APPLICATION OF BOUNDARY ELEMENT METHOD TO STREAMLINE GENERATION AND PRESSURE TRANSIENT TESTING

Jitendra Kikani

July 1989

Reservoir Simulation Industrial Affiliates

APPLICATION OF BOUNDARY ELEMENT METHOD TO STREAMLINE GENERATION AND PRESSURE TRANSIENT TESTING

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FORTHEDEGREEOF

DOCTOR OF PHILOSOPHY

By

Jitendra Kikani

September 1989

ABSTRACT

Reservoir geometry and heterogeneities significantly influence the flow field and consequently the pressure transient response in underground reservoir fluid flow. With the advent of accurate instrumentation, it is important to include the behavior of complex reservoir systems in the pressure transient design and analysis procedures.

Boundary element method is used in this work to account for such effects. The usefulness of **this** method lies in the fