mathrm{n}}^{\mathrm{A}}} \tag{12.49}$$

The intermediate k_{r} curves are calculated by interpolation including a shape parameter λ:

$$k_{\mathrm{rn}}(S_{\mathrm{n}}) = k_{\mathrm{rn}}^{\mathrm{a}}(S_{\mathrm{n}}^{\mathrm{A}}) \left(\frac{S_{\mathrm{n}} - S_{\mathrm{nc}}^{*}}{S_{\mathrm{n}}^{\mathrm{A}} - S_{\mathrm{nc}}^{*}}\right)^{\lambda} \tag{12.50}$$

where k_{rn}^{a} is obtained from curve (a).

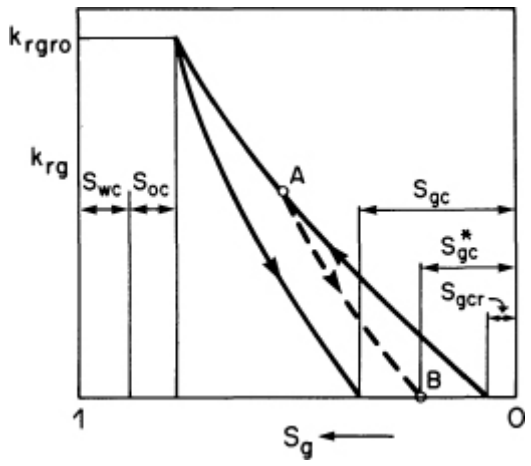


FIG. 12.11. Hysteresis of k_{rg} according to Coats (1976a).

A similar model of k_{rn} hysteresis was suggested by Coats (1976). His model treats k_{rg} hysteresis in an o–g system and it also takes into account the critical gas saturation S_{gcr} in the drainage cycle. The experimental drainage curve obtained in the presence of S_{wc} is first normalised as $k_{\mathrm{rg}} = f(S_{\mathrm{g}})$, where

$$\bar{k}_{\mathrm{rg}} = k_{\mathrm{rg}}/k_{\mathrm{rgro}} = \mathrm{f}(\bar{S}_{\mathrm{g}}) \tag{12.51a}$$

$$\bar{S}_g = \frac{S_g - S_{gcr}}{1 - S_{oc} - S_{wc} - S_{gcr}} \tag{12.51b}$$

where S_{oc} is the residual oil saturation in the drainage cycle and k_{rgro} is k_{rg} at S_{oc} (Fig. 12.11).

The current residual gas saturation S_{gc}^{*} is calculated as a function of the maximum gas saturation S_g^A reached:

$$S_{gc}^{*} = S_{gc} \frac{S_g^A}{1 - S_{wc} - S_{oc}} \tag{12.52}$$

When the saturation decreases from S_g^A to $S_g < S_g^A$, an effective residual saturation is calculated by

$$S_{gc}^{eff} = \omega S_{gc}^{*} + (1 - \omega) S_{gcr} \tag{12.53}$$

where

$$\omega = \frac{S_g^A - S_g}{S_g^A - S_{gc}^{*}} \tag{12.54}$$

Then k_{rg} is obtained from eqn. (12.51a) as a function of

$$\bar{S}_g = \frac{S_g - S_{gc}^{eff}}{1 - S_{wc} - S_{oc} - S_{gcr}} \tag{12.55}$$

When this procedure is used while S_g is increasing, ω will be zero, $S_g^{eff} = S_{gcr}$, and k_{rg} will follow the original drainage curve. If S_g is decreasing, ω will increase to 1 and S_{gc}^{eff} will increase towards S_{gc}^{*}. When S_g reaches S_{gc}^{*}, $k_{rg} = k_{rg} = 0$ (point B).

Another model, proposed by Evrenos and Comer (1969a), is limited because of the assumed analytical form of relative permeabilities.

Hysteresis of oil relative permeability in three-phase flow can be accounted for by using history-dependent two-phase data in three-phase k_{ro} models discussed in Chapter 2, Section 2.7.

12.7 SIMULATION OF NATURALLY FRACTURED RESERVOIRS

The pore structure of reservoir rocks is very complex. As discussed by Warren and Root (1963), the porosity of many reservoirs can be placed in two classes:

(a) Primary porosity, which is highly interconnected and can be correlated with permeability. This would be the porosity of homogeneous rocks such as sandstones.

(b) Secondary porosity formed by fractures, solution channels or vugular voids in porous media. This porosity is the product of geological movements and chemical processes. Although it does not usually contain a large fraction of the hydrocarbon reserves, it greatly affects the flow.

Since the differential equations used in simulation assume continuous properties (including porosity and permeability), they are not strictly applicable to a dual-porosity system. Isolated fractures can be modelled by a row of grid blocks with $\phi = 1$ (void space) and an appropriate permeability (Craft and Hawkins, 1959, p. 188). However, averaging of ϕ and k within large blocks containing several fractures may give misleading results because the character of flow in the fractures and in the matrix is different.

A dual-porosity system is usually idealised according to Fig. 12.12. Here the porous medium is represented by isolated matrix blocks of primary porosity connected by a regular system of fractures. The matrix blocks have most of the porosity and relatively small permeability. Because the pressure drop across the system is dictated by the permeability of fractures, which is high, the main mechanism of production from the matrix blocks is capillary imbibition or gravity drainage and volume expansion with declining pressure. On the other hand, even though the pore capacity of the fractures is small, their conductivity to flow is high; therefore most of the flow takes place in the fractures. Because of low P_c, flow in fractures may be assumed to be fully segregated. In some cases, natural convection can also be an important mechanism in vertical fractures (Peaceman, 1976).

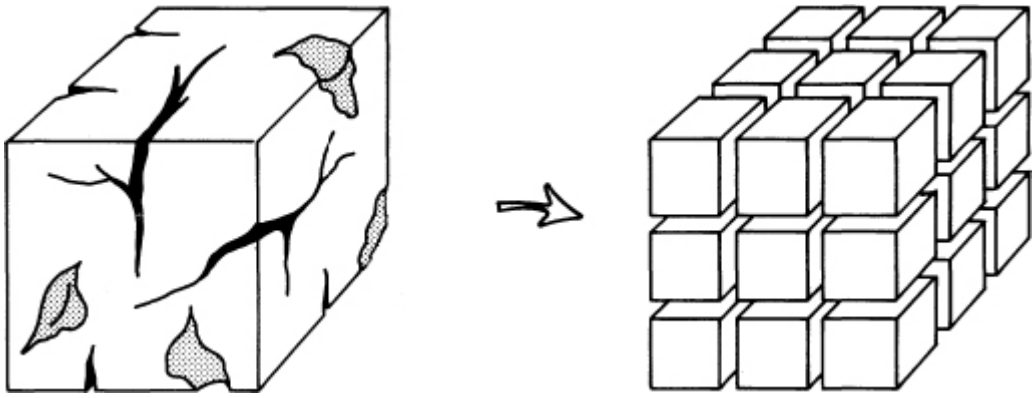


FIG. 12.12. Idealisation of fractured reservoirs.

Methods for simulating dual-porosity systems have been proposed by several authors (Kleppe and Morse, 1974; Yamamoto *et al*., 1971; Asfari and Witherspoon, 1973; Kazemi *et al*., 1976; Rossen, 1977, and others). Kazemi *et al.* (1976) assume that the fracture system can be represented by a continuum and describe the flow in the fractures by a mass balance equation which includes terms to account for flow, accumulation in the fractures and mass transfer from the matrix blocks to the fractures. In addition, since the matrix blocks are not interconnected, they write mass balance equations for the matrix blocks, which involve only accumulation, and mass transfer to the fractures. In this procedure, each computational block can contain several matrix blocks (for example, the element in Fig. 12.12 may be considered to be a single computational block). Rossen (1977) uses the same approach, but instead of solving simultaneously for the matrix blocks, includes the mass transfer as implicit production terms. Thus, fractured reservoirs can be modelled by the techniques developed for continuous reservoirs. The additional feature is the presence of *distributed time-dependent source/sink terms* which account for the fluid exchange between the fractures and the matrix.

12.8 AUTOMATIC TIME-STEP CONTROL

Manual selection of time steps during a simulation study can be frustrating, in particular for cases where the well rates change drastically during a simulation run. The proper selection of time step must ensure: (a) stability of solution, and (b) acceptable time truncation errors. Both of these requirements can be, at least qualitatively, related to the rate of change of pressure and saturations in the reservoir. *The faster the changes, the smaller the time steps which should be used*. For example, the time step during a single-well simulation can vary from 0•01 day immediately after a large rate change to several months when the reservoir is near depletion.

Automatic time-step selection procedures have been suggested for ODEs (Carnahan *etal*., 1969, p. 363; Krogh, 1973; Byrne and Hindmarsh, 1975). A common drawback of these methods is that they usually result in significant increases in computing time. Consequently, reservoir simulators usually employ empirical controls.

For IMPES type methods, use of largest stable time steps will also give the smallest truncation errors (see Chapter 5, Section 5.5.2). Although the maximum stable time step can be estimated from the equations presented in Chapters 5 and 9, experience shows that stability is maintained when the maximum saturation change over the grid is kept below a given limit DSLIM for every time step. The value of DSLIM is typically 5–10%, but it is somewhat problem dependent. The same type of control is employed for SS and SEQ methods. The limit on $\Delta_t S_l$ also controls the truncation errors. For this

reason, the time-step control may also employ a limit DPLIM on the maximum pressure change over the time step (Todd *et al.*, 1972).

An automatic time-step control algorithm may be written as follows:

(a) Compute

$$\mathrm{DSMAX}^n = \max_{ijk} \{|\Delta_t S^n_{lijk}|\}$$

$$\mathrm{DPMAX}^n = \max_{ijk} \{|\Delta_t p^n_{ijk}|\}$$

(b) Calculate Δt^{n+1} as

$$\Delta t^{n+1} = \min \{\Delta t_s, \Delta t_p\}$$

where

$$\Delta t_s = \Delta t^n \frac{\mathrm{DSLIM}}{\mathrm{DSMAX}^n} \qquad \Delta t_p = \Delta t^n \frac{\mathrm{DPLIM}}{\mathrm{DPMAX}^n}$$

The algorithm will maintain saturation and pressure changes close to DSLIM and DPLIM. Because the actual changes are nonlinear, the limiting values could be considerably exceeded with predicted Δt. This can be prevented by adding the following step:

(c) After the time step $n + 1$ is completed, calculate DSMAX^{n+1}, DPMAX^{n+1}. The time step is acceptable if

$$\mathrm{DSMAX}^{n+1} \leq C_1\,\mathrm{DSLIM} \qquad \mathrm{DPMAX}^{n+1} \leq C_2\,\mathrm{DPLIM}$$

where C_1 and C_2 are greater than one. Otherwise, calculate Δt^* as $\Delta t^* = \min \{\Delta t_s, \Delta t_p\}$, where

$$\Delta t_s = \Delta t^{n+1} \frac{\mathrm{DSLIM}}{\mathrm{DSMAX}^{n+1}} \qquad \Delta t_p = \Delta t^{n+1} \frac{\mathrm{DPLIM}}{\mathrm{DPMAX}^{n+1}}$$

set Δt^{n+1} to Δt^* and resolve the equations for time step $n + 1$.

This ensures that the maximum changes do not exceed C_1 DSLIM or C_2 DPLIM. Procedures of this type can be improved by introducing additional constants and by monitoring additional important variables (e.g., amount of overshoot, Todd *et al*., 1972; Chappelear and Rogers, 1974).

12.9 CONCLUDING REMARKS

In addition to topics introduced in this chapter, there are several other techniques which may be important for development or application of models in special cases. Of these, mention should be made of several versions of the method of characteristics (also known as point-tracking, particle-in-cell, etc.).

References can be found, for example, in Harlow and Amsden (1971), Roache (1972), Hirt *et al*. (1974), and Settari *et al*. (1977). An application for polymer flooding is given by Vela *et al*. (1976).

EXERCISE

Exercise 12.1

Derive the relationship between the stock tank densities, R_s and compositional properties of a two-component system.

Solution Outline

Denote ω_{o1}, ω_{gl} the mass fractions of oil component and ω_{o2}, ω_{g2} of the gas component in oil and gas phases, and m_o, m_g the masses of components. Then

$$\omega_{g1} = 0 \qquad \omega_{g2} = 1 \qquad \mathrm{K}_1 = 0 \tag{A}$$

$$\omega_{o2} = \frac{m_g}{m_o + m_g} \qquad \omega_{o1} = \frac{m_o}{m_o + m_g} \tag{B}$$

Using $B_o = V_o\rho_{oSTC}/m_o$, $V_o = (m_o + m_g)/\rho_o$, $R_s = V_{gSTC}/V_{oSTC} = (m_g\rho_{oSTC})/(m_o\rho_{gSTC})$, we get

$$\omega_{o1} = \rho_{oSTC}/(\rho_o B_o)$$
$$\omega_{o2} = \omega_{o1} R_s\rho_{gSTC}/\rho_{oSTC} = R_s\rho_{gSTC}/(\rho_o B_o)$$
$$\mathrm{K}_2 = 1/\omega_{o2}$$

Using eqn. (2.12), ω_{o1}, ω_{o2} and K_2 can be expressed through stock tank densities and R_s:

$$\omega_{o1} = \frac{\rho_{oSTC}}{\rho_{oSTC} + R_s \rho_{gSTC}} \tag{C}$$

$$\omega_{o2} = \frac{R_s \rho_{gSTC}}{\rho_{oSTC} + R_s \rho_{gSTC}} \tag{D}$$

$$K_2 = \frac{\rho_{oSTC} + R_s \rho_{gSTC}}{R_s \rho_{gSTC}} \tag{E}$$

Write these equations with appropriate constants for engineering units.

CHAPTER 13

PRACTICAL CONSIDERATIONS

This chapter will briefly discuss some of the practical questions which must be considered at the stage when one is ready to write or use a simulation program. The treatment will be brief and is intended to give the reader an idea of the steps involved and the decisions which must be made. A detailed treatment of these topics is beyond the scope of this book.

13.1 PROGRAM DEVELOPMENT

13.1.1 Development of the Mathematical Model

This step involves writing down the system of partial differential equations and auxiliary relations, and it is dealt with mostly in Chapter 2. The mathematical model can of course portray only the phenomena that have been included in the formulation of the differential equations. Therefore, this step requires the knowledge of the physical processes taking place in the reservoir and their relative importance. This knowledge is often incomplete for a particular reservoir; nevertheless, one finds that most reservoir processes can be described by one of the several basic *types of models*, such as the black-oil model which has been dealt with throughout this book. The choice of the mathematical model then determines the *class* of problems which can be simulated, using a program based on that particular model. Thus, reservoir simulation models can be classified in several ways according to their features and capabilities:

(a) *PVT treatment*
1. Black-oil models (or gas condensate models, see Spivak and Dixon, 1973)
2. Compositional models
3. Single phase (e.g. gas) models

(b) *Treatment of flow in porous media*
1. Conventional (Darcy's Law)
2. Models with extended Darcy's Law to account for high-velocity effects
3. Fractured porous media models

(c) *Mass and heat transfer*
1. Immiscible flow models (black oil)
2. Miscible flow models
3. Isothermal models
4. Thermal models (steam and *in situ* combustion)

(d) *Geometry*
1. One-dimensional models
2. Two-dimensional models (areal, cross-sectional and radial)
3. Three-dimensional models

5. Other special models (chemical flooding, etc.)

(e) *Co-ordinate system*
1. Cartesian
2. Cylindrical
3. Spherical
4. General curvilinear

(f) *Treatment of surface facilities*
1. Reservoir model only
2. Reservoir + wellbore flow
3. Reservoir + wellbore + surface pipeline network and equipment

This classification is fairly arbitrary and its main purpose is to show the variety of models in existence. Several features may be combined in a single model (in particular, the dimensionality features under (d)).

13.1.2 Development of the Numerical Model

The development of suitable numerical techniques (numerical models) has been our principal objective throughout this book. We have also analysed the limitations of various techniques and have provided guidelines as to their applicability to different problems throughout the text. Some of the major points in the design of a numerical model are:

1. Choice of space discretisation (e.g., 5 or 9 point in 2-D) and the handling of nonlinearities in space.
2. Choice of the method for multiphase flow (IMPES, SEQ or SS).
3. Time approximation and the treatment of nonlinearities in time.
4. Solution technique for algebraic equations.
5. Treatment of individual wells.
6. Design of run-time controls.

13.1.3 Development of the Computer Model (program)

The programmer should keep in mind several considerations when developing a simulation program, in particular a program that is likely to be used frequently and by others. Some of the important considerations are:

1. Running speed (efficiency of coding).
2. Optimum use of available storage (this concern has been partially alleviated recently by the introduction of 'virtual storage' computers).
3. Convenience of input/output (I/O) features for the user.
4. Program portability.
5. Re-start facility.
6. Diagnostic messages.

Many of these considerations pose conflicting demands. For example, the optimisation of running time is, almost always, in conflict with the minimisation of core. A good simulation program is always based on a compromise solution (Kernighan and Plauger, 1974).

Most modern programs are written in the FORTRAN language and we will therefore make use of this language in the examples to be discussed in this chapter.

13.1.3.1 *Programming Efficiency*

Running speed of a code (part of a program) can be seriously affected by inefficient programming. Some particularly useful methods of increasing program efficiency are discussed below.

Use of multiple subscripts. Wherever possible the use of single-subscripted (vector) arrays for two- and three-dimensional problems is recommended. For example, consider the following trivial example:

```
(a)  DIMENSION SW(10, 10, 10), SO(10, 10, 10), SG(10, 10, 10)
     DO 100 I = 1, NX
     DO 100 J = 1, NY
     DO 100 K = 1, NZ
     SO(I,J,K) = 1. – SW(I,J,K) – SG(I,J,K)
   100 CONTINUE
```

Since the compiler translates multiple subscripts into a single subscript (say M), the calculation

$$M = 100*(K-1) + 10*(J-1) + I$$

will be internally performed during every pass through the loop.

The same loop can be coded as follows:

```
(b)  DIMENSION SW(1000), SO(1000), SG(1000)
     M = 0
     DO 100 I = 1, NX
     DO 100 J = 1, NY
     DO 100 K = 1, NZ
     M = M +1
     SO(M) = 1. – SW(M) – SG(M)
   100 CONTINUE
```

In this case, only one addition is required to get the subscript. Further reduction in computer work is obtained by coding the loop as follows:

```
(c)   NXYZ = NX*NY*NZ
      DO 100 M = 1, NXYZ
      SO(M) = 1. – SW(M) – SG(M)
  100 CONTINUE
```

This saves incrementing and checking on loop variables I and J (total NX*NY + NX times). Still faster code can be obtained by 'unscrolling' the loop (Schaefer, 1973).

Even when it is required to use nested loops (I,J,K) in any order, the calculation of M as shown in (b) can be done for the cost of only slightly more than one addition per pass. Although most compilers have the capability to optimise multiple subscripts, their efficiency of optimisation can be seriously degraded when large subroutines are compiled. (See Schaefer, 1973, for an overview of compiler optimisation.) Therefore, *it is recommended that single subscripted variables be used whenever possible*.

Handling of tabulated functions. Most physical properties (fluid and rock properties, etc.) are submitted in the form of tables. The values of a dependent variable for an arbitrary value of the corresponding independent variable are found by interpolation between tabulated entries. This process, also called 'table look-up' requires searching in the table of arguments (independent variable). Various searching algorithms are analysed by Knuth (1973); he shows that for a small number of entries sequential search is the most efficient, while for large tables binary search is optimal. For tables with 10–20 entries, sequential table look-up is recommended (Knuth, 1973; Section 6.2.1). Efficiency of table look-up is greatly improved if the arguments can be *equally spaced*. Then the search for the right entry will require only one multiplication regardless of the number of entries. Subroutines for sequential and binary table look-up are included in Appendix B. These routines assume that the table of arguments is in increasing order, but can be arbitrarily spaced.

Use of subroutines. Calling a subroutine requires a certain amount of time for internal 'book-keeping' which increases with the number of arguments transferred. Frequent calling of subroutines (e.g., within a loop) should be avoided. For example, it is better to duplicate the code for table look-up whenever required than to call it as a subroutine. When it is necessary to call a subroutine frequently, arguments should be passed via common blocks rather than through the argument list. Some aspects of the development of subroutines are discussed by Cody (1974).

Use of input/output (*I/O*). Whenever possible, unformatted, as opposed to formatted, I/O should be used. This is important for files used to read and write restart records and scratch (auxiliary) files.

The time required to execute an I/O statement (formatted or not), is a function of the number of items in the I/O list. If an array A is to be written on a scratch tape for future use, this can be done as follows:

```
(a)  DIMENSION A(1000)
     NB = 850
     WRITE (6) (A(I), I = 1, NB)
```

This WRITE represents 850 data items. The same result can be achieved by

```
(b)  DIMENSION A(1000)
     WRITE (6) A
```

which represents only one data item and will execute much faster. Notice that here we have also written 150 additional words of A which are not required; consequently the size of the output record has also increased.

Miscellaneous

1. Avoid mixed arithmetic (i.e., INTEGER to REAL conversions)
2. Replace 2∗A by A + A
3. Multiplication is faster than division on most machines, therefore replace A/2. by A∗.5 etc.
4. Use A∗A instead of A∗∗2.
5. Remove all constant expressions from loops.

Most compilers are capable of optimising some of the above, but it is always better to do it at the source code level. As a general rule, *closest attention should be given to the part of the program which is executed most often*. More discussion of FORTRAN optimisation can be found in Larson (1971).

Further increase in speed can be realised by machine-dependent optimisation. For example, the segment of the program that performs the most work can be coded to utilise the parallel processing capability of the machine or it can be coded in machine language. Parallel processing (Fernbach, 1973) can increase the speed in proportion to the number of processors utilised, but not all algorithms are amenable to this approach. Coding in machine language compared to compiler-translated code can increase speed by as much as 2–3 times.

13.1.3.2 *Optimisation of Storage*

Storage requirements can be reduced by

1. Equivalencing arrays which store temporary information to other arrays.
2. Using variables in a common block for names of local (auxiliary) variables which are used only within a subroutine.
3. Re-computing values repeatedly rather than storing them.
4. Overlaying parts of the core during execution. Typically, the core occupied by the binary code of subroutines used to initialise the simulator can be overlayed after the initialisation has been completed.
5. On some computers it is possible to specify the size of I/O buffers so that they are smaller than the default value.
6. Internal disking (writing and reading from a scratch tape). This may be necessary, for example, when the use of Gaussian elimination for a matrix of large band-width requires more than the available core.
7. Dynamic allocation of storage during execution.

Almost all of the above steps will decrease the running speed. Use of virtual storage will increase the execution time only moderately compared to running the same program in core, but the residence time (i.e., the total elapsed time the job is active) increases several times because of internal I/O between the virtual storage and core.

13.1.3.3 *User-oriented Features*

Some features that are important for good simulators are:

1. Options to suppress or expand print-out at any given time step.
2. Options to read data in different ways so that a minimum of data preparation is required.
3. Coding to check the input data for consistency and to ensure that the data are physically meaningful. This feature is particularly useful in ‘debugging’ the data.
4. Re-start capability.

It is also useful to be able to display results in graphical form (printer plots and/or maps). Obviously, there is no limit to the number of optional features that can be included in a single program and the decision is again a matter of compromise.

13.1.3.4 *Program Portability*

In order to minimise the problems created by differences between compilers, the programmer should avoid using those FORTRAN features which are unique to a given compiler. Some useful rules are:

1. Avoid use of special characters in variable names.
2. Use a maximum of six characters for variable names.
3. Use Hollerith statements when dealing with literals, and in storing literals use only A4 combinations.
4. Do not exceed 132 print positions in output FORMAT statements.
5. Avoid use of any bit operations (such as masking, etc.).
6. Avoid coding in machine language and use of special I/O features (such as BUFFER I/O on CDC or ASYNCHRONEOUS I/O on IBM equipment).

Again, the last requirement, in particular, is in conflict with consideration of program speed, as discussed above (see Aird *et al*., 1977).

13.2 PROGRAM USAGE

A typical use of a simulation program is to design an oil recovery project. This is called a *simulation study* and it is usually planned in a number of steps which are summarised below. Many of the questions that must be resolved during a study very closely parallel the steps involved in developing a program (Sections 13.1.1 and 13.1.2), but from a different viewpoint. We will first summarise the steps usually required and then discuss some of them in more detail. As the reader will appreciate after reading this section, the application of simulation technology is more of an art than a science and therefore the guidelines given here should always be scrutinised in the light of a particular problem on hand (Coats, 1969).

13.2.1 Steps Involved in a Simulation Study

A typical study will involve the following steps:

(a) Define the reservoir engineering problem and the technical and economic objectives of the study.
(b) Gather all available data and decide if additional data will be required.
(c) Interpret available geological and petrophysical data.
(d) Analyse the PVT behaviour of fluids involved.
(e) Select the type of simulator to be used and design the model (i.e., dimensions, grid, solution method, etc.).

(f) Adjust the parameters of the model by 'history matching' actual historical performance of the reservoir.
(g) Predict the future behaviour under different conditions to accomplish the objectives of the study.
(h) Interpret the results, make recommendations, edit the output and report.

Although the above steps are fairly typical, they may vary widely from one problem to the next. Simulation studies vary from simple runs with 'off-the-shelf' simulators to large undertakings requiring the use of several models and extensive special programming. Furthermore, the direction of the study may change during its course as we gain a better understanding of the depletion process being modelled and relative importance of various parameters. This may necessitate not only some change in the dimensionality and type of model used, but it may also indicate the need for more field or laboratory data. These aspects are discussed by Coats (1969) and can be traced in many papers describing simulation studies. Papers by McCulloch *et al*. (1969), Weaver (1972) and Dandona and Morse (1975) are good examples of black-oil model applications.

We will now briefly discuss the steps listed above, with the exception of (e) and (f) which will be treated in subsequent sections.

Step (a)*:* It is important to have the questions which are to be answered by simulation *clearly defined*. The approach to the study depends in an essential way on its objectives. The questions of 'economical' nature are usually of the type:

Which is the best recovery scheme for the reservoir?
What is the optimum production rate?
What is the optimum well spacing?
Is infill drilling economical?

In other situations questions of 'feasibility' type may have to be answered:

Will a certain process work within a reasonable range of controlling parameters?

Questions of'remedial' type may also be asked:

What is the cause of an operational problem and can it be cured?

Step (*b*)*:* Data gathering may be time consuming for large reservoirs. The available data should be reviewed for completeness and reliability. All suspect data should be adjusted if appropriate or discarded.

Step (c)*:* Available geological and petrophysical data are used to construct a 'geological model' of the reservoir which defines the structure, thickness, porosity and

permeabilities, stratification, faults, and other physical features of the reservoir. The 'geological picture' of the reservoir is very important and frequently it is re-evaluated later during the history match process. The methods used to develop geological data for simulation are described by Harris (1975).

Step (*d*)*:* PVT analysis will show if the fluids must be represented compositionally, or if they can be represented by a black-oil system. Compositional models are far more complex than the black-oil models discussed in this book. All data should be checked and smoothed.

Step (*e*)*:* Running the model to predict future behaviour is a relatively straightforward part of the study. For fields that are approaching depletion, care must be taken to simulate properly the actual operating practices, such as installation of pumps and compressors, and workover jobs. Some of these may require special programming.

Step (*f*)*:* Interpretation of the results may involve some additional calculations such as material balance calculations on processing facilities, pipeline calculations, and frequently economic analyses. Editing the output for reporting is necessary because the printed output is usually voluminous and must be condensed. However, all data and results generated should be kept for future reference, for example, on a tape. This is particularly important in cases where a future 'update' of the study is anticipated.

13.2.2 Selection and Design of the Model

Construction of the numerical model of a reservoir is one of the 'grey' areas where the art and science of simulation meet. The 'best' reservoir model is the simplest model which still represents realistically all important aspects of the physical process of interest. However, because of the rather limited knowledge one usually has of the reservoir and fluid parameters, there is often a tendency to use a more sophisticated model than is actually necessary. It is always beneficial to first see if the use of an analytical method or simple computer model will answer the question. Such a preliminary study will usually result in an improved understanding of the problem.

When a simulation study is necessary, several factors must be considered in the selection and design of the model, such as the following:

The kind of answers required;
accuracy of the answers required;
uncertainty in reservoir definition;
uncertainty in the field data available for simulation;
uncertainty in fluid and rock properties;
understanding of the physics of the actual process (mathematical model);

errors inherent in the assumptions made in the development of the numerical model; and
truncation errors.

Many of these factors are hard to quantify, and, consequently, the reasoning for selection and design of a model for a reservoir will differ from case to case. We will now discuss some of the more important aspects in detail.

13.2.2.1 *Dimensionality of the Model*

Since the computer time and storage requirements increase drastically with the number of dimensions, the smallest possible dimension should be used.

Single-block models and one-dimensional (1-D) models can be treated analytically in certain cases. Often 1-D simulators are used for estimates in highly idealised situations, for example, for reservoirs with very limited known description, and in the simulation of laboratory experiments. 1-D models can be also adequate in situations where only sensitivity of answers to certain variables is of interest.

2-D Models are adequate under a variety of conditions. Reservoirs with good vertical communication can be usually represented by a 2-D areal model using the vertical equilibrium (VE) concept (see Chapter 12). Behaviour of single wells can be studied in two-dimensional radial coordinates in most cases. Exceptions are wells close to the boundaries of the reservoir or close to fluid contacts in horizontal flooding, where the assumption of symmetry is not reasonable. Many reservoirs may be successfully studied by looking only at a typical part of the reservoir, modelled usually in 2-D. These so-called 'window' models are much less expensive than full three-dimensional studies. A classical example is the simulation of a quarter of a five-spot by a 2-D areal model.

The questions of appropriate dimensionality and associated errors are discussed by Coats (1969) and Price (1971). Examples of cross-sectional studies of reef reservoirs are provided in the papers by McCulloch *et al*. (1969) and DesBrisay *et al*. (1975). Finally, we remark that frequently a simulation study involves use of several models. A typical example is the use of a 2-D cross-sectional model to generate VE curves and/or a 2-D radial model to generate well-coning curves, used in conjunction with a 2-D areal model. An example of a study, progressing from 1-D to 3-D is found in Spivak *et al*. (1975).

Treatment of aquifers must be also considered when selecting the dimensionality of the model.

13.2.2.2 *Treatment of Reservoir Fluids*

Again, the simplest representation consistent with the goals of the study should be used. In the black-oil model representation, the running time of the SS and SEQ method increases sharply with increasing number of phases, but it is essentially unchanged for the IMPES method (Chapter 9, Section 9.3.3). Therefore, the elimination of a phase that is immobile from the model may, in some cases, result in significant savings.

When the compositional effects are important, they can be often approximated by a black-oil simulator. Because compositional simulators are generally more expensive to run, this will result in considerable savings. Approaches of this type are applied to oil and gas condensate reservoirs. Miscible displacement can be also treated by a black-oil model. We have discussed these techniques in Chapter 12.

13.2.2.3 *Selection of Grid*

Following are rough guidelines for grid design:

(a) The orientation of the grid should be, if possible, along the known permeability orientation. In cross-sectional problems, model layers should follow geological layers.
(b) Grid definition should be increased in the areas of interest and/or where the solution is expected to change rapidly. Smaller blocks are usually required around wells.
(c) Grid definition should be increased in areas of heterogeneity, if accurate answers are required.
(d) Abrupt and large changes in grid sizes introduce large truncation errors. They should be avoided in the construction of an irregular grid.
(e) In an areal simulation, all wells should be separated by several blocks if the effect of well pattern, infill drilling etc., is to be studied.

The most important rule is, however, that *whenever possible*, *the necessary grid definition should be determined by a grid sensitivity study*. Such a study amounts to performing a series of simulation runs with increasing grid definition until the computed results do not change within the accuracy required. Grid sensitivity studies are particularly important for multiphase flow. They are commonly performed for cross-sectional problems to determine the minimum number of layers required. In areal and 3-D simulations, the size of the problem is frequently too large to perform a grid refinement experiment. In such cases, grid sensitivity can be studied on a small selected part of the reservoir.

In many cases, the accuracy of solution cannot be determined because the grid size is limited by computer limitations or by cost factors. Only in simple cases can the spatial truncation error be determined by comparison with exact solutions. Truncation errors

are often termed ‘numerical dispersion’ (Lantz, 1971) and they are of particular importance in simulating miscible displacement, compositional phenomena and other problems requiring high accuracy. Conventional finite-difference techniques are often inadequate in such situations because the required number of blocks is unreasonably large. In such cases special models not discussed in this book (point-tracking, variational) must be used.

13.2.2.4 *Selection of the Method of Solution and the Technique for Solving the Algebraic Equations*

When a choice (of IMPES, SEQ or SS) is available, the cheapest method of solution should be selected. This may involve some tests of stability by running with a typical set of data. As discussed in Chapter 9 (Section 9.3.3), progressing from IMPES to SEQ to SS increases drastically the running time per time step. In practice, the speed is expressed in terms of time/grid block/time step in order to eliminate the influence of grid size.

Because the stability also increases when going to more implicit methods, the best overall efficiency is generally achieved by IMPES for ‘easy’ problems and by SS (with implicit transmissibilities) for ‘difficult’ problems.

The choice of the technique for solving the matrix equation is also critical. As discussed in Chapter 8 (Sections 8.3 and 8.4) and in Chapters 10 and 11, direct methods are preferred for smaller problems and iterative methods for larger problems. Iterative methods also become faster when it is necessary to iterate (update the coefficients) within the time-step solution. With iterative methods, this updating can be done during the solution process with only a small decrease in the convergence rate, while for direct methods work increases in proportion to the number of iterations required per time step.

Apart from the above considerations of optimising the running speed, the decision will be also influenced by the type of study to be performed. For a problem where only a limited number of runs are required, it may be cheaper to use a more implicit method and a direct elimination technique because of their reliability. On the other hand, for a large study with a number of history match runs, it may be worthwhile to optimise both the implicitness of the method and the solution technique.

13.2.2.5 *Selection of Time Steps and Run-time Control Parameters*

Time step size must be selected to ensure both stability and accuracy. The first requirement relates the maximum stable step size to the grid size. For example, for the IMPES method this relationship is given by eqn. (5.60), and by eqn. (9.25) when the stability is limited by transmissibilities. The maximum time step that can be used is proportional to grid size also for other solution methods. Therefore, with grid

refinement the cost of a computer run will increase *more* than what is indicated by the grid ratio, because the time step must also be decreased.

In practice, both stability and accuracy can be achieved by an automatic time step selection using methods discussed in Chapter 12.

The control parameters which must be selected may include maximum pressure and saturation changes per time step, convergence tolerances, maximum number of iterations, number of iterations between updates of coefficients, etc. Their selection will be different from case to case and some experimentation may be necessary to minimise running time. In general, the tolerances should be 'tighter' when the problem solved is more difficult.

13.2.3 History Matching

History matching is an important part of any study. Matching model predictions with historical performance of the reservoir provides the only practical test of the validity of a computer model of the reservoir. History matching consists of adjusting the parameters of the model (such as permeability, porosity, etc.) until the computed results for the historical period are 'close' to the historical data. The historical information can be of various types. For a given production schedule, the matching data usually are:

observed WOR and GOR;
observed average pressures (shut-in pressures), or pressures in observation wells; or
observed flowing well pressures.

In cases when a well is producing at a constant total withdrawal rate or a constant pressure, the matching variable may be taken as the oil rate.

Less frequent matching data are:

breakthrough time; and
transient pressure during a draw-down or build-up test (Chain *et al*., 1976).

The process of history matching is time consuming, sometimes frustrating, and it usually represents a large portion of the cost of a study. History match is usually done by manually adjusting data through a trial-and-error procedure. There has been considerable research effort devoted to automating this process by using 'inverse simulation'. In this procedure equations are solved for the values of selected reservoir parameters, so that the differences between the computed results and observed behaviour are minimised (Coats *et al*., 1970; Dupuy *et al*., 1971; Slater and Durrer,

1971; Thomas and Hellums, 1972; Chen *et al*., 1974; Carter *et al*., 1974; Chavent *etal*., 1975; Gavalas *etal*., 1976; Wasserman and Emanuel, 1976; Bishop *et al*., 1976). However, at the current stage of development these techniques are of limited use for practical problems, except in simple cases such as single-phase (gas) reservoirs.

The general rule in manual history matching is to *change the parameters which have the largest uncertainty and also the largest influence on the solution*. The sensitivity of the solution to some of the parameters is frequently established during the history match process itself. There are no hard-and-fast rules about how to proceed in any given case. The following hints may be useful:

(a) The match of average pressure is affected by fluid volumes in-place, size of the aquifer and the degree of communication between the reservoir and the aquifer. However, a poor match of WOR and GOR will also cause a poor match for the average pressure.

(b) Pressure draw-down is affected most by horizontal permeability and by skin effects.

(c) WOR and GOR depend primarily on draw-down (i.e., again on permeability), but also on the position of fluid contacts, and the thickness of the transition zone (which depends on P_c). The shape of the curves after breakthrough depends on the relative permeability curves, but the breakthrough time depends mainly on the end-points of these curves, i.e., the effective permeabilities with only one of the phases flowing. In areal and 3-D studies, models cannot accurately match the individual well behaviour without the special treatment discussed in Chapter 9 (Section 9.7). In such cases, the match consists of finding the appropriate pseudo-functions. Studies of this type may be complemented by matching single-well performance to improve the accuracy of reservoir description.

(d) Matching breakthrough times is one of the most difficult tasks. Breakthrough times are sensitive to truncation errors (numerical dispersion) and their accurate matching requires finer grid than would otherwise be necessary.

The quality of the match and therefore the amount of confidence we can have in the model depends substantially on the amount of historical data available for the match. With insufficient data (for example, if pressure data are not available), a match can be obtained with different sets of reservoir descriptions. On the other hand, when large amounts of data are available, an unsuccessful attempt for a match will indicate that some of the basic assumptions made in the development of the model may have to be revised (geologic structure, PVT behaviour, extent of reservoir, presence of aquifer, etc.). In some cases an unsuccessful match may also indicate inaccuracies in the data.

In most black-oil simulations, both pressure and rate (WOR, GOR) data are usually necessary to obtain a good match. Even when a good match has been obtained with one set of historical data, there is no guarantee that the model will accurately match additional historical data without further adjustment of reservoir parameters. Therefore it is important to use the maximum available amount of data in the matching process.

Obviously, the reliability of the predictions of future performance also depends on the amount of data available for the match. Even with a good history match, the reliability of predictions decreases as we look farther and farther ahead. It is therefore beneficial to 'update' a simulation study after a period of time by matching with the new historical data. This will normally result in some adjustment of predictions for the future period.

The techniques of history matching may also be used to interpret laboratory data (e.g. Archer and Wong, 1973).

13.3 CONCLUDING REMARKS

Considerable practical experience in reservoir engineering and computer simulation is required in order to take full advantage of the technology of reservoir simulation.

Because of the fact that assumptions are involved at several stages of the process of model development and the flow equations can only be solved approximately, it is necessary to use good engineering judgement in conducting reservoir studies and in the interpretation of results obtained.

APPENDIX A

REVIEW OF MATRIX ALGEBRA

A.1 INTRODUCTION

This appendix is intended to provide a review of the properties of vectors and matrices that are essential for a study of reservoir simulation. Many excellent books are available on this subject and should be consulted for a thorough understanding of the concepts introduced here. In particular, the books by Bellman (1960), Pease (1965) and Lancaster (1969) are recommended for further study.

The next section of this appendix provides some basic concepts and definitions. This is followed by a section dealing with fundamental theorems important for reservoir simulation.

A.2 BASIC CONCEPTS

A.2.1 Field

DEFINITION: A *field, F*, is a collection of numbers within which the operations of addition, subtraction, multiplication, and division, except by zero, are always possible and give a unique element of the field. Note that the real numbers form a field, while the positive real numbers do not (since subtraction is not always possible).

A.2.2 Vector

DEFINITION: A *vector* is a set of n numbers from F arranged in a definite order. The numbers so arranged are called the *elements* or *components* of the vector. The standard ordering of a vector is

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad x_i \in F \tag{A.1}$$

where the symbol ϵ means a 'member of'. The same elements when written in a row, form the *transpose* of x:

$$\boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, \ldots, x_n)$$

where n is the *dimension* of x or x^{T}.

DEFINITION: Two vectors are said to be equal if corresponding components are equal.

Let x and y be two vectors with n elements each from field F. Let α and β be two scalars from field F. Then we have

$$\text{DEFINITION:} \quad \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y} \qquad z_i = x_i + y_i \qquad i = 1, \ldots n$$

And

$$\text{DEFINITION:} \quad \boldsymbol{z} = \alpha \boldsymbol{x} \qquad z_i = \alpha x_i \qquad i = 1, \ldots n$$

The following properties can be derived from the above two definitions:

$$\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x} \quad \text{(commutative)} \tag{A.2}$$

$$\boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) = (\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} \quad \text{(associative)} \tag{A.3}$$

$$\boldsymbol{x} + \boldsymbol{0} = \boldsymbol{x} \quad (\boldsymbol{0} \text{ is the null vector}) \tag{A.4}$$

$$\boldsymbol{x} + (-\boldsymbol{x}) = \boldsymbol{0} \quad (-\boldsymbol{x} \text{ is the negative of } \boldsymbol{x}) \tag{A.5}$$

$$\alpha(\beta \boldsymbol{x}) = (\alpha\beta)\boldsymbol{x} \quad \text{(associative)} \tag{A.6}$$

$$(\alpha + \beta)\boldsymbol{x} = \alpha \boldsymbol{x} + \beta \boldsymbol{x} \quad \text{(distributive)} \tag{A.7}$$

$$\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha \boldsymbol{x} + \alpha \boldsymbol{y} \quad \text{(distributive)} \tag{A.8}$$

$$\alpha \boldsymbol{x} = \boldsymbol{x}\alpha \quad \text{(commutative)} \tag{A.9}$$

$$\alpha \boldsymbol{x} = \boldsymbol{x} \quad \text{if} \quad \alpha = 1 \tag{A.10}$$

A.2.3 Linear Vector Spaces

DEFINITION: A *linear vector space* is a set of vectors, S, such that: (1) the sum of any two vectors from S is also a member of S, and (2), the product of any scalar α with a member of S is also in S.

The linear vector space is spanned over the field F when the elements of all the vectors in S and scalars α are from the field F.

DEFINITION: A set of j vectors $\boldsymbol{x}_1, x_2, \ldots, x_j$ is *linearly independent* in the field F if there exists no set of scalars $\alpha_1, \alpha_2, \ldots, \alpha_j$ of the field F, not all zero, such that

$$\alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2 + \cdots + \alpha_j \boldsymbol{x}_j = \boldsymbol{0} \tag{A.11}$$

Otherwise the vectors are said to be linearly dependent. Note that if a solution to (A.11) exists with $\alpha_i \neq 0$, then

$$\boldsymbol{x}_i = -\frac{1}{\alpha_i} \sum_{k=1, k \neq i}^{j} \alpha_k \boldsymbol{x}_k \tag{A.12}$$

In eqn. (A.12) $\boldsymbol{x}_i$ is expressed as a linear combination of other vectors in the set, hence it is not linearly independent of these vectors.

DEFINITION: A linear vector space is of *dimension* k if it contains at least one set of k linearly independent vectors (excluding null vectors), but does not contain any set of $k + 1$ linearly independent vectors.

DEFINITION: The set of k linearly independent vectors for a vector space, S, of dimension k is a *basis* for S. Note that all other vectors in S can be formed by a linear combination of the basis vectors.

A.2.4 Linear System of Equations. Matrices

The system of linear equations

$$\sum_{j=1}^{n} a_{ij} x_j = k_i \qquad i = 1, 2, \ldots, n \tag{A.13}$$

May be written as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{k} \tag{A.14}$$

where $\boldsymbol{x}$, $\boldsymbol{k}$ are n-dimensional vectors and $\mathbf{A}$ is a square $(n \times n)$ array of numbers with n rows and n columns:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & \cdot & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \tag{A.15}$$

The operation $\mathbf{A}\boldsymbol{x}$ is *defined* so that it results in the system of equations given by (A. 13). We can also express $\mathbf{A}$ as

$$\mathbf{A} = [a_{ij}]$$

The *main diagonal* of $\mathbf{A}$ is the set of elements on the diagonal running from the top left to the bottom right corner. The matrix is called *diagonal* if

$$a_{ij} = 0 \qquad i \neq j \tag{A.16}$$

and a matrix is called the *identity matrix*, $\mathbf{I}$, if

$$a_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \tag{A.17}$$

A matrix $\mathbf{B}$ with elements b_{ij} such that

$$b_{ij} = a_{ji} \qquad \text{for all } i \text{ and } j \tag{A.18}$$

is called the *transpose* of $\mathbf{A}$, $\mathbf{A}^{\mathrm{T}}$. The *null* matrix, $\mathbf{0}$, has zero elements.

DEFINITION: The sum of two matrices $\mathbf{A}$, $\mathbf{B}$ is a matrix $\mathbf{C}$ with elements

$$c_{ij} = a_{ij} + b_{ij} \qquad i = 1, \ldots, n \qquad j = 1, \ldots, n \tag{A.19}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are all $n \times n$ matrices

DEFINITION: The product of a scalar α with matrix $\mathbf{A}$ is a matrix $\mathbf{C}$ with elements

$$c_{ij} = \alpha a_{ij} \tag{A.20}$$

DEFINITION: The product of matrices $\mathbf{A}$ and $\mathbf{B}$ is a matrix $\mathbf{C}$ defined by

$$\mathbf{C} = \mathbf{BA} \qquad c_{ij} = \sum_{k} b_{ik} a_{kj} \tag{A.21}$$

Note that generally **AB** # **BA**.

DEFINITION: Matrices **A** and **B** *commute* if $\mathbf{AB} = \mathbf{BA}$ (A.22)

DEFINITION: The inverse of **A** (if it exists) is a matrix A^{-1} defined by

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \tag{A.23}$$

The following properties may be derived from the above definitions

$$(\mathbf{A} + \mathbf{B})x = \mathbf{A}x + \mathbf{B}x \tag{A.24}$$

$$\mathbf{A}(x + y) = \mathbf{A}x + \mathbf{A}y \tag{A.25}$$

$$\alpha\mathbf{A}x = (\alpha\mathbf{A})x = \mathbf{A}(\alpha x) = (\mathbf{A}x)\alpha \tag{A.26}$$

$$(\mathbf{A}x)^{\mathrm{T}} = x^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \tag{A.27}$$

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC} \tag{A.28}$$

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC} \tag{A.29}$$

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C} \tag{A.30}$$

$$\mathbf{0A} = \mathbf{A0} = \mathbf{0} \tag{A.31}$$

$$\mathbf{IA} = \mathbf{AI} = \mathbf{A} \tag{A.32}$$

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{AA}^{-1} = \mathbf{I} \tag{A.33}$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{A.34}$$

$$(\mathbf{ABC}\ldots)^{-1} = (\ldots\mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}) \tag{A.35}$$

A.2.5 Determinant of a Matrix

The *determinant* of a matrix is a number denoted by det (**A**). For a 2 × 2 matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}$$

For an arbitrary matrix **A**,

$$\det(\mathbf{A}) = \sum_{k=1}^{n} a_{ik}\alpha_{ik} = \sum_{k=1}^{n} a_{kj}\alpha_{kj}$$

for any row i or any column j. In the above equation α_{ik} is the *cofactor* of a_{ik},

$$\alpha_{ik} = (-1)^{i+k} \det(\mathbf{M_{ik}})$$

and $\mathbf{M_{ik}}$ is the $(n-1) \times (n-1)$ matrix obtained by removing the ith row and kth column. It is the *minor* of a_{ik}.

DEFINITION: Matrix **A** is *singular* if $\det(\mathbf{A}) = 0$.

A.2.6 Eigenvalues and Eigenvectors

A vector $\boldsymbol{x}_i$, that satisfies

$$\mathbf{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i$$

is called an *eigenvector* of **A**, and the scalar λ_i is the corresponding *eigenvalue*.

The *characteristic equation* of an $(n \times n)$ matrix **A** is

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

The determinant is a polynomial in λ of degree n. The roots of the characteristic equation are the eigenvalues of **A**.

DEFINITION: The spectral radius of **A** is

$$\rho(\mathbf{A}) = \max_i |\lambda_i|$$

A.2.7 Norms of Vectors and Matrices

A norm of a vector $\boldsymbol{x}$ is a real number, denoted by $\|\boldsymbol{x}\|$, which satisfies all of the following conditions:

(a) $\|\alpha\boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\|$ where α is a scalar
(b) $\|\boldsymbol{x}\| + \|\boldsymbol{y}\| \geq \|\boldsymbol{x} + \boldsymbol{y}\|$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$
(c) $\|\boldsymbol{x}\| \geq 0$ with equality if and only if $\boldsymbol{x} = \boldsymbol{0}$

The l_p norm of a vector, denoted by $\| \|_p$, is defined by

$$\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p} \qquad \text{for all} \quad p \geq 1$$

Three norms often used are:

$$l_1 : \|\boldsymbol{x}\|_1 = \sum_i |x_i|$$

$$l_2 : \|\boldsymbol{x}\|_2 = \left(\sum_i x_i^2\right)^{1/2} \qquad \text{(Euclidean norm)}$$

$$l_\infty : \|\boldsymbol{x}\|_\infty = \max_i |x_i|$$

A matrix norm is a real number that satisfies the corresponding conditions (a), (b) and (c); and in addition the following condition is also satisfied:

(d) $\|\mathbf{A}\| \, \|\mathbf{B}\| \geq \|\mathbf{AB}\|$

A matrix norm is consistent with the vector norm if it satisfies

(e) $\|\mathbf{A}\boldsymbol{x}\| \leq \|\mathbf{A}\| \, \|\boldsymbol{x}\|$ for all $\boldsymbol{x}$

If the equality in (e) holds for at least one $\boldsymbol{x} \neq \boldsymbol{0}$, then the matrix norm is said to be subordinate to the vector norm.

The matrix norms of $\mathbf{A}$ subordinate to l_1, l_2 and l_α norms of $\boldsymbol{x}$ are:

$$\|\mathbf{A}\|_1 = \max_k \sum_{i=1}^{n} |a_{ik}| \qquad \text{(maximum column sum)}$$

$$\|\mathbf{A}\|_2 = [\rho(\mathbf{A}^{\mathrm{T}}\mathbf{A})]^{1/2}$$

$$\|\mathbf{A}\|_\infty = \max_i \sum_{k=1}^{n} |a_{ik}| \qquad \text{(maximum row sum)}$$

Some properties of norms

1. $\|\boldsymbol{x} - \boldsymbol{y}\| \geq |\,\|\boldsymbol{x}\| - \|\boldsymbol{y}\|\,|$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors
2. $\|\mathbf{A}\| \geq \rho(\mathbf{A})$ for any norm of matrix $\mathbf{A}$
3. $\|\mathbf{A}\|_2 = \rho(\mathbf{A})$ if $\mathbf{A}$ is a symmetric matrix
4. $\dfrac{1}{\|\mathbf{A}^{-1}\|} \leq |\lambda| \leq \|\mathbf{A}\|$ where $\mathbf{A}$ is a non-singular matrix, and λ is any eigenvalue of A.

A.2.8 Additional Definitions

NOTE: $\mathbf{A}$ is an $n \times n$ matrix and all vectors have n elements unless otherwise specified.

1. The *inner* (or *scalar*) product of two members $\boldsymbol{x}$ and $\boldsymbol{y}$ of a vector space is defined by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i$$

Note that

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle = \|\boldsymbol{x}\|_2^2$$

2. The matrix $\mathbf{A}$ is *orthogonal* if

$$\mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}}$$

3. The matrix $\mathbf{A}$ is *diagonally dominant* if

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| = \alpha_i \qquad \text{for all } i$$

and *strictly diagonally dominant* if strict inequality is valid for all i.

4. The matrix $\mathbf{A}$ is *tridiagonal* if

$$a_{ij} = 0 \qquad \text{for} \quad |i - j| > 1$$

5. The matrix $\mathbf{A}$ is *block diagonal* if

$$\mathbf{A} = \begin{bmatrix} \mathbf{B}_1 & & & \\ & \mathbf{B}_2 & & \\ & & . & \\ & & & \mathbf{B}_k \end{bmatrix}$$

where each $\mathbf{B}_i$ ($i = 1, 2, \ldots, k$) is a square matrix. $\mathbf{B}_i$ are not necessarily of the same order.

6. The matrix **A** is *upper triangular* if

$$a_{ij} = 0 \qquad i > j$$

and *lower triangular* if

$$a_{ij} = 0 \qquad j > i$$

7. The matrix **A** is *irreducible* if there exists no permutation transformation

$$\mathbf{PAP}^{-1}$$

which reduces **A** to the form

$$\begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix}$$

where **S** and **Q** are square sub-matrices of order s and q, respectively, with $s + q = n$, and **0** is a $s \times q$ null matrix.
The *permutation* matrix **P** has one non-zero element in each row and column and all of these elements are ones.

8. The matrix **A** is *irreducibly diagonally dominant* if it is *irreducible* and *diagonally dominant* with

$$|a_{ii}| > \alpha_i \qquad \text{for at least one } i$$

9. Matrices **A** and **B** are *similar* if

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}$$

and T is non-singular. The above operation on **A** is called a *similarity transformation*. Similar matrices have the same eigenvalues.

10. Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are *orthogonal* if

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$$

Orthogonal vectors are linearly independent.

11. **A** is *positive definite* if

$$\langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x} \rangle > \mathbf{0} \qquad \text{for all } \boldsymbol{x} \neq \boldsymbol{0}$$

and **A** is *positive semi-definite* if

$$\begin{aligned} \langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x} \rangle &\geq 0 \qquad \text{for all } \boldsymbol{x} \text{ and} \\ &= 0 \qquad \text{for at least one } \boldsymbol{x} \neq \boldsymbol{0} \end{aligned}$$

A.3 SOME FUNDAMENTAL THEOREMS

NOTE: A is a real $n \times n$ *matrix.*

A.3.1

A symmetric matrix has n real eigenvalues and n mutually orthogonal eigenvectors.

A.3.2

If **A** is *strictly* or *irreducibly diagonally dominant*, then **A** is *non-singular*. If, in addition all diagonal entries of **A** are positive, then the real part of the eigenvalues of **A** is also positive (Varga, 1962, p. 22).

A.3.3

If **A** is *symmetric* and *strictly diagonally dominant* or *irreducibly diagonally dominant* with positive real diagonal entries, then **A** is *positive definite.*

A.3.4

System of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{k}$ has a unique solution if $\det(\mathrm{A}) \neq 0$.

A.3.5

If **A** is *strictly* or *irreducibly diagonally dominant*, $\mathbf{A}\boldsymbol{x} = \boldsymbol{k}$ has a unique solution.

A.3.6

If $\mathbf{A}$ is a real matrix such that $a_{ii} \geq 0$ and $a_{ij} \leq 0$ for $i \neq j$, then $\mathbf{A}$ is singular if and only if

$$\sum_{j=1}^{n} a_{ij} = 0 \qquad i = 1, \ldots, n$$

(Taussky, 1949).

A.3.7

$\mathbf{A}\boldsymbol{x} = \boldsymbol{k}$ has a unique solution if $\mathbf{A}$ is *positive definite*

A.3.8 (Faddeeva, 1959, p. 60)

$$\lim_{m \to \infty} \mathbf{A}^m = \mathbf{0} \qquad \text{if and only if} \quad \rho(\mathbf{A}) < 1$$

A.3.9 (Faddeeva, 1959, p. 61)

$$\lim_{m \to \infty} \mathbf{A}^m = \mathbf{0} \qquad \text{if} \quad \|\mathbf{A}\| < 1$$

where $\|\mathbf{A}\|$ is any norm of $\mathbf{A}$.

APPENDIX B

SELECTED COMPUTER PROGRAMS

```
      SUBROUTINE THOMAS(A,B,C,D,X,N)
      DIMENSION A(1),B(1),C(1),D(1),X(1),Q(1001),G(1001)
C
C          SOLUTION OF TRIDIAGONAL SYSTEM OF EQUATIONS
C          A — MAIN DIAGONAL
C          B — SUPERDIAGONAL
C          C — SUBDIAGONAL
C          D — R.H.S. VECTOR
C          X — SOLUTION VECTOR
C          N — ORDER OF MATRIX (NUMBER OF UNKNOWNS)
C     WI=A(1)
      G(1) = D(1)/WI
      DO 1 I=2,N
      Q(I –1) = B(I –1)/WI
      WI = A(I) – C(I)*Q(I – 1)
1     G(I) = (D(I) – C(I)*G(I – 1))/WI
      X(N) = G(N)
      DO 2 I =2,N
      J=N – 1 + 1
2     X(J) = G(J) – Q(J)*X(J + 1)
      RETURN
      END



      SUBROUTINE TANG(A,B,C,D,X,N,ICIRC)
      DIMENSION A(1),B(1),C(1),D(1),X(1),BETA(100),GAMMA(100),CHI(100)
C
C          SOLUTION OF TRIDIAGONAL SYSTEM OF EQUATIONS WITH
           CYCLIC
```

```
C           STRUCTURE
C           A — MAIN DIAGONAL
C           B — SUPERDIAGONAL
C           C — SUBDIAGONAL
C           D — R.H.S.VECTOR
C           X — SOLUTION VECTOR
C           N — ORDER OF MATRIX (NUMBER OF UNKNOWNS)
C           ICIRC — INDICATOR FOR TRIDIAGONAL OR CYCLIC STRUCTURE
C           ICIRC = 0 — STANDARD TRIDIAGONAL FORM (C(1) = 0 AND B(N)
            = 0)
C           ICIRC = 1— CYCLIC STRUCTURE
C
      NM1=N–1
      CHI(1) = 0.
      BETA(1)= –1.
      CHI(2) = D(1)/B(1)
      BETA(2) = A(1)/B(1)
      GAMMA(2) = C(1)/B(1)
      DO 10 I= 2,NM1
      CHI(I + 1) = (D(I) – A(I)*CHI(I) – C(I)*CHI(I – 1))/B(I)
10    BETA(I + 1) = –(A(I)*BETA(I) + C(I)*BETA(I – 1))/B(I)
      IF(ICIRC.EQ.0) GO TO 40
      DO 20 I = 2,NM1
20    GAMMA(I + 1)= – (A(I)*GAMMA(I) + C(I)*GAMMA(I – 1))/B(I)
      AA=CHI(N)/(1. + GAMMA(N))
      BB= BETA(N)/(1. + GAMMA(N))
      UX = A(N) – C(N)*GAMMA(NM1)
      CC = (D(N) – C(N)*CHI(NM1))/UX
      DD = (B(N) – C(N)*BETA(NM1))/UX
      X(1) = (AA – CC)/(BB – DD)
      X(N) = (BB*CC – AA*DD)/(BB – DD)
      DO 30 I =2,NM1
30    X(I) = CHI(I) – BETA(I)*X(1) –GAMMA(I)*X(N)
      RETURN
40    AA = CHI(N)
      BB= BETA(N)
      CC= (D(N) – C(N)*CHI(NM1))/A(N)
      DD = (B(N) – C(N)*BETA(NM1))/A(N)
      X(1) = (AA – CC)/(BB – DD)
```

```
      X(N) = (BB*CC – AA*DD)/(BB – DD)
      DO 50 I=2,NM1
50    X(I) = CHI(I) – BETA(I)*X(1)
      RETURN
      END



      SUBROUTINE INVERT(A)
      DIMENSION A(3,3)
C
C       INVERSION OF 3 BY 3 MATRIX
C       INVERSE IS STORED IN PLACE OF THE ORIGINAL MATRIX
C
C
C       FORWARD ELIMINATION
C
      DO 200 I = 1,2
      IP1 = I + 1
      DO 100 J = IP1,3
      A(J,I) = A(J,I)/A(I,I)
      DO 100 K = IP1,3
100   A(J,K) = A(J,K) – A(J,I)*A(I,K)
200   CONTINUE
C
C             CALCULATE L INVERSE AND U INVERSE
C
      A(3,l) = A(2,1)*A(3,2) – A(3,l)
      A(2,l) = – A(2,l)
      A(3,2) = –A(3,2)
      A(3,3) = 1./A(3,3)
      A(2,3) = – A(2,3)*A(3,3)/A(2,2)
      A(2,2) = 1./A(2,2)
      A(1,3) = – (A(1,2)*A(2,3) + A(1,3)*A(3,3))/A(1,1)
      A(l,2) = – A(1,2)*A(2,2)/A(1,1)
      A(1,1) = 1./A (l,l)
C
C             CALCULATE A INVERSE
C
```

```
      A(1,1) = A (1,1) + A(1,2)*A(2,1) + A(1,3)*A(3,1)
      A(2,1) = A(2,2)*A(2,1) + A(2,3)*A(3,1)
      A(3,1) = A(3,3)*A(3,1)
      A(l,2) = A(1,2) + A(1,3)*A(3,2)
      A(2,2) = A(2,2) + A(2,3)*A(3,2)
      A(3,2) = A(3,2)*A(3,3)
      RETURN
      END



      SUBROUTINE BITRI(A,B,C,D,X,N)
      DIMENSION A(N,4),B(N,4),C(N,4),D(N,2),X(N,2)
      DIMENSION XAMDA(101,4),GAMMA(101,2)
C
C           SOLUTION OF BI-TRIDIAGONAL SYSTEM OF EQUATIONS
C           A — MAIN DIAGONAL
C           B — SUPERDIAGONAL
C           C — SUBDIAGONAL
C           D — R.H.S.VECTOR
C           X — SOLUTION
C           N — NUMBER OF BLOCKS.NUMBER OF UNKNOWNS IS 2*N.
C
C           ALL VECTORS AND MATRICES HAVE BLOCK STRUCTURE WITH 2
            BY 2
C           BLOCKS.THE ORDERING OF ELEMENTS IN BLOCKS IS
C
C                                   1   2           1
C                                         AND
C                                   3   4           2
      DO 10 I= 1,N
      BETA1 = A(I,1)
      BETA2=A(I,2)
      BETA3 = A(I,3)
      BETA4 = A(I,4)
      IF(I.EQ.1) GO TO 20
      BETA 1= BETA 1 – C(I,1)*XAMDA(I – 1,1) – C(1,2)*XAM DA(I – 1,3)
      BETA2 = BETA2 – C(I,1)*XAMDA(I – 1,2) – C(I,2)*XAMDA(I – 1,4)
      BETA3 = BETA3 – C(I,3)*XAMDA(I – 1,1) – C(I,4)*XAMDA(I –1,3)
```

```
      BETA4 = BETA4 – C(I,3)*XAMDA(I – 1,2) – C(1,4)*XAMDA(I – 1,4)
20    XMI = BETA1*BETA4 –BETA2*BETA3
      XMIR= 1./XMI
      DELTA1=D(I,1)
      DELTA2 = D(I,2)
      IF(I.EQ.1) GO TO 30
      DELTA1 = DELTA1 –C(I,1)*GAMMA(I – 1,1)–C(I,2)*GAMMA(I – 1,2)
      DELTA2 = DELTA2 –C(I,3)*GAMMA(I – 1,1)–C(I,4)*GAMMA(I – 1,2)
30    GAMMA(I,1) = (BETA4*DELTA1 – BETA2*DELTA2)*XMIR
      GAMMA(I,2) = (BETA1*DELTA2–BETA3*DELTA1)*XMIR
      IF(I.EQ.N) GO TO 10
      XAMDA(I,1) = (BETA4*B(I,1) – BETA2*B(I,3))*XMIR
      XAMDA(I,2) = (BETA4*B(I,2) – BETA2*B(I,4))*XMIR
      XAMDA(I,3) = (BETA1*B(I,3) – BETA3*B(I,1))*XMIR
      XAMDA(I,4) = (BETA1*B(I,4) – BETA3*B(I,2))*XMIR
10    CONTINUE
      X(N,1) = GAMMA(N,1)
      X(N,2) = GAMMA(N,2)
      NP1 =N + 1
      DO 50 K=2,N
      I=NP1 – K
      X(I,1) = GAMMA(I,1) – XAMDA(I,1)*X(I + 1,1) – XAMDA(I,2)*X(I + 1,2)
      X(I,2) = GAMMA(I,2) – XAMDA(I,3)*X(I + 1, 1) –XAMDA(I,4)*X(I + 1,2)
50    CONTINUE
      RETURN
      END



      SUBROUTINE TRITRI(A,B,C,D,X,N)
      DIMENSION A(N,9),B(N,9),C(N,9),D(N,3),X(N,3)
      DIMENSION XAMDA(101,9),GAMMA(101,3),BETA(9),THETA(9)
C
C          SOLUTION OF TRI-TRIDIAGONAL SYSTEM OF EQUATIONS
C          A — MAIN DIAGONAL
C          B — SUPERDIAGONAL
C          C — SUBDIAGONAL
C          D — R.H.S. VECTOR
C          X — SOLUTION
```

```
C       N — NUMBER OF BLOCKS.TOTAL NUMBER OF UNKNOWNS IS
C       3*N.
C
C
C       ALL VECTORS AND MATRICES HAVE BLOCK STRUCTURE WITH 3
C       BY 3
C       BLOCKS.THE ORDERING OF ELEMENTS IN BLOCKS IS
C
C                                   1   2   3          1
C                                   4   5   6   AND    2
C                                   7   8   9          3
C
      DO 10 I = 1,N
      IM1 = I – 1
      L= – 2
      DO 20 II = 1,3
      L=L + 3
      BETA(L) = A(I,L)
      BETA(L + 1) = A(I,L + 1)
      BETA(L + 2) = A(I,L + 2)
      IF(I.EQ.1) GO TO 20
      BETA(L) = BETA(L) – C(I,L)*XAMDA(IM1, 1) – C(I,L + 1)*XAMDA(IM1,4)
     1       –C(I,L + 2)*XAMDA(IM1,7)
      BETA(L+1)=BETA(L+1)–  C(I,L)*XAMDA(IM1,2)  –  C(I,L  +
      1)*XAMDA(IM1,5)
     1       –C(I,L + 2)*XAMDA(IM1,8)
      BETA(L+2)=BETA(L+2)–  C(I,L)*XAMDA(IM1,3)  –  C(I,L  +
      1)*XAMDA(IM1,6)
     1       –C(I,L + 2)*XAMDA(IM1,9)
20    CONTINUE
      DELTA1 = D(I,1)
      DELTA2 = D(I,2)
      DELTA3 = D(I,3)
      IF(I.EQ.1) GO TO 30
      DELTA1 =DELTA1 – C(I,1)*GAMMA(IM1,1)–C(I,2)*GAMMA(IM1,2)
     1       – C(I,3)*GAMMA(IM1,3)
      DELTA2 = DELTA2 – C(I,4)*GAMMA(IM1,1) – C(I,5)*GAMM A(IM1,2)
     1       – C(I,6)*GAMMA(IM1,3)
      DELTA3 = DELTA3 – C(I,7)*GAMMA(IM1,1) – C(I,8)*GAMMA(IM1,2)
```

```
     1        – C(I,9)*GAMMA(IM1,3)
30    CONTINUE
      THETA(l) = BETA(5)*BETA(9) – BETA(6)*BETA(8)
      THETA(2) = BETA(6)*BETA(7) – BETA(4)*BETA(9)
      THETA(3) = BETA(4)*BETA(8) – BETA(5)*BETA(7)
      THETA(4) = BETA(3)*BETA(8) – BETA(2)*BETA(9)
      THETA(5) = BETA(1)*BETA(9) – BETA(3)*BETA(7)
      THETA(6) = BETA(2)*BETA(7) – BETA(1)*BETA(8)
      THETA(7) = BETA(2)*BETA(6) – BETA(3)*BETA(5)
      THETA(8) = BETA(3)*BETA(4) – BETA(1)*BETA(6)
      THETA(9) = BETA(1)*BETA(5) – BETA(2)*BETA(4)
      XMI = THETA(1)*BETA(1) + THETA(2)*BETA(2) + THETA(3)*BETA(3)
      XMIR= 1./XMI
      GAMMA(I,1)=
      (THETA(1)*DELTA1+THETA(4)*DELTA2+THETA(7)*DELTA3)*XMIR
      GAMMA(I,2)=
      (THETA(2)*DELTA1+THETA(5)*DELTA2+THETA(8)*DELTA3)*XMIR
      GAMMA(I,3)=
      (THETA(3)*DELTA1+THETA(6)*DELTA2+THETA(9)*DELTA3)*XMIR
      IF(I.EQ.N) GO TO 10
      L = – 2
      DO 40 II = 1,3
      L = L + 3
      IP3 = II + 3
      IP6 = II + 6
      XAMDA(I,L) = (THETA(II)*B(I,1) + THETA(IP3)*B(I,4) + THETA(IP6)
     1        *B(I,7))*XMIR
      XAMDA(I,L + 1) = (THETA(II)*B(I,2) + THETA(IP3)*B(I,5) + THETA(IP6)
     1        *B(I,8))*XMIR
      XAMDA(I,L + 2) = (THETA(II)*B(I,3) + THETA(IP3)*B(I,6) + THETA(IP6)
     1        *B(I,9))*XMIR
40    CONTINUE
10    CONTINUE
      X(N,1) = GAMMA(N,1)
      X(N,2) = GAMMA(N,2)
      X(N,3) = GAMMA(N,3)
      NP1=N + 1
      DO 50 K=2,N
      I = NP1 – K
```

```
      IP1=I + 1
      X(I,1) = GAMMA(I,1) – XAMDA(I,1)*X(IP1,1) – XAMDA(I,2)*X(IP1,2)
     1        –XAMDA(I,3)*X(IP1,3)
      X(I,2) = GAMMA(I,2) –XAMDA(I,4)*X(IP1,1) – XAMDA(I,5)*X(IP1,2)
     1        –XAMDA(I,6)*X(IP1,3)
      X(I,3) = GAMMA(I,3) – XAMDA(I,7)*X(IP1,1) – XAMDA(I,8)*X(IP1,2)
     1        –XAMDA(I,9)*X(IP1,3)
50    CONTINUE
      RETURN
      END



      SUBROUTINE GBAND(A,D,X,N,M,EPS,IERR,IFRST)
      DIMENSION A(1),D(1),X(1)
C
C         SOLUTION OF SYSTEM OF EQUATIONS WITH BAND MATRIX BY
C         STANDARD ELIMINATION WITHOUT PIVOTING
C
C         A – ONE–DIMENSIONAL ARRAY CONTAINING THE BAND OF THE
            MATRIX
C         STORED BY ROWS. THE REQUIRED DIMENSION OF A IS N*(2*M +
            1) – M*M – M
C         M – NUMBER OF DIAGONALS ABOVE THE MAIN
            DIAGONAL.NUMBER OF
C         DIAGONALS BELOW THE MAIN DIAGONAL IS ALSO
            M,THEREFORE THE
C         TOTAL BANDWIDTH IS 2*M + 1
C         N – NUMBER OF EQUATIONS
C         D – R.H.S. VECTOR
C         X – SOLUTION VECTOR
C         EPS – THE ELEMENT ON THE MAIN DIAGONAL IS COMPARED TO
            EPS
C         DURING ELIMINATION. IF IT IS SMALLER, VALUE OF THE
            COUNTER ERR IS INCREMENTED.
C         IERR – NUMBER OF TIMES THE ELEMENT ON THE MAIN
            DIAGONAL WAS
C         SMALLER THAN EPS
C         IFRST – IF IFRST = 0, MATRIX IS INVERTED AND THE INVERSE IS
```

```
C           STORED IN PLACE OF THE ORIGINAL MATRIX. IF IFRST.GT.0,
C           ROUTINE ASSUMES THAT THE MATRIX HAS BEEN INVERTED
            AND WILL
C           CALCULATE SOLUTION CORRESPONDING TO THE NEW R.H.S.
            VECTOR.
C           THIS OPTION IS USEFUL WHEN SOLVING REPEATEDLY THE SAME
C           EQUATIONS WITH DIFFERENT R.H.S. VECTORS.
C
      IERR = 0
      J = 1
      DO 10 I= 1,N
      IE= M
      IF(I + M – N) 21,21,22
22    IE=N–I
21    IEAUX = M
      IF (I–M) 23,23,24
23    IEAUX = I
24    IE1 = IE + IEAUX
      MBIG = IE
      J1=J+IE1
      J2=J1
      IF(IFRST.GT.0) GO TO 27
      IF(ABS(A(J))—EPS) 25,25,27
25    IERR = IERR + 1
27    IF(MBIG) 10,10,26
26    DO 20 JO = l,MBIG
      S = A(J1)/A(J)
      IF(IFRST.GT.0) GO TO 35
      DO 30 K=1,MBIG
      J1K = J1+K
      JK = J + K
30    A(J 1K) = A(J 1K) – A(JK)*S
35    CONTINUE
      IAUX=JO+I
      D(IAUX) = D(IAUX) – D(I)*S
      IE= M
      IF(IAUX+M – N) 31,31,32
32    IE = N – IAUX
31    IEAUX = M
```

```
      IF(IAUX – M) 33,33,34
33    IEAUX = IAUX
34    IE1 = IE +IEAUX
20    J1 = J1 + IE1
10    J = J2 + 1
      J = J – M – 1
      NP1 = N + 1
      DO 40 IINV = 1,N
      1= NP1 – IINV
      IE = M
      IF(I + M – N) 41,41,42
42    IE = N – I
41    MBIG = IE
      X(I) = D(I)
      IF(MBIG) 44,44,43
43    DO 50 K = 1,MB1G
      1K= I + K
      JK = J + K
50    X(I)=X(I) – X(IK)*A(JK)
44    X(I)=X(I)/A(J)
      IE= M
      IF(I + M – NP1) 51,51,52
52    IE = NP1 – I
51    1EAUX = M
      IF(I – 1 – M) 53,53,54
53    IEAUX = I – 1
54    IE1 = IE + IEAUX
      J=J – IE1 – 1
40    CONTINUE
      RETURN
      END



      SUBROUTINE TABSEQ(X,Y,N,XX,YY)
      DIMENSION X(1),Y(1)
C     TABLE LOOK-UP USING SEQUENTIAL SEARCH
C     LINEAR INTERPOLATION BETWEEN TABLE VALUES USED
C
```

```
C       X – VECTOR OF INDEPENDENT VALUES (ARGUMENTS)
C       Y – VECTOR OF DEPENDENT VARIABLES (FUNCTION VALUES)
C       N – NUMBER OF TABLE ENTRIES
C       XX – ARGUMENT
C       YY – INTERPOLATED FUNCTION OF ARGUMENT XX
        IF(XX.LT.X(1)) GO TO 99
        I = 1
  100   I = I+1
        IF(I.GT.N) GO TO 98
        IF(XX.GT.X(I)) GO TO 100
        YY = Y(I – 1) + (Y(I) – Y(I – 1))*(XX – X(I – 1))/(X(I) – X(I – 1))
        RETURN
   99   YY = Y(1)
        WRITE(6,89)XX
   89   FORMAT(1H ,38HWARNING – ARGUMENT OUT OF TABLE ,XX = ,F12.5)
        RETURN
   98   YY = Y(N)
        WRITE(6,89) XX
        RETURN
        END



        SUBROUTINE TABBIN(X,Y,NXX,YY)
        DIMENSION X(1),Y(1)
C       TABLE LOOK-UP USING BINARY SEARCH
C       LINEAR INTERPOLATION BETWEEN TABLE VALUES USED
C
C       X- VECTOR OF INDEPENDENT VALUES (ARGUMENTS)
C       Y- VECTOR OF DEPENDENT VARIABLES (FUNCTION VALUES)
C       N- NUMBER OF TABLE ENTRIES
C       XX- ARGUMENT
C       YY- INTERPOLATED FUNCTION OF ARGUMENT XX
        IF(XX.LT.X(1)) GO TO 99
        IF(XX.GT.X(N))GO TO 98
        N1 = 1
        N2 = N
  100   NDIF = N2 – N1
        IF(NDIF.EQ.1) GO TO 20
```

```
    NH = NDIF/2 + N1
    IF(XX.LT.X(NH)) GO TO 10
    N1 = NH
    GO TO 100
10  N2 = NH
    GO TO 100
20  YY = Y(N1) + (Y(N2) – Y(N1))*(XX – X(N 1))/(X(N2) – X(N 1))
    RETURN
99  YY = Y(l)
    WRITE(6,89)XX
    RETURN
89  FORMAT(1H ,38HWARNING — ARGUMENT OUT OF TABLE ,XX =
,F12.5)
98  YY = Y(N)
    WRITE(6,89) XX
    RETURN
    END
```

BIBLIOGRAPHY

Ahlberg, J. H., Nilson, E. N. and Walsh, J. L. (1967). *The Theory of Splines and their Applications*, Academic Press, New York.

Aird, T. J., Battiste, E. L. and Gregory, W. C. (1977). Portability of mathematical software coded in FORTRAN, *ACM Trans. Math. Software* , **3,** No. 2, pp. 113–27.

Akbar, A. M., Arnold, M. D. and Harvey, A. H. (1974). Numerical simulation of individual wells in a field simulation model, *Soc. Petrol. Eng. J.*, **14,** No. 4, pp. 315–20.

Al-Hussainy, R., Ramey, H. J., Jr. and Crawford, P. B. (1966). The flow of real gases through porous media, *Trans. SPE of AIME* , **18,** pp. 624–36 (JPT).

Ames, W. F. (1969). *Numerical Methods for Partial Deferential Equations*, Thomas Nelson & Sons, Don Mills.

Arakawa, A. (1966). Computational design for long-term numerical integration of the equations of fluid motion: two-dimensional incompressible flow. Part 1, *J. Compt. Phys*., **1,** pp. 119–43.

Archer, J. S. and Wong, S. W. (1973). Use of a reservoir simulator to interpret laboratory waterflood data, *Soc. Petrol. Eng. J*., **13,** No. 6, pp. 343–7.

Aris, R. (1962). *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*, Prentice-Hall, New Jersey.

Aronofsky, J. S. and Jenkins, R. (1954). A simplified analysis of unsteady radial gas flow, *Trans. SPE of AIME* , **201,** pp. 149–54.

Asfari, A. and Witherspoon, P. A. (1973). Numerical simulation of naturally fractured reservoirs, *SPE* 4290, *3rd Symposium on Numerical Simulation of Reservoir Performance*, Houston.

Ashford, F. E. (1969). Computed relative permeability drainage and imbibition, *SPE* 2582, 44*th Annual Meeting*, Denver.

Aziz, K. (1968). Numerical methods for three dimensional reservoir models, *J. Can. Petrol. Technol*., **7,** No. 1, pp. 41–6.

Aziz, K. and Settari, A. (1972). A new iterative method for solving reservoir simulation equations, *J. Can. Petrol. Technol*., **11**, No. 1, pp. 62–8.

Bakes, M. D. (1965). An alternative method of solution of certain tridiagonal systems of linear equations, *Compt. J*., 7, pp. 135–6.

Bank, R. E. (1976). Marching algorithms and block Gaussian elimination, in: *Sparse Matrix Computations*, ed. J. R. Bunch and D. J. Rose, Academic Press, New York, pp. 292–308.

Barakat, J. Z. and Clark, J. A. (1966). On the solution of the diffusion equations by numerical methods, *J. Heat Transfer*, **88**, pp. 421–7.

Bear, J. (1972). *Dynamics of Fluids in Porous Media*, American Elsevier, New York.

Bellman, R. (1960). *Introduction to Matrix Analysis*, McGraw-Hill, New York.

Bird, R. B., Stewart, W. E. and Lightfoot, E. N. (1960). *Transport Phenomena*, J. Wiley & Sons, New York.

Bishop, K. A., Brett, V. S., Green, D. W. and McElhiney, J. E. (1976). The application of sensitivity analysis to reservoir simulation, *SPE* 6102, 515*st Annual Fall Meeting*, New Orleans.

Björdammen, J. B. and Coats, K. H. (1969). Comparison of alternating direction and successive overrelaxation techniques in simulation of reservoir fluid flow, *Soc. Petrol. Eng. J.*, **9**, No. 1, pp. 47–58.

Blades, D. N. and Stright, D. H., Jr. (1975). Predicting high volume lift performance in wells coning water, *J. Can. Petrol. Technol.*, **14**, No. 4, pp. 62–70.

Blair, P. M. and Weinaug, C. F. (1969). Solution of two-phase flow problems using implicit difference equations, *Trans. SPE of AIME*, **246**, pp. 417–24 (SPEJ).

Blum, E. K. (1972). *Numerical Analysis and Computation: Theory and Practice,* Addison-Wesley, Mass.

Bondor, P. L., Hirasaki, G. J. and Tham, M. J. (1972). Mathematical simulation of polymer flooding in complex reservoirs, *Soc. Petrol. Eng. J.*, **12**, No. 5, pp. 369–82.

Botset, H. G. (1946). Electrolytic model and its applications to study of recovery problems, *Trans. SPE of AIME*, 165, pp. 15–25.

Bournazel, C. and Jeanson, B. (1971). Fast water-coning evaluation method, *SPE* 3628, 46*th Annual Fall Meeting*, New Orleans.

Brandon, D. M., Jr. (1974). The implementation and use of sparse matrix techniques in general simulation programs, *Compt. J.,* **17**, pp. 165–71.

Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems, *Math. of Compt.*, **31**, No. 138, pp. 333–90.

Brayton, R. K., Gustavson, F. G. and Willoughby, R. A. (1970). Some results on sparse matrices, *Math. of Compt.*, **24**, No. 112, pp. 937–54.

Breitenbach, E. A., Thurnau, D. H. and Van Poolen, H. K. (1969). The fluid flow simulation equations, *Soc. Petrol. Eng. J.*, **9**, No. 2, pp. 155–69.

Brian, P. L. T. (1961). A finite difference method of high order accuracy foi the solution of three-dimensional transient heat conduction problems, *A.I.Ch.E. J.*, **7**, No. 3, pp. 367–70.

Briggs, J. E. and Dixon, T. N. (1968). Some practical considerations in the numerical solution of two dimensional reservoir problems, *Trans. SPE of AIME*, **243**, pp. 185–94.

Browning, G., Kreiss, H. O. and Oliger, J. (1973). Mesh refinement, *Math. of Compt.*, **27**, No. 121, pp. 29–40.

Bruce, W. A. (1943). An electrical device for analyzing oil-reservoir behaviour, *Trans. SPE of AIME*, **151**, pp. 112–24.

Bruce, G. H., Peaceman, D. W., Rachford, H. H., Jr. and Rice, J. D. (1953). Calculations of unsteady-state gas flow through porous media, *Trans. SPE of AIME*, **198**, pp. 79–92 (JPT).

Bunch, J. R. and Rose, D. J. (eds.) (1976). *Sparse Matrix Computations*, Academic Press, New York.

Buzbee, B. L., Golub, G. H. and Nielson, C. W. (1970). On direct methods for solving Poisson's equations, *SIAM J. Numer. Anal.*, **7**, No. 4, pp. 627–56.

Byrne, G. D. and Hindmarsh, A. C. (1975). Polyalgorithm for the numerical solution of ordinary differential equations, *ACM Trans. on Mathematical Software*, **1**, No. 1, pp. 71–96.

Calahan, D. A. (1968). A stable accurate method of numerical integration for non linear systems, *IEEE Proc.*, **56,** pp. 744–5.

Calhoun, J. C., Jr., Lewis, M., Jr. and Newman, R. C. (1949). Experiments on the capillary properties of porous solids, *Trans. SPE of AIME*, **186,** pp. 180–96.

Cardenas, A. F. (1973). The PDEL language and its use to solve field problems—user's manual, *University of California*, Los Angeles.

Cardwell, W. T., Jr. (1959). The meaning of the triple value in non-capillary Buckley-Leverett theory, *Trans. SPE of AIME*, **216,** pp. 271–6.

Carnahan, B., Luther, H. A. and Wilkes, J. O. (1969). *Applied Numerical Methods*, J. Wiley & Sons, New York.

Carre, B. D. (1961). The determination of the optimum accelerating factor for successive overrelaxation, *Compt. J.*, **4,** pp. 73–8.

Carslaw, H. S. and Jaeger, J. C. (1959). *Conduction of Heat in Solids*, 2nd edn., Clarendon Press, Oxford.

Carter, R. D. and Tracy, C. W. (1960). An improved method for calculating water influx, *Trans. SPE of AIME*, **219,** pp. 415–17 (JPT).

Carter, R. D. (1966). Performance predictions for gas reservoirs considering two dimensional unsteady state flow, *Trans. SPE of AIME*, **237,** pp. 35–43 (SPEJ).

Carter, R. D., Kemp, L. F., Jr., Pierce, A. C. and Williams, D. L. (1974). Performance matching with constraints, *Trans. SPE of AIME*, **257,** pp. 187–96 (SPEJ).

Carver, M. B. (1973). A FORTRAN oriented simulation system for the general solution of sets of implicitly coupled partial differential equations, *Proceedings* 1974 *Summer Computer Simulation Conference*, Houston, pp. 104–12.

Chain, R., Mounford, C. J., Raghavan, R. and Thomas, G. W. (1976). Determination of reservoir properties from back pressure tests with applications to reservoir simulation, *J. Petrol. Technol.,* **28,** pp. 603–10.

Chappelear, J. E. and Rogers, W. (1974). Some practical considerations in the construction of a semi-implicit simulator, *Soc. Petrol. Eng. J.*, **14,** No. 3, pp. 216–20.

Chappelear, J. E. and Hirasaki, G. J. (1976). A model of oil-water coning for 2-D areal reservoir simulation, *Trans. SPE of AIME*, **261**, pp. 65–72 (SPEJ).

Chavent, G., Dupuy, M. and Lemonnier, P. (1975). History matching by use of optimal control theory, *Trans. SPE of AIME*, **259,** pp. 74–86 (SPEJ).

Chen, W. H., Gavalas, G. R., Seinfeld, J. H. and Wasserman, M. L. (1974). A new algorithm for automatic history matching, *Trans. SPE of AIME*, **257,** pp. 593–608 (SPEJ).

Ciment, M. (1971). Stable difference schemes with uneven mesh spacings, *Math. of Compt.*, **25,** No. 114, pp. 219–27.

Ciment, M. and Sweet, R. A. (1973). Mesh refinements for parabolic equations, *J. Compt. Phys.*, **12,** pp. 513–25.

Coats, K. H. and Terhune, M. H. (1966). Comparison of alternating direction explicit and implicit procedures in two dimensional flow calculations, *Trans. SPE of AIME*, **237,** pp. 350–62 (SPEJ).

Coats, K. H., Nielsen, R. L., Terhune, M. H. and Weber, A. G. (1967). Simulation of three-dimensional, two-phase flow in oil and gas reservoirs, *Trans. SPE of AIME*, **240,** pp. 377–88 (SPEJ).

Coats, K. H. (1968). *Elements of Reservoir Simulation*, Lecture Notes, University of Texas, reprinted by Intercomp Resources Development and Engineering Inc., Houston.

Coats, K. H. (1968a). An analysis for simulating reservoir performance under pressure maintenance by gas and/or water injection, *Soc. Petrol. Eng. J.*, **8,** No. 4, pp. 331–40.

Coats, K. H. (1968b). A treatment of the gas percolation problem in simulation of three-dimensional three-phase flow in reservoirs, *Trans. SPE of AIME*, **243,** pp. 413–19

(SPEJ).

Coats, K. H. (1969). Use and misuse of reservoir simulation models, *J. Petrol. Technol*., **21,** pp. 1391–8.

Coats, K. H., Dempsey, J. R. and Henderson, J. H. (1970). A new technique for determining reservoir description from field performance data, *Trans. SPE of AIME*, **249,** pp. 66–74 (SPEJ).

Coats, K. H., Dempsey, J. R., Ancell, K. L. and Gibbs, D. E. (1971). Analysis and prediction of gas well performance, *SPE* 3474, 46*th Annual Fall Meeting*, New Orleans.

Coats, K. H., Dempsey, J. R. and Henderson, J. H. (1971a). The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir performance, *Trans. SPE of AIME*, **251,** pp. 63–71 (SPEJ).

Coats, K. H., George, W. D. and Marcum, B. E. (1974). Three-dimensional simulation of steamflooding, *Trans. SPE of AIME*, **257**, pp. 573–92 (SPEJ).

Coats, K. H. (1976). Simulation of steamflooding with distillation and solution gas, *Soc. Petrol Eng. J*., **16,** No. 5, pp. 235–47.

Coats, K. H. (1976a). A fully implicit steamflood model, *SPE* 6105, 51*st Annual Fall Meeting*, New Orleans.

Coats, K. H. (1976b). Private communication.

Cody, W. J. (1974). Construction of numerical subroutine libraries, *SIAM Rev*., **16,** No. 1, pp. 36–46.

Collatz, L. (1966). *The Numerical Treatment of Differential Equations*, 3rd edn., Springer-Verlag, Berlin.

Collins, R. E. (1961). *Flow of Fluids Through Porous Materials*, Van Nostrand Reinhold, New York.

Cook, R. E., Jacoby, R. H. and Ramesh, A. B. (1974). A beta-type reservoir simulator for approximating compositional effects during gas injection, *Soc. Petrol. Eng. J*., **14,** No. 5, pp. 471–81.

Corey, A. T., Rathjens, C. H., Henderson, J. H. and Wyllie, M. R. J. (1956). Three-phase relative permeability, *Trans. SPE of AIME*, **207,** pp. 349–51 (JPT).

Cornell, D. and Katz, D. L. (1953). Pressure gradients in natural gas reservoirs, *Trans. SPE of AIME*, **198,** pp. 61–71 (JPT).

Cottin, R. H. (1971). Optimization of the completion and exploitation of a well under two phase coning, *Rev. L'Inst. Franc. Petrol*, **26,** pp. 171–97 (in French).

Courant, R. and Hilbert, D. (1953). *Methods of Mathematical Physics*, Vol. 1 and 2, Interscience Publishers, New York.

Craft, B. C. and Hawkins, M. F. (1959). *Applied Petroleum Reservoir Engineering*, Prentice-Hall, New Jersey.

Craig, F. F., Jr. (1971). *The Reservoir Engineering Aspects of Waterflooding*, SPE Monograph Series, **3,** SPE–AIME, Dallas.

Crandall, S. H. (1955). An optimum implicit recurrence formula for the heat conduction equation, *Quart. Appl. Math*., **13,** pp. 318–20.

Culham, W. E. and Varga, R. S. (1971). Numerical methods for time-dependent nonlinear boundary value problems, *Soc. Petrol. Eng. J*., **11,** No. 4, pp. 374–88.

Cuthill, E. H. and Varga, R. S. (1959). A method of normalized block iteration, *J. Assoc. Compt. Mach*., **6,** pp. 236–44.

Cuthill, E. (1972). Several strategies for reducing the band-width of matrices, in: *Sparse Matrices, and Their Applications*, eds. D. J. Rose and R. A. Willoughby, Plenum Press, New York, pp. 157–66.

Cutler, J. M. and Rees, W. A. (1970). A study of water coning in the oil creek reservoir, North Antioch Field, Oklahoma, *SPE* 2814, *2nd Symposium on Numerical Simulation of Reservoir Performance*, Dallas.

Dandona, A. K. and Morse, R. A. (1973). How flooding rate and gas saturation affect waterflood performance, *Oil Gas J*., **71,** No. 27, pp. 60–4; and Vol. **71,** No. 28, pp. 69–73.

Dandona, A. K. and Morse, R. A. (1975). Down-dip waterflooding of an oil reservoir having a gas cap, *Trans. SPE of AIME*, **259,** pp. 1005–16 (JPT).

Danilov, D. L., Konovalov, A. N. and Yakuba, S. I. (1968). Equations and boundary value problems in the theory of two phase flows through porous media, *Dokl. Akad. Nauk. USSR*, **183,** No. 2, pp. 307–10 (in Russian).

Darcy, H (1856). Les Fontaines publiques de la Ville de Dijon, Dalmount, Paris (reprinted in Hubbert, 1969).

DesBrisay, C. L., Gray, J. W. and Spivak, A. (1975). Miscible flood performance of the Intisar 'D' Field, Libyan Arab Republic, *J. Can. Petrol. Technol*., pp. 935–43.

Dietrich, J. K. and Bondor, P. L. (1976). Three-phase oil relative permeability models, *SPE* 6044, 515*st Annual Fall Meeting*, New Orleans.

Douglas, J., Jr. (1955). On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods, *J. SIAM*, **3,** No. 1, pp. 42–65.

Douglas, J., Jr and Rachford, H. H., Jr. (1956). On the numerical solution of heat conduction problems in two and three space variables, *Amer. Math. Soc. Trans*., **82,** No. 2, pp. 421–39.

Douglas, J., Jr., Peaceman, D. W. and Rachford, H. H., Jr. (1959). A method for calculating multi-dimensional immiscible displacement, *Trans. SPE of AIME*, **216,** pp. 297–306.

Douglas, J., Jr., (1960). A numerical method for the solution of a parabolic system, *Numer. Math*., **2,** pp. 91–8.

Douglas, J., Jr., (1961). A survey of numerical methods for parabolic differential equations, in: *Advances in Computers*, **2,** pp. 1–54, ed. F. L. Alt, Academic Press, New York.

Douglas, J., Jr. (1962). Alternating direction methods for three space variables, *Numer. Math*., **4,** pp. 41–63.

Douglas, J., Jr. and Jones, B. F., Jr. (1963). On predictor–corrector methods for nonlinear parabolic differential equations, *J. SIAM*, **11,** No. 1, pp. 195–204.

DuFort, E. C. and Frankel, S. P. (1953) Stability conditions in the numerical treatment of parabolic differential equations, *Math. Tables Aids Compt*., **7,** pp. 135–52.

Dupont, T., Kendall, R. P. and Rachford, H. H., Jr. (1968). An approximate factorization procedure for solving self-adjoint elliptic difference equations, *SIAM J. Numer. Anal*., **5,** No. 3, pp. 559–73.

Dupuy, M., Jain, C. and Besset, P. (1971). Automatic adjustment possibilities for reservoir model in single-period production, *Rev. L'Inst. Franc. Petrol*., **26,** No. 4, pp. 269–300 (in French).

Earlougher, R. C., Jr. (1977). *Advances in Well Test Analysis*, SPE Monograph Series, **5.**

Ehle, B. L. (1973). A-stable methods and Padé approximations to the exponential, *SIAM J. Math. Anal*., **4,** No. 4, pp. 671–80.

Eisenstat, S. C., Schultz, M. H. and Sherman, A. H. (1976). Considerations in the design of software for sparse Gaussian elimination, in: *Sparse Matrix Computations*, eds. J. R. Bunch and D. J. Rose, Academic Press, New York, pp. 263–73.

Elenbaas, J. R. and Katz, D. L. (1947). A radial turbulent flow formula, *Trans. SPE of AIME*, **174,** pp. 25–40.

Emmanuel, A. E. and Cook, G. W. (1974). Pseudo-relative permeability for well modeling, *Soc. Petrol. Eng. J*., **14,** No. 1, pp. 7–9.

Engeli, M., Ginsburg, T., Ruthishauser, H. and Stiefel, E. (1959). Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems, *Mitteilungen aus dem Institut für Angewandte Mathematik*, Nr. 8, Birkhauser-Verlag, Basel/Stuttgart, **107,** pp. 206–7.

ERCB (1975). *The Theory and Practice of the Testing of Gas Wells*, 3rd edn., Energy Resources Conservation Board, Alberta, Canada.

Evans, D. J. and Forrington, C. V. D. (1963). Notes on the solution of certain tridiagonal systems of linear equations, *Compt. J.*, **5,** pp. 327–8.

Evans, D. J. (1971). The numerical solution of the fourth boundary value problem for parabolic partial differential equations, *J. Inst. Math. Appl.*, **7,** pp. 61–75.

Evans, D. J. (1971*a*). An algorithm for the solution of symmetric general three term linear systems, *Compt. J.*, **14,** No. 4, pp. 444–5.

Evrenos, A. I. and Comer, A. G. (1969). Sensitivity studies of gas–water relative permeability and capillarity in reservoir modeling, *SPE* 2608, 44*th Annual Fall Meeting*, Denver.

Evenros, A. I. and Comer, A. G. (1969a). Numerical simulation of hysteretic flow in porous media, *SPE* 2693, 44*th Annual Fall Meeting*, Denver.

Faddeev, D. K; and Faddeeva, V. N. (1963). *Computational Methods of Linear Algebra*, translated by R. C. Williams, W. H. Freeman, San Francisco.

Faddeeva, V. N. (1959). *Computational Methods of Linear Algebra*, Dover Publications, New York.

Fancher, G. H. and Lewis, J. A. (1933). Flow of simple fluids through porous materials, *Ind. Eng. Chem. Fund.*, **24,** pp. 1139–47.

Fay, C. H. and Prats, M. (1951). The application of numerical methods to cycling and flooding problems, *Proc. 3rd World Petroleum Congress*, Section II, pp. 555–64.

Fayers, F. J. and Sheldon, J. W. (1959). The effect of capillary pressure and gravity on two-phase fluid flow in a porous medium, *Trans. SPE of AI ME*, **216,** pp. 147–55.

Fernbach, S. (1973). Class V computers to challenge mathematicians, *SIAM News*, **6,** No. 6.

Fetkovich, M. J. (1971). A simplified approach to water influx calculations—finite aquifer systems, *J. Petrol. Technol.*, **23,** pp. 814–28.

Finol, A. and FarouqAli, S. M. (1975). Numerical simulation of oil production with simultaneous ground subsidence, *Trans. SPE of AIME*, **259,** pp. 411–24 (SPEJ).

Forester, C. K. (1977). Higher order monotonic convective difference schemes, *J. Compt. Phys.*, **23,** No. 1, pp. 1–22.

Forschheimer, P. (1901). Wasserbewegung durch Boden, *Z. Ver. Deut. Ing.*, **45,** pp. 1782–8 (in German).

Forsythe, G. E. and Wasow, W. (1960). *Finite-Difference Methods for Partial Differential Equations*, J. Wiley & Sons, New York.

Fox, L. (1964). *An Introduction to Numerical Linear Algebra*, Clarendon Press, Oxford.

Fox, D. G. and Deardorff, J. W. (1972). Computer methods for simulation of multidimensional, nonlinear, subsonic, incompressible flow, *J. Heat Transfer*, **94,** pp. 337–46.

Frick, T. C. and Taylor, R. W. (1962). *Petroleum Production Handbook*, Vol. II, *Reservoir Engineering*, SPE–AIME, Dallas.

Fussell, D. D. and Yanosik, J. L. (1976). An iterative sequence for phase equilibrium calculations incorporating the Redlich–Kwong equation of state, *SPE* 6050, 51*st Annual Fall Technical Conference and Exhibition*, New Orleans.

Fussell, L. T. and Fussell, D. D. (1977). A new mathematical technique for compositional models incorporating the Redlich–Kwong equation of state, *SPE* 6891, 52*nd Annual Technical Conference and Exhibition*, Denver.

Gavalas, G. R., Shah, P. C. and Seinfeld, J. H. (1976). Reservoir history matching by Bayesian estimation, *Trans. SPE of AIME*, **261,** pp. 337–50 (SPEJ).

Gear, C. W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey.

Geertsma, J., Craes, G. A. and Schwarz, N. (1956). Theory of dimensionally scaled models of petroleum reservoirs, *Trans. SPE of AIME*, **207,** pp. 118–27 (JPT).

Geertsma, J. (1974). Estimating the coefficient of inertial resistance in fluid flow through porous media, *Soc. Petrol. Eng. J.*, **14,** No. 5, pp. 445–50.

George, A. (1973). Nested dissection of a regular finite element mesh, *SIAM J. Numer. Anal.*, **10,** No. 2, pp. 345–63.

George, A. and Liu, W. H. (1975). A note on fill for sparse matrices, *SIAM J. Numer. Anal.*, **12,** No. 3, pp. 452–5.

Girault, V. (1974). Theory of a finite difference method on irregular networks, *SIAM J. Numer. Anal.*, **11,** No. 2, pp. 260–82.

Goult, R. J., Hoskins, R. F., Milner, J. A. and Pratt, M. J. (1974). *Computational Methods in Linear Algebra*, Stanley Thornes (Publishers), London.

Gourlay, A. R. (1970). Hopscotch: A fast second order partial differential equation solver, *J. Inst. Math. Appl.*, **6,** pp. 375–90.

Gourlay, A. R. and McGuire, G. R. (1971). General hopscotch algorithm for the numerical solution of partial differential equations, *J. Inst. Math. Appl.*, **7,** pp. 216–27.

Govier, G. W. (1961). Interpretation of the results of the back pressure testing of gas wells, *CIM Trans.*, **64**, pp. 511–14.

Govier, G. W. and Aziz, K. (1972). *The Flow of Complex Mixtures in Pipes*, Van Nostrand Reinhold, New York.

Greenspan, D. (1965). *Introductory Numerical Analysis of Elliptic Boundary Value Problems*, Harper and Row, New York.

Gruska, J. and Poliak, P. (1967). Riešienie systémov linearných rovníc s pásovými maticami, *Aplikace Matematiky*, **12,** No. 6, pp. 439–98 (in Czech).

Gustavson, F. G. and Hachtel, G. D. (1973). Program implementation of a network design program based on the sparse tableau adjoint method, *Proceedings of the* 16*th Midwest Symposium on Circuit Theory*, **1,** University of Waterloo, Ontario.

Gustavson, F. (1975). Finding the block lower triangular form of a sparse matrix, *IBM Research* RC 5667, (No. 24420), *Computer Science Mathematics.*

Hageman, L. A. and Kellogg, R. B. (1968). Estimating optimum overrelaxation parameters, *Math. of Compt*., **22,** No. 101, pp. 60–8.

Harary, F. (1971). Sparse matrices and graph theory, in: *Large Sparse Sets of Linear Equations*, ed. J. K. Reid, Academic Press, London, pp. 139–50.

Harlow, F. H. and Amsden, A. A. (1971). *Fluid Dynamics*, Report No. LA 4700, Los Alamos Scientific Laboratory, Los Alamos, New Mexico.

Harris, D. G. (1975). The role of geology in reservoir simulation studies, *J. Petrol. Technol*., **27,** pp. 625–32.

Hauber, W. C. (1964). Prediction of waterflood performance for arbitrary well patterns and mobility ratios, *Trans. SPE of AIME*, **231,** pp. 95–103 (JPT).

Hawthorne, R. G. (1974). A model of fluid displacement in multilayer porous media including capillary pressure effects between layers, *SPE* 5126, 49*th Annual Fall Meeting*, Houston.

Hearn, C. L. (1971). Simulation of stratified waterflooding by pseudo relative permeability curves, *J. Petrol. Technol*., **23,** pp. 805–13.

Henrici, P. (1962). *Discrete Variable Methods in Ordinary Differential Equations*, J. Wiley & Sons, New York.

Henry, R. L. (1976). Effect of including composition-dependent *K* values to generate data for a modified beta-type simulator, *J. Petrol. Technol*., **28,** pp. 1180–2.

Hiatt, W. N. (1958). Injected fluid coverage of multi-well reservoirs with permeability stratification, *Drilling and Production Practice, API*, pp. 165–94.

Hiatt, W. N. (1968). Mathematical basis of two-phase, incompressible, vertical flow through porous media and its implications in the study of gravity-drainage-type petroleum reservoirs, *Trans. SPE of AIME*, **243,** pp. 225–30 (SPEJ).

Higgins, R. V. and Leighton, A. J. (1962). A computer method to calculate two-phase flow in irregularly bounded porous medium, *Trans. SPE of AIME*, **225,** pp. 679–83 (JPT).

Higgins, R. V. and Leighton, A. J. (1962a). Computer prediction of water drive of oil and gas mixtures through irregularly bonded porous media—Three-phase flow, *Trans. SPE of AIME*, **225,** pp. 1048–54 (JPT).

Hildebrand, F. B. (1956). *Introduction to Numerical Analysis*, McGraw-Hill, New York.

Hildebrand, F. B. (1965). *Methods of Applied Mathematics*, 2nd edn., Prentice-Hall, New Jersey.

Hirasaki, G. J. and O'Dell, P. M. (1970). Representation of reservoir geometry for numerical simulation, *Trans. SPE of AIME*, **249,** pp. 393–404 (SPEJ).

Hirt, C. W. (1968). Heuristic stability theory for finite-difference equations, *J. Compt. Phys*., **2,** pp. 339–55.

Hirt, C. W., Amsden, A. A. and Cook, J. L. (1974). An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *J. Compt. Phys*., **14,** pp. 227–53.

Holloway, C. C., Thomas, L. K. and Pierson, R. G. (1975). Reduction of grid orientation effects in reservoir simulation, *SPE* 5522, 50*th Annual Fall Meeting*, Dallas.

Houpeurt, A. (1959). On the flow of gases in porous media, *Rev. L'Inst. Franc. Petrol*., **14,** No. 11, pp. 1468–97; No. 12, pp. 1637–84.

Hubbert, M. K. (1940). The theory of ground-water motion, *J. Geol*., **48,** No. 8, part 1, pp. 785–944 (reprinted with corrections in Hubbert, 1969).

Hubbert, M. K. (1956). Darcy's Law and the field equations of the flow of underground fluids, *Trans. SPE of AIME*, **207,** pp. 222–39 (JPT), (reprinted in Hubbert, 1969).

Hubbert, M. K. (1969). *The Theory of Ground-Water Motion and Related Papers*, Hafner Publishing, New York.

Hurst, W. (1953). Determination of performance curve in five-spot waterflood, *Petrol. Eng*., **25,** pp. B40–B48.

Hurst, W. (1958). Simplifications of the material balance formulas by the Laplace transformation, *Trans. SPE of AIME*, **213,** pp. 292–303.

Jacks, H. H., Smith, O. J. E. and Mattax, C. C. (1973). The modelling of a three-dimensional reservoir with a two-dimensional reservoir simulator—The use of dynamic pseudo functions, *Soc. Petrol. Eng. J*., **13,** No. 3, pp. 175–85.

Jennings, A. (1971). Solution of variable bandwidth positive definite simultaneous equations, *Compt. J*., **14,** No. 4, p. 446.

Kafka, J. (1968). Nets with subdivided meshes in the numerical solution of Dirichlet's Problem, *Zeitschrift für Angewandte Mathematik und Mechanik*, **48,** Heft 8 und Sonderheft, S. T79.

Karplus, W. J. (1956). Water coning before breakthrough—an electronic analog treatment, *Trans. SPE of AIME*, **207,** pp. 240–5 (JPT).

Karplus, W. J. (1958). *Analog Simulation*, McGraw-Hill, New York.

Katz, D. L., Cornell, D., Kobayashi, R., Poettman, F. H., Vary, J. A., Elenbaas, J. R. and Weinaug, C. F. (1959). *Handbook on Natural Gas Engineering*, McGraw-Hill, New York.

Katz, D. L., Tek, M. R., Coats, K. H., Katz, M. L., Jones, S. C. and Miller, M. C. (1963). Movement of underground water in contact with natural gas, *AGA Monograph on Project No*. 31, American Gas Association, New York.

Katz, D. L. and Coats, K. H. (1968). *Underground Storage of Fluids*, Ulrich Books Inc., Ann Arbor, Michigan.

Kazemi, H. (1975). A reservoir simulator for studying productivity variation and transient behavior of a well in a reservoir undergoing gas evolution, *Trans. SPE of AIME*, **259,** pp. 1401–12 (JPT).

Kazemi, H., Merrill, L. S., Jr., Porterfield, K. L. and Zeman, P. R. (1976). Numerical simulation of water–oil flow in naturally fractured reservoirs, *Trans. SPE of AIME*, **261,** pp. 317–26 (SPEJ).

Kazemi, H., Shank, G. D. and Vestal, C. R. (1977). An efficient multicomponent numerical simulator, *SPE* 6890, 52*nd Annual Technical Conference and Exhibition*, Denver.

Keast, P. and Mitchell, A. R. (1966). On the instability of the Crank Nicholson formula under derivative boundary conditions, *Compt. J*., **9,** No. 1, pp. 110–14.

Kelley, L. G. (1968). *Handbook of Numerical Methods and Applications*, Addison-Wesley, Mass.

Kernighan, B. W. and Plauger, P. J. (1974). *The Elements of Programming Style*, McGraw-Hill, New York.

Killough, J. E. (1976). Reservoir simulation with history-dependent saturation functions, *Trans. SPE of AIME*, **261,** pp. 37–48 (SPEJ).

Kleppe, J. and Morse, R. A. (1974). Oil production from fractured reservoirs by water displacement, *SPE* 5084, 49*th Annual Fall Meeting*, Houston.

Knuth, D. E. (1973). *The Art of Computer Programming*, **3,** Addison-Wesley, Mass.

Ko, S. C. M. (1977). A sequential solution approach for the simulation of coning behavior in petroleum reservoirs, *M.Sc. Thesis*, University of Calgary.

Krogh, F. T. (1973). Algorithms for changing the step size, *SIAM J. Numer. Anal*., **10,** No. 5, pp. 949–65.

Kron, G. (1963). *Diacoptics, Piecewise Solution of Large Scale Systems*, McDonald, New York.

Kumar, A. (1977). Steady flow equations for wells in partial water-drive reservoirs, *J. Petrol. Technol.*, **29,** pp. 1654–6.

Kyte, J. R. and Berry, D. W. (1975). New pseudo functions to control numerical dispersion, *SPE* 5105, 49*th Annual Fall Meeting*, Houston.

Lamb, Sir Horace (1932). *Hydrodynamics*, 6th edn., Dover Publications, New York.

Lancaster, P. (1969). *Theory of Matrices*, Academic Press, New York.

Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Stand.*, **45,** No. 159, pp. 255–82.

Lanczos, C. (1952). Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Stand.*, **49,** No. 159, pp. 33–53.

Land, C. S. (1968). The optimum gas saturation for maximum oil recovery from displacement by water, *SPE* 2216, 43*rd Annual Fall Meeting*, Houston.

Land, C. S. (1968a). Calculation of imbibition relative permeability for two- and three-phase flow from rock properties, *Trans. SPE of AIME*, **243,** pp. 149–56 (SPEJ).

Lantz, R. B. (1970). Rigorous calculation of miscible displacement using immiscible reservoir simulators, *Trans. SPE of AIME*, 249, pp. 192–202 (SPEJ).

Lantz, R. B. (1971). Quantative evaluation of numerical diffusion (truncation error), *Trans. SPE of AIME*, **251,** pp. 315–20 (SPEJ).

Lapidus, L. (1962). *Digital Computation for Chemical Engineers*, McGraw-Hill, New York.

Lapidus, L. and Seinfeld, J. H. (1971). *Numerical Solution of Ordinary Differential Equations*, Academic Press, New York.

Larkin, B. K. (1964). Some stable explicit difference approximations to the diffusion equation, *Math. of Compt.*, **18,** No. 84, pp. 196–202.

Larson, C. (1971). The efficient use of FORTRAN, *Datamation*, **17,** pp. 24–31.

Le Blanc, J. L. (1971). A streamline simulation model for predicting the secondary recovery of oil, *Ph.D. Thesis*, University of Texas, Austin.

Ledet, W. P. and Himmelblau, D. M. (1970). *Decomposition Procedures for Solving of Large Scale Systems*, Academic Press, New York.

Lees, M. (1966). A linear three level difference scheme for quasilinear parabolic equations, *Math. of Compt.*, **20,** pp. 516–22.

Letkeman, J. P. and Ridings, R. L. (1970). A numerical coning model, *Soc. Petrol. Eng. J.*, **19,** No. 4, pp. 418–24.

Letkeman, J. P. (1976). Semi-direct iterative methods in numerical reservoir simulation, *SPE* 5730, *4th Symposium on Numerical Simulation of Reservoir Performance*, Los Angeles.

Leverett, M. C. and Lewis, W. B. (1941). Steady flow of gas–oil–water mixtures through unconsolidated sands, *Trans. SPE of AI ME*, **142,** pp. 107–16.

MacDonald, R. C. and Coats, K. H. (1970). Methods for numerical simulation of water and gas coning, *Trans. SPE of AI ME*, **249,** pp. 425–36 (SPEJ).

Marchuk, G. I. (1975). *Methods of Numerical Mathematics*, Springer-Verlag, New York.

Markowitz, H. M. (1957). The elimination form of the inverse and its applications to linear programming, *Management Science*, **3,**-pp. 255–69.

Martin, J. C. (1968). Partial integration of equations of multiphase flow, *Trans. SPE of AIME*, **243,** pp. 370–80 (SPEJ).

Matthews, C. S. and Russell, D. G. (1967). *Pressure Buildup and Flow Tests in Wells*, SPE Monograph Series, **1.**

McConnell, A. J. (1957). *Applications of Tensor Analysis*, Dover Publications, New York.

McCreary, J. G. (1971). A simple method for controlling gas percolation in numerical simulation of solution gas drive reservoirs, *Trans. SPE of AIME*, **251,** pp. 85–91 (SPEJ).

McCulloch, R. C., Langton, J. R. and Spivak, A. (1969). Simulation of high relief reservoirs, Rainbow field, Alberta, Canada, *J. Petrol. Technol*., **21,** pp. 1399–408.

McDonald, A. E. and Trimble, R. H. (1977). Efficient use of mass storage during elimination for sparse sets of simulation equations, *Soc. Petrol. Eng. J*., **17,** No. 4, pp. 300–16.

Meijerink, J. A. and Vander Vorst, H. A. (1977). An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. of Compt*., **31,** No. 137, pp. 148–62.

Mikhlin, S. G. (1964). *Variational Methods in Mathematical Physics*, translated by T. Boddington, Pergamon Press, London.

Mikhlin, S. G. and Smolitskiy, K. L. (1967). *Approximate Methods for Solution of Differential and Integral Equations*, American Elsevier, New York.

Mitchell, A. R. (1969). *Computational Methods in Partial Differential Equations*, J. Wiley & Sons, New York.

Monin, A. S. and Yaglom, A. M. (1971). *Statistical Fluid Mechanics: Mechanics of Turbulence*, MIT Press, Cambridge, Mass.

Morel-Seytoux, H. J. (1966). Unit mobility ratio displacement calculations for pattern floods in homogeneous medium, *Trans. SPE of AIME*, **237,** pp. 217–27 (SPEJ).

Morrow, N. R. and Harris, C. C. (1965). Capillary equilibrium in porous materials, *Trans. SPE of AIME*, **234,** pp. 15–24 (SPEJ).

Morrow, N. R. (1970). Irreducible wetting-phase saturations in porous media, *Chem. Eng. Sci*., **25,** No. 11, pp. 1799–816.

Morse, P. M. and Feshbach, H. (1953). *Methods of Theoretical Physics,* McGraw-Hill, New York.

Mrosovsky, I. and Ridings, R. L. (1974). Two-dimensional radial treatment of wells within a three-dimensional reservoir model, *Soc. Petrol. Eng. J*., **14,** No. 2, pp. 127–31.

Mungan, N. (1972). Relative permeability measurements using reservoir fluids, *Trans. SPE of AIME*, **253,** pp. 398–402 (SPEJ).

Muskat, M. (1937). *Flow of Homogeneous Fluids Through Porous Media*, McGraw-Hill, New York.

Muskat, M. (1949). *Physical Principles of Oil Production*, McGraw-Hill, New York.

Muskat, M. (1949a). Calculation of initial fluid distribution in oil reservoirs, *Trans. SPE of AIME*, **179,** pp. 119–27.

Nicolaides, R. A. (1975). On multiple grid and related techniques for solving discrete elliptic systems, *J. Compt. Phys*., **19,** pp. 418–31.

Nolen, J. S. and Berry, D. W. (1972). Tests of the stability and time-step sensitivity of semi-implicit reservoir simulation techniques, *Trans. SPE of A IME*, **253,** pp. 253–66 (SPEJ).

Nolen, J. S. (1973). Numerical simulation of compositional phenomena in petroleum reservoirs, *SPE* 4274, 3*rd Symposium on Numerical Simulation of Reservoir Performance*, Houston.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

Osher, S. (1970). Mesh refinements for the heat equation, *SIAM J. Numer. Anal*., **7,** No. 2, pp. 199–205.

Ostrowski, A. M. (1973). *Solution of Equations in Euclidean and Banach Spaces*, 3rd edn., Academic Press, New York.

Owens, W. W. and Archer, D. L. (1971). The effect of rock wettability on oil–water relative permeability relationships, *Trans. SPE of AIME*, **251,** pp. 873–8 (JPT).

Parter, S. V. (1959). On 'Two-line' iterative methods for the Laplace and biharmonic difference equations, *Numer. Math*., **1,** pp. 240–52.

Parter, S. V. (1961). Multi-line iterative methods for elliptic difference equations and fundamental frequencies, *Numer. Math.*, **3,** pp. 22–34.

Patton, J. T., Coats, K. H. and Colegrove, G. T. (1971). Prediction of polymer flood performance, *Trans. SPE of AIME*, **251,** pp. 72–84 (SPEJ).

Peaceman, D. W. and Rachford, H. H., Jr., (1955). The numerical solution of parabolic and elliptic differential equations, *J. SIAM*, **3,** No. 1, pp. 28–41.

Peaceman, D. W. and Rachford, H. H., Jr., (1962). Numerical calculation of multidimensional miscible displacement, *Trans. SPE of AIME*, **225,** pp. 327–39 (SPEJ).

Peaceman, D. W. (1966). *Lecture Notes on Numerical Solution of Elliptic and Parabolic Partial Differential Equations*, Esso Production Research Company.

Peaceman, D. W. (1967). Numerical solution of the nonlinear equations for two-phase flow through porous media, in: *Nonlinear Partial Differential Equations—Methods of Solutions*, ed. W. F. Ames, Academic Press, New York, pp. 79–91.

Peaceman, D. W. (1976). Convection in fractured reservoirs—Numerical calculation of convection in a vertical fissure, including the effect of matrix-fissure transfer, *Trans. SPE of AIME*, **261,** pp. 281–301 (SPEJ).

Peaceman, D. W. (1977). A nonlinear stability analysis for difference equations using semi-implicit mobility, *Soc. Petrol. Eng. J.*, **17,** No. 1, pp. 79–91.

Peaceman, D. W. (1977a). Interpretation of well-block pressures in numerical reservoir simulation, *SPE* 6893, 52*nd Annual Fall Technical Conference and Exhibition*, Denver.

Pease, M. C., III (1965). *Methods of Matrix Algebra*, Academic Press, New York.

Peng, D-Y. and Robinson, D. B. (1976). A new two-constant equation of state, *Ind. Eng. Chem. Fundam.*, **15,** No. 1, pp. 59–64.

Perkins, F. M., Jr., and Collins, R. E. (1960). Scaling laws for laboratory flow models of oil reservoirs, *Trans. SPE of AIME*, **219,** pp. 383–5 (JPT).

Polubarinova-Kochina, P. Ya. (1962). *Theory of Ground-Water Movement*, translated by J. M. Roger de Wiest, Princeton University Press, New Jersey.

Poston, S. W., Ysrael, S. C., Hossain, A. K. M. S., Montgomery, E. F., III and Ramey, H. J., Jr. (1970). The effect of temperature on irreducible water saturation and relative permeability of unconsolidated sands, *Trans. SPE of AIME*, **249,** pp. 171–80.

Poussin, F., de la Vallée (1968). An acceleration relaxation algorithm for iterative solution of elliptic equations, *SIAM. J. Numer. Anal.*, **5,** No. 2, pp. 340–51.

Pozzi, A. L. and Blackwell, R. J. (1963). Design of laboratory models for study of miscible displacement, *Trans. SPE of AIME*, **228,** pp. 28–40 (SPEJ).

Price, H. S., Varga, R. S. and Warren, J. E. (1966). Application of oscillation matrices to diffusion–convection equations, *J. Math. and Physics*, **45,** No. 3, pp. 301–11.

Price, H. S. (1971). Let the problem dictate the degree of sophistication to use in reservoir simulation, Paper 7110, *22nd Annual Technical Meeting Petrol. Soc. of CIM*, Banff.

Price, H. S. and Coats, K. H. (1974). Direct methods in reservoir simulation, *Trans. SPE of AIME*, **257,** pp. 295–308 (SPEJ).

Price, H. S., Settari, A. and Varga, R. S. (1978). Higher-order time approximations for nonlinear differential equations, to be published.

Quon, D., Dranchuk, P. M., Allada, S. R. and Leung, P. K. (1966). Application of the alternating direction explicit procedure to two-dimensional natural gas reservoirs, *Trans. SPE of AIME*, **237,** pp. 137–42 (SPEJ).

Raithby, G. D. (1976). A critical evaluation of upstream differencing applied to problems involving fluid flow, *Comp. Meth. in Appl. Mech. and Engineering*, **9,** No. 1, pp. 75–103.

Rapoport, L. A. (1955). Scaling laws for use in design and operation of water-oil flow models, *Trans. SPE of AIME*, **204,** pp. 143–50 (JPT).

Redford, D. A., Flock, D. L., Peters, E. and Lee, J. (1976). Laboratory model flow test systems of *in situ* recovery from Alberta Oil Sands, 26*th Can. Chem. Eng. Conf., Symposium on Tar Sands*, Toronto.

Reid, J. K. (1966). A method for finding optimum successive overrelaxation factor, *Compt. J.*, **9,** pp. 200–4.

Reid, J. K. (ed.) (1971). *Large Sparse Sets of Linear Equations*, Academic Press, London.

Richardson, L. F. (1910). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, *Philos., Trans. Roy. Soc. London Ser. A.*, **210**, pp. 307–57, and *Proc. Roy. Soc. London Ser. A.*, **83,** pp. 335–6.

Richardson, J. G. and Stone, H. L. (1973). A quarter century of progress in the application of reservoir engineering, *J. Petrol. Technol.*, **25,** pp. 1371–9.

Richtmyer, R. D. and Morton, K. W. (1967). *Difference Methods for Initial-Value Problems*, 2nd edn., Interscience Publishers, New York.

Roache, P. J. (1972). *Computational Fluid Dynamics*, Hermosa Publishers, New Mexico.

Robertson, G. E. and Woo, P. T. (1976). Grid orientation effects and the use of orthogonal curvilinear coordinate in reservoir simulation, *SPE* 6100, 51*st Annual Fall Meeting*, New Orleans.

Robinson, R. M. (1971). Numerical simulation of coning and free gas percolation in a solution-gas-drive reservoir around a single well, *Ph.D. Thesis*, University of Texas, Austin.

Rose, D. J. and Willoughby, R. A. (eds.) (1972). *Sparse Matrices and Their Applications*, Plenum Press, New York.

Rosenbrock, H. H. (1963). Some general implicit processes for the numerical solution of differential equations, *Compt. J.*, **5,** pp. 329–30.

Rossen, R. H. (1977). Simulation of naturally fractured reservoirs with semi-implicit source terms, *Soc. Petrol. Eng. J.*, **17,** No. 3, pp. 201–10.

Runchal, A. K. and Wolfstein, M. (1969). Numerical integration procedure for steady state Navier–Stokes equations, *J. Mech. Eng. Sci.*, **11,** No. 5, pp. 445–53.

Saul'jev, V. K. (1964). *Integration of Equations of Parabolic Type by the Method of Nets*, Pergamon Press, London.

Schaefer, M. (1973). *A Mathematical Theory of Global Program Optimization*, Prentice-Hall, New Jersey.

Schechter, S. (1960). Quasi-tridiagonal matrices and type-insensitive difference equations, *Quart. Appl. Math.*, **18,** pp. 285–95.

Schechter, R. S. (1967). *The Variational Method in Engineering*, McGraw-Hill, New York.

Scheidegger, A. E. (1974). *Physics of Flow Through Porous Media*, 3rd edn., University of Toronto Press, Toronto.

Schilthuis, R. J. (1936). Active oil and reservoir energy, *Trans. SPE of AIME*, **118,** pp. 33–52.

Schlichting, H. (1968). *Boundary Layer Theory*, 6th edn., McGraw-Hill, New York.

Schneider, G. E., Strong, A. B. and Yovanovich, M. M. (1975). Finite difference modelling of the heat conduction equation in general orthogonal coordinates using Taylor series expansion: in *Advances in Computer Methods for Partial Differential Equations*, ed. R. Vichnevetsky, AICA, Belgium.

Schwarz, H. R. (1968). *Numerik Symmetrischen Matrizen*, Springer-Verlag, Berlin (in German).

Seth, M. S. (1974). A semi-implicit method for simulating reservoir behaviour, *SPE* 4979, 49*th Annual Fall Meeting*, Houston.

Settari, A. and Aziz, K. (1972). Use of irregular grid in reservoir simulation, *Soc. Petrol. Eng. J.*, **12,** No. 2, pp. 103–14.

Settari, A. (1973). Numerical simulation of three-phase coning in petroleum reservoirs, *Ph.D. Thesis*, The University of Calgary, Calgary.

Settari, A. and Aziz, K. (1973). A generalization of the additive correction methods for the iterative solution of matrix equations, *SIAM J. Numer. Anal*., **10,** No. 3, pp. 506–21.

Settari, A. and Aziz, K. (1974). Use of irregular grid in cylindrical coordinates, *Trans. SPE of AIME*, **257,** pp. 396–412 (SPEJ).

Settari, A. and Aziz, K. (1974a). A computer model for two phase coning simulation, *Soc. Petrol. Eng. J*., **14,** No. 3, pp. 221–36.

Settari, A. and Aziz, K. (1975). Treatment of nonlinear terms in the numerical solution of partial differential equations for multiphase flow in porous media, *Int. J. Multiphase Flow*, **1,** pp. 817–44.

Settari, A., Price, H. S. and Dupont, T. (1977). Development and application of variational methods for simulation of miscible displacement in porous media, *Soc. Petrol. Eng. J*., **17,** No. 3, pp. 228–46.

Shacham, M. and Kehat, E. (1976). A direct method for the solution of large sparse systems of linear equations, *Compt. J*., **10,** No. 4, pp. 353–9.

Sheffield, M. (1969). Three phase flow including gravitational, viscous, and capillary forces, *Trans. SPE of AIME*, **246,** pp. 255–69 (SPEJ).

Sheffield, M. (1970). A non-iterative technique for solving parabolic partial differential equation problems, *SPE* 2803, *2nd Symposium on Numerical Simulation of Reservoir Performance*, Dallas.

Sheldon, J. W., Zondek, B. and Cardwell, W. T. (1959). One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium, *Trans. SPE of AIME*, **216,** pp. 290–6.

Shutler, N. D. (1969). Numerical, three-phase simulation of the linear steamflood process, *Trans. SPE of AIME*, **246,** pp. 232–46 (SPEJ).

Sigmund, P. M. (1976). Prediction of molecular diffusion at reservoir conditions. Part II. Estimating the effects of molecular diffusion and convective mixing in multicomponent systems, *J. Can. Petrol. Technol*., **15,** No. 3, pp. 53–62.

Slater, G. E. and Durrer, E. J. (1971). Adjustment of reservoir simulation models to match field performance, *Trans. SPE of AIME*, **251,** pp. 295–305 (SPEJ).

Smith, G. D. (1965). *Numerical Solution of Partial Differential Equations with Exercises and Worked Solutions,* Oxford University Press, New York.

Snell, R. W. (1962). Three-phase relative permeability in unconstituted sand, *J. Inst. Petrol*., **84,** pp. 80–8.

Soave, G. (1972). Equilibrium constants from a modified Redlich–Kwong equation of state, *Chem. Eng. Sci.,* 27, pp. 1197–203.

Sobocinski, D. P. and Cornelius, A. J. (1965). A correlation for predicting water coning time, *Trans. SPE of AIME*, **234,** pp. 594–600 (JPT).

Sokolnikoff, I. S. (1951). *Tensor Analysis: Theory and Application*, J. Wiley & Sons, New York.

Sonier, F., Besset, P. H. and Ombret, R. (1973). A numerical model of multiphase flow around a well, *Soc. Petrol. Eng. J.,* 13, No. 6, pp. 311–20.

Sonier, F. and Chaumet, P. (1974). A fully implicit three-dimensional model in curvilinear coordinates, *Trans. SPE of AIME*, **257,** pp. 361–70 (SPEJ).

Spillette, A. G., Hillestad, J. G. and Stone, H. L. (1973). A high-stability sequential solution approach to reservoir simulation, *SPE* 4542, 48*th Annual Fall Meeting,* Las Vegas.

Spivak, A. and Coats, K. H. (1970). Numerical simulation of coning using implicit production terms, *Trans. SPE of AIME*, **249**, pp. 257–67 (SPEJ).

Spivak, A. and Dixon, T. N. (1973). Simulation of gas condensate reservoirs, *SPE* 4271, *3rd Symposium on Numerical Simulation of Reservoir Performance,* Houston.

Spivak, A. (1974). Gravity segregation in two phase processes, *Trans. SPE of AIME*, **257**, pp. 619–32 (SPEJ).

Spivak, A., Perryman, T. L. and Norris, R. A. (1975). A compositional simulation study of the Sacroc Unit CO_2 Project, *9th World Petrol. Congress Proceedings,* Tokyo, **4**, Applied Science Publishers, London, pp. 187–200.

Spivak, A., Price, H. S. and Settari, A. (1977). Solution of the equations for multidimensional two phase immiscible flow by variational methods, *Soc. Petrol. Eng. J.*, **17**, No. 1, pp. 27–41.

Staggs, H. M. and Herbeck, E. F. (1971). Reservoir simulation models—An engineering overview, *J. Petrol. Technol.*, **23**, pp. 1428–36.

Steen, H. D. and Farouq Ali, S. M. (1971). Comparative evaluation of the strongly implicit procedure (SIP) for gas reservoir simulation, *J. Can. Petrol. Technol.*, **10**, No. 4, pp. 35–9.

Steffensen, R. J. and Sheffield, M. (1973). Reservoir simulation of a collapsing gas saturation requiring areal variation in bubble-point pressure, *SPE* 4275, *3rd Symposium on Numerical Simulation of Reservoir Performance*, Houston.

Stone, H. L. and Garder, A. O., Jr. (1961). Analysis of gas-cap or dissolved-gas reservoirs, *Trans. SPE of AIME,* **222**, pp. 92–104 (SPEJ).

Stone, H. L. (1968). Iterative solution of implicit approximations of multi-dimensional partial differential equations, *SIAM J. Numer. Anal.*, **5**, No. 3, pp. 530–58.

Stone, H. L. (1970). Probability model for estimating three-phase relative permeability, *Trans. SPE of AIME,* **249**, pp. 214–18 (JPT).

Stone, H. L. (1973). Estimation of three-phase relative permeability and residual oil data, *J. Can. Petrol. Technol*., **12**, No. 4, pp. 53–61.

Stright, D. S., Jr. (1973). Grand Forks—Modelling a three-dimensional reservoir with two-dimensional reservoir simulators, *J. Can. Petrol. Technol*., **12**, No. 4, pp. 46–52.

Stright, D. H., Jr., Aziz, K., Settari, A. and Starratt, F. (1977). Carbon dioxide injection into bottom-water, undersaturated viscous oil reservoirs, *J. Petrol. Technol*., **29**, pp. 1248–58.

Suarez, A. and Farouq Ali, S. M. (1976). A comparative evaluation of different versions of the strongly implicit procedure (SIP), *4th Symposium on Numerical Simulation of Reservoir Performance,* Los Angeles.

Tang, I. C. (1969). A simple algorithm for solving linear equations of a certain type, *Zeitschrift für Angewandte Mathematik und Mechanik,* Heft **8**, Band 49, S. 508.

Taussky, O. (1949). A recurring theorem on determinants, *Amer. Math. Monthly,* **56**, pp. 672–5.

Telkov, A. O. (1971). Solution of static problems in coning with account of changes in thickness of the oil-saturated portion of a formation, *Neft. Khoz.* **49**, No. 1, pp. 34–7 (in Russian).

Thomas, L. K., Katz, D. L. and Tek, M. R. (1968). Threshold pressure phenomena in porous media, *Trans. SPE of AIME*, **243**, pp. 174–84 (SPEJ).

Thomas, L. K. and Heliums, L. J. (1972). A non linear automatic history matching technique for reservoir simulation models, *Trans. SPE of AIME,* **253**, pp. 508–14 (SPEJ).

Thomas, L. K., Lumpkin, W. B. and Reheis, G. M. (1976). Reservoir simulation of variable bubble-point problems, *Trans. SPE of AIME*, **261**, pp. 10–16 (SPEJ).

Thurnau, D. H. (1963). Algorithm 195, BANDSOLVE, *Comm. ACM*, **6**, No. 8, p. 441.

Tinney, W. F. and Walker, J. W. (1967). Direct solutions of sparse network equations by optimally ordered triangular factorization, *IEEE Proc*., **55**, pp. 1801–9.

Todd, J. (1962). *Survey of Numerical Analysis*, McGraw-Hill, New York.

Todd, M. R. and Longstaff, W. J. (1972). The development, testing and application of a numerical simulator for predicting miscible flood performance, *Trans. SPE of AIME,* **253**, pp. 874–82 (JPT).

Todd, M. R., O'Dell, P. M. and Hirasaki, G. J. (1972). Methods for increased accuracy in numerical reservoir simulators, *Trans. SPE of AIME,* **253**, pp. 515–30 (SPEJ).

Toronyi, R. M. and Farouq Ali, S. M. (1974). Determining interblock transmissibility in reservoir simulators, *J. Petrol. Technol*., **26**, No. 1, pp. 77–8.

Traylor, C. and Sheffield, M. (1971). The comparable efficiency of mathematical techniques used in reservoir simulation, *SPE* 3560, 46*th Annual Fall Meeting,* New Orleans.

Trimble, R. H. and McDonald, A. E. (1976). A strongly coupled, implicit well coning model, *SPE* 5738, *4th Symposium on Numerical Simulation of Reservoir Performance,* Los Angeles.

Van-Quy, N., Simandoux, P. and Corteville, J. (1972). A numerical study of diphasic multicomponent flow, *Trans. SPE of AIME,* **253**, pp. 171–84 (SPEJ).

Van Everdingen, A. F. and Hurst, W. (1949). Application of the Laplace transformation to flow problems in reservoirs, *Trans. SPE of AIME,* **186**, pp. 305–24.

Van Leer, B. (1977). Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, *J. Compt. Phys*., **23**, pp. 276–99.

Van Poollen, H. K., Breitenbach, E. A. and Thurnau, D. H. (1968). Treatment of individual wells and grids in reservoir modelling, *Soc. Petrol. Eng. J*., **8**, No. 4, pp. 341–6.

Varga, R. S. (1962). *Matrix Iterative Analysis,* Prentice-Hall, New York.

Vela, S., Peaceman, D. W. and Sandvik, E. I. (1976). Evaluation of polymer flooding in a layered reservoir with crossflow, retention, and degradation, *Soc. Petrol. Eng. J*., **16**, No. 2, pp. 82–96.

Vinsome, P. K. W. (1976). Orthomin, an iterative method for solving sparse banded sets of simultaneous linear equations, *SPE* 5279, *4th Symposium on Numerical Simulation of Reservoir Performance,* Los Angeles.

Von Rosenberg, D. U. (1969). *Methods for the Numerical Solution of Partial Differential Equations, American Elsevier, New York.*

Wachspress, E. L. and Habetler, G. J. (1960). An alternating-direction-implicit iteration technique, *J. SIAM,* **8**, pp. 403–24.

Wachspress, E. L. (1962). Optimum alternating-direction iteration parameters for a model problem, *J. SIAM*, **10**, pp. 339–50.

Wachspress, E. L. (1966). *Iterative Solution of Elliptic Systems,* Prentice-Hall, New Jersey.

Wallis, J. and Aziz, K. (1975). Some aspects of the computer modelling of pollution from an atmospheric source, *J. Can. Petrol. Technol*., **14**, No. 1, pp. 21–5.

Warner, H. R., Jr. (1977). An evaluation of miscible CO_2 flooding in waterflooded sandstone reservoirs, *J. Petrol. Technol*., **29**, pp. 1339–48.

Warren, J. E. and Root, P. J. (1963). The behaviour of naturally fractured reservoirs, *Trans. SPE of AI ME*, **228**, pp. 245–55 (SPEJ).

Warren, J. E. and Cosgrove, J. J. (1964). Prediction of waterflood behavior in a stratified system, *Trans. SPE of AIME,* **231**, pp. 149–57 (SPEJ).

Wasserman, M. L. and Emanuel, A. S. (1976). History matching three dimensional models using optimal control theory, *J. Can. Petrol. Technol*., **15**, No. 4, pp. 70–7.

Wattenbarger, R. A. and Ramey, H. J. (1968). Gas well testing with turbulence, damage and wellbore storage, *Trans. SPE of AIME,* **243**, pp. 877–87 (JPT).

Wattenbarger, R. A. and Thurnau, D. H. (1976). Application of SSOR to three-dimensional reservoir problems, *SPE* 5728, *4th Symposium on Numerical Simulation of Reservoir Performance,* Los Angeles.

Watts, J. W. (1971). An iterative matrix solution method suitable for anisotropic problems, *Trans. SPE of AIME,* **251**, pp. 47–51 (SPEJ).

Watts, J. W. (1972). Private communication.

Watts, J. W. (1973), A method for improving line successive overrelaxation in anisotropic problems—A theoretical analysis, *Trans. SPE of AIME,* **255**, pp. 105–18 (SPEJ).

Watts, J. W. and Huang, H. Y. (1976). An approximate three dimensional reservoir simulation model, *SPE* 6103, *51st Annual Fall Technical Conference and Exhibition,* New Orleans.

Weaver, R. H. (1972). Simulation of waterflood behavior in a reservoir previously invaded by water, *J. Petrol. Technol*., **24**, pp. 909–15.

Weinstein, H. G., Stone, H. R. and Kwan, T. V. (1969). Iterative procedure for solution of parabolic and elliptic equations in three dimensions, *Ind. Eng. Chem. Fund*., **8**, pp. 281–7.

Weinstein, H. G., Stone, H. L. and Kwan, T. V. (1970). Simultaneous solution of multiphase reservoir flow equations, *Soc. Petrol. Eng. J*., **16**, No. 2, pp. 99–110.

Welge, H. J. (1949). Displacement of oil from porous media by water or gas, *Trans. SPE of AIME,* **179**, pp. 133–45.

Welge, H. and Weber, A. G. (1964). Use of two-dimensional methods for calculating well coning behavior, *Trans. SPE of AIME,* **231**, pp. 344–55 (SPEJ).

West, W. J., Garvin, W. W. and Sheldon, S. W. (1954). Solution of the equations of unsteady state two-phase flow in oil reservoirs, *Trans. SPE of AIME,* **201**, pp. 217–29.

Westlake, J. R. (1968). *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations, John Wiley & Sons, New York.*

Whitaker, S. (1966). The equations of motion in porous media, *Chem. Eng. Sci*., **21**, pp. 291–300.

Whitaker, S. (1967). Diffusion and dispersion in porous media, *A.I.Ch.E. J*., **13**, No. 3, pp. 420–7.

Whitaker, S. (1969). Advances in theory of fluid motion in porous media, *Indust, and Engineering Chemistry*, **61**, No. 12, pp. 14–28.

Wilkinson, J. H. (1963). *Rounding Errors in Algebraic Processes*, Prentice-Hall, New Jersey.

Wilkinson, J. H. and Reinsch, C. (1971). *Linear Algebra Handbook for Automatic Computation*, Vol. 2, Die Grundlehren der Mathematischen Wissenschaften in Einzelldarstellungen, Band. **186**, Springer-Verlag, New York.

Wilson, G. M. (1969). A modified Redlich–Kwong equation of state, application to general physical data calculations, Paper 15C, *65th National A.I.Ch.E. Meeting*, Cleveland.

Wilson, E. A. (1975). A special version of the Crout reduction for variable bandwidth matrices, *Summer Computer Simulation Conference*, San Francisco, pp. 104–12.

Woo, P. T., Roberts, S. J. and Gustavson, F. G. (1973). Application of sparse matrix techniques in reservoir simulation, *SPE* 4544, 48*th Annual Fall Meeting*, Las Vegas.

Woo, P. T. and Emanuel, A. S. (1976). A block successive overrelaxation method for coupled equations, *SPE* 6106, 51*st Annual Fall Technical Conference and Exhibition*, New Orleans.

Woods, E. G. and Khurana, A. K. (1977). Pseudofunctions for water coning in a three dimensional reservoir simulator, *Soc. Petrol. Eng. J*., **17**, No. 4, pp. 251–62.

Wright, D. E. (1968). Nonlinear flow through granular media, *J. Hydraulics Division*, *Proc. Amer. Soc. Civ. Engrs*., **94**, pp. 851–72.

Yamamoto, R. H., Ford, W. T., Padgett, J. B. and Boubeguira, A. (1971). Compositional reservoir simulator for fissured systems—The single block model, Soc. *Petrol. Eng.* J., **11**, No. 2, pp. 113–28.

Yanenko, N. N. (1971). *The Method of Fractional Steps*, Springer-Verlag, New York.

Yanosik, J. L. and McCracken, T. A. (1976). A nine-point finite difference reservoir simulator for realistic prediction of unfavourable mobility ratio displacements, *SPE* 5734, 4*th Symposium on Numerical Simulation of Reservoir Performance,* Los Angeles.

Young, D. M. (1971). *Iterative Solution of Large Linear Systems,* Academic Press, New York.

Young, D. M. and Gregory, R. T. (1973). *A Survey of Numerical Mathematics*, **1**, and **2**, Addison-Wesley, Mass.

INDEX

***Note about the Index**: Since the pagination of an e-book is dependent on the software and magnification, we decided to leave the page numbers as they refer to the hard/soft cover editions, to serve as an approximate guide to the reader.*

Alternating direction explicit (ADE) methods, 228
  ADI, comparison with, 233
  problems, 3-D, 36
Alternating direction implicit (ADI) methods, 230–3
  ADE, comparison with, 233
  equations, block pentadiagonal, 353–4
  problems, 3-D, 360–1, 365–6
Amplification factor, 59, 60, 141
Analog models, 2
Approximate solution, 54
Approximation, local order of, 53–5
Aquifers
  influx, calculation methods, 325–8
  'pot', 325
  simulation, 325–8
  steady-state infinite, 326
  unsteady-state, 326
Areal models, 211, 213, 220, 304–5
Areal problems, 204, 304–5, 316, 328–37
Arithmetic mean, 84
Automatic time-step control, 403–4
  algorithm, 403–4

Backward difference, 52
  approximation, 46, 57
Band matrices, 203
Basis functions, 50
β-model, 9–11
  see also Black-oil model
Black-oil model, 1, 9–11

simulation of compositional effects, using, 394–5
Block
pentadiagonal equations, 350–6
methods
BSOR, 352
comparison of iterative and direct, 355
direct, 351–2
iterative, 352–5
iterative ADI, 353–4
SIP, 354
pressure, 84
tridiagonal equations, solution of, 200–3
volume, 212
Boundary conditions, 69, 70, 213–19, 244
boundary open to flow, 317
closed boundary, 317
compatibility conditions and constraints, 319
cyclic, 74, 75
differential form of, 184
Dirichlet, 70, 94
discretisation of, 187–93, 216–19
distribution of production between blocks, 321
finite-difference formulation, 319
flow boundaries, 215–16
kinds of
1st, 70
2nd, 71
3rd, 73
4th, 74
multiphase flow in two dimensions, 316–23
Neumann, 120
no-flow or closed boundaries, 215
problem of assigning, 318
time approximation of, 192
Boundary effects, 184
Boundary value problem, 50, 103
Bounding curves, 396
Bubble-point pressure, 384–9
Buckley–Leverett

problem, 151, 153, 177
theory of two-phase displacement, 145

Capillary
hysteresis, 396
pressure, 29
Central-difference approximation, 46–7
‘Chapeau’ functions, 49
Choleski’s algorithm, 120
Chord slope, 128
Compositional effects, simulation of, 394–5
Compositional models, 394
Compressible fluids, general equation for, 13
Computer
models, 1, 2
development of, 407–12
questions answerable by, 3–4
scope of, 3
programs, 1
selected, 433–9
Coning
models
reservoir models, coupling with, 332
studies of, 332
problems, 306–7
simulation, 332
Consistency, local order of, 53–4
Consistent approximations, 54
Convection–diffusion equations, 154, 390–2
Convergence, 54, 150, 166, 292–4
explicit method, of, 56
implicit method, of, 57
Crank–Nicholson (CN) method, 62–3, 103, 211, 213, 231, 232
Critical saturation, 30
Cross-sectional model, 211, 216, 220
Cross-sectional problems, 206, 305, 316, 328–37
Curvilinear grid, 235–41, 328–9

Darcy velocity, 391

Darcy's law, 5, 16, 17, 24–7, 184, 221, 318
  multiphase flow, and, 12–13
  single-phase flow, and, 11–12
Densities of phases at reservoir conditions, 10
Difference approximation of positive type, 99
Difference equations
  boundary points, for, 319
  irregular grid, for, 75, 87–90
  matrix form, 81–2, 90–2, 245
  positive type, 213
Difference operator, 55
  definition, 54
Difference schemes, 60
stability of, 213
Differential equations, ordinary, 51–2
Differential formulation, 317
Differential operator, 49
Dirac δ-functions, 85
Dirichlet boundary conditions, 94
Discretisation
  blocks, into, 48
  boundary conditions, of, 187–93, 216–19
  error, 45, 53, 58
  flow equations, of, 209–13
  matrix form, of, 226–8
  one-dimensional flow equations, of
    Cartesian co-ordinates, in, 75–85
    radial cylindrical co-ordinates, in, 86–92
  space, in, 44
  three-dimensional problems, of, 358–9
  time, in, 51
  two dimensions, in, 307–16
Discretised problem, 50
Douglas and Rachford method, 232
Douglas method, 231
Dual-porosity system, 401, 402

Eigenvalues, 427
Eigenvectors, 427

Elemental models, 2–3
Elliptic problems, 94
Energy Resources Conservation Board of Alberta (ERCB), 207, 220, 221
Errors of solution, 55
Euler's method, 64
  modified, 65
Explicit method, 52, 61–2, 67
  convergence of, 56
Explicit Runge–Kutta methods, 66
Explicit time approximations, 98–100
Extrapolation, 104

Field, definition, 422
Finite-difference approximations, 43–60, 127, 209
Finite-difference equation(s), 93–105
  material balance, 150
  solution and material balance, existence of, 93–7
Finite-difference operator, 55, 77
Fixed tangent and fixed secant method, 101
Flow boundaries, 215–16
Fluid flow equations, 5–41
  boundary conditions, 17–18
  discretisation of, 209–13
  high flow rates (inertial and turbulent effects), 24–5
  miscellaneous effects, 27
  multiphase flow, 16
    alternative form, 18–24
    'hyperbolic' form, 20–4
    'parabolic' form, 18–20
  non-Darcy effects, 24–7
  pseudopotential, use of, 17–18
  single-phase flow, 13–16
Fluid motions in porous media, 5–6
Fluid properties, 27
Forward-difference approximation, 46
Fourier series
  method, 59
  stability analysis, 140, 141
Fractional flow equations, 140

Fractional step methods, 230

Gas
  flow equations, 15, 25
  percolation, 169–71
Gas–oil system, 32, 35
Gauss–Seidel method, 353
Global discretisation error, 54
Green's theorem, 48
Grid(s)
  construction, 234–41
    curvilinear, 235–41
    irregular, in 2-D, 234
    two methods of, 69
  definition, 417
  design, 416
  orientation
    effect, 156
      factors influencing, 334
      techniques for eliminating, 335
    phenomenon, 332–7
  point, pressure at, 220
  selection of, 416
  sensitivity study, 417
  systems, 69

History matching, 418–20
Hyperbolic equations, 156
Hysteresis
  capillary pressure and relative permeabilities, of, 395
  numerical treatment of, 398–401
physical model of, 396–8

Implicit method, 52, 57, 62, 68
Implicit pressure–explicit saturation method (IMPES), 125, 135–8, 290, 313, 403
  analysis, 138–50
  derivations of, other, 137–8
  existence and uniqueness of solution, 149–50
  method, selection of, 417
  stability of, 139–43

transmissibilities, with respect to, 144—6
2-D, in, 309–13
three-dimensional problems, 371
three-phase flow, for, 135–7
Implicit Runge–Kutta type methods, 66
Implicit time approximations, 100
Implicit transmissibilities, linearised, 159
Inertia–turbulent (IT) effects, 25
Initial conditions, 219
Input/output, use of, 410
Integral method, 47–9
INTERCOMP models, 345
Irreducible saturation, 29
Irregular grid, difference equations, 87–90

Jacobi matrix, 159, 165
Jacobian, 66

Klinkenberg phenomenon, 25

Laminar–inertia–turbulent (LIT) equation, 25
Law of mass conservation
multiphase flow, and, 8–11
single-phase flow, and, 6–8
Lax's equivalence, 58
Linear equations, 424
Linear vector spaces, definition, 423–4
Linearisation methods, 104–5
Locally one-dimensional method, 230, 232
Logarithmic mean, 88–90
radius, 88
LU factorisation, 250–2

Markowitz criterion, modified, 297
Mass transfer between oil and gas phases, 9
Material balance, 94, 165, 182
properties of difference equations to ensure, 97
Material balance equation, 339
Mathematical models, 1–2, 8
development of, 5, 406–7

Matrix(ces), 424
algebra, 422–31
definitions and basic concepts, 422–30
fundamental theorems, 430–1
definitions, 428–30
determinant of, 426
equations, methods for solving, 372–3
method, 59–60
norms of, 427
Matrix-banding schemes, 257
Methods, comparison of, 67–8
Miscible displacement
immiscible simulation, using, 392–4
simulation of, 390–4
Model
dimensionality of, 415
selection and design, 414–18
Multiphase flow
Darcy's law, and, 12–13
fluid flow equations, 16
alternative forms, 18–24
law of mass conservation, and, 8–11
methods, comparison of, 373
one dimension, in, 125–99
non-linearities, treatment of, 150–71
production terms, treatment of, 183–93
solution method
implicit pressure–explici
saturation (IMPES), 135–8
sequential (SEQ), 171–83
simultaneous (SS), 126–34
production in, handling of, 330
three dimensions, in, 371–3
two dimensions, in, 303–49
conditions
boundary, 316–23
initial, 323—5
solution methods and their comparison, 307–16
computer requirements, with, 313

Natural ordering, 351
Naturally fractured reservoirs, 401–2
Navier–Stokes equations, 135, 155
Neumann
  boundary conditions, 120
  problem, 94
Newton iteration, 347
  feasibility of, 344
Newtonian fluids, 24
Newton's method (or Newton–Raphson method), 100, 102, 103, 105, 161, 166–8
  modified, 168
Nonlinearities, 220
  P_c function, due to, 167
  treatment of, 97–105, 150–71
Non-Newtonian behaviour, 26
Non-unique solution, special cases of, 120
Numerical model, 2, 18
  development of, 407

Oil relative permeability, 33
Oil–water system, 32, 35
Ordering schemes, 252–3
Ordinary differential equations (ODE), 63–4, 68
Orthogonal co-ordinate system, 238
Outlet (or boundary) effect, 185
Overall efficiency, 315
Overshoot, handling of, 347

Parabolic equation, 51
Parabolic problems, 95
Peaceman–Rachford method, 126, 230
Pentadiagonal matrix equations, 244–302
  block, *see* Block pentadiagonal equations
  methods
    additive correction, 268–71
    direct, 250–60
      comparison with iterative methods, 294–97
    incomplete LU decomposition, 281
    iterative, 261–94

ADI, 271–7, 292
block, 261
comparison of, 281–9
comparison with direct methods, 294–7
point, 261
practical considerations, 290–4
line and block SOR, 267–8, 291–2, 295–6
one-dimensional correction, 268–70
Peaceman–Rachford, 275
point Gauss–Seidel, 264
point Jacobi, 263
point successive over relaxation, 265
power, 266
slot, 271
strongly implicit, 277–81
two-dimensional correction, 270
Wachspress, 275
ordering schemes, 252–3
Physical models, 2–3
Porosity
rock, of, 38
values, 6
Porous media, fluid motions, 5–6
Potential allocation method, 322
Predictor–corrector method, 233
Pressure
conditions, 190
grid point, at, 220
Pressure-dependent properties, 84
Productivity coefficient, 226
Program
development, 406–12
portability, 412
usage, 412–20
Programming efficiency, 408
Pseudo-capillary pressure, 379
Pseudo-functions, 376–83
displacement in stratified reservoirs controlled by viscous forces, for 382
dynamic, 382

individual wells, for, 383
Pseudo-optimal ordering schemes, 257, 260
Pseudo-relative permeability curves, 331, 380

Redlich–Kwong equation, 394
Reflection technique, 72, 216
Relative permeability, 30–8
Reservoir
engineering, 1
fluids, 416
performance prediction, 1
simulation, 1
volume, 174
Residual saturation, 30
Reynold's number, 24
Rock properties, 29
Rounding errors, 58
Run-time control parameters, 418
Runge–Kutta method, 65

Saturation
equations, 183
functions, history-dependent, 395–401
gradient, 187
pressure, 386, 387, 388
Scaled models, 2–3
Scanning curves, 396, 398
Second-order methods, 66
Semi-implicit method, 104
Sequential solution (SEQ) method
automatic time-step control, 403
multiphase flow
one dimension, in, 171–83
two dimensions, in, 313–16
numerical results, 177–80
other forms and derivatives, 176
selection of, 417
simultaneous solution (SS) model, comparison with, 345
three-dimensional problems, in, 371

three-phase flow, 180–3
two-phase flow, 171–5
variations of, 182
Simulation study, 412–15
Simultaneous solution (SS) method
analysis, 138–50
automatic time-step control, 403
equations of matrix form, 128
existence and uniqueness of solutions, 146–9
formulation
explicit gravity terms, without, 132–4
p, *S*, in terms of, 133–4
multiphase flow
one dimension, in, 125–34
two dimensions, in, 313–16
remarks on, 131
selection of, 417
sequential solution (SEQ) model, comparison with, 345
stability
one dimension, in, 139, 143
two dimensions, in, 309–13
with respect to transmissibilities, 144–6
three-dimensional problems, 371
three-phase flow, 131–2
Single-phase flow
Darcy's law, and, 11–12
fluid flow equations, 13–16
law of mass conservation, and, 6–8
one dimension, in, 42–112
three dimensions, in, problems, 358–71
two dimensions, in, 204–43
special methods for, 228–34
treatment of individual wells, 220
Single-well problems, 207, 220, 306, 330, 337–47
stability and efficiency of various treatments of transmissibilities, comparison of, 342–6
Slightly compressible fluid, equations for, 13
Slip phenomenon, 25
Source term, 72, 85

Sparse matrix techniques, 253–60
Sparsity-conserving ordering schemes, 257
Splitting-up methods, 230, 232
Stability, definition of, 58
Steady-state linear flow problems, 42
Stone Model
  I, 35
  II, 36
Storage
  optimisation, 411
  requirements, 314–15
Stream tube models, 383–4
Streamlines, distribution of, 384
Strong nonlinearities, 151
Strongly implicit (SIP) algorithm, 373
Strongly, implicit (SIP) method, three-dimensional problems, 367–9
Subroutines, 410
Swelling curves, 384

Tang's algorithm, 118–19
Taylor series method, 45
Theory of Pade approximations, 52
Third-order method, 67
Thomas' algorithm, 114–17, 201, 248, 342
Three-dimensional problems, 357–75
  basic equation and discretisation, 358–9
  methods
    alternating direction explicit (ADE), 360
    alternating direction implicit (ADI), 360–1
    BSOR and related, 364–5
    comparison of, 369–71
    direct, 361–3
    implicit pressure–explicit saturation (IMPES), 371
    iterative, 364
      alternating direction implicit (ADI), 365–6
    sequential solution (SEQ), 371
    simultaneous solution (SS), 371
    strongly implicit (SIP), 367–9
  multiphase flow, 371–3

number of multiplications and divisions per iteration per grid point for various iterative methods, 371
single-phase flow, 358–71
special, for, 359–61
Three-phase flow, 126
equations, 200
methods
implicit pressure–explicit saturation (IMPES), 135–7
sequential solution (SEQ), 180–3
simultaneous solution (SS), 131
Threshold
phenomenon, 25
pressure, 29
Time derivative approximations, 127
Time steps, selection of, 418
Transient linear flow problems, 42, 43
Transient radial flow problems, 43
Transmissibility(ies)
allocation method, 322
averaging of, 83, 84
coefficients, slopes in, 346
comparison of methods for implicit treatment of, 166–7
explicit, 157
implicit treatment, 337
linearised implicit, 159
production terms, in, 337
simple iteration on Matrix **T**, 158
stability with respect to, 312
time, approximation to, in, 156–7
Trial solution, 59
Tridiagonal matrix equations, 113–24
symmetric, 119–20
see also Block tridiagonal equations
Truncation error, 45, 53, 155–6, 333
Turbulence factor, 24
Two-dimensional flow
multiphase 303–49
single fluid, of, 204–43
Two-dimensional models, 208

Two-dimensional problems
classification of, 204–8, 303–7
number of multiplications and divisions per iteration per grid point for various iterative methods, 371
Two-phase flow, 125, 390
sequential solution (SEQ) method for, 171–5
simultaneous solution (SS) method for, 126

Upstream weighting, 153–5

Variable coefficients, 83, 91–2
Variable secant or chord method, 101
Variable tangent method, 101
Variational method, 101
Vectors
definition, 422
norms of, 427
Vertical equilibrium
concept, 381
model, 377

Water–oil systems, 34
Weak nonlinearities, 151
Weighting formulae, 150, 153, 155
Well
model
block-centred grid or off-centre well, for, 341–2
centre well, for, 338–41
production terms, 337–42
pressure, prediction of, 330
Wellbore equations, combination with reservoir equation, 340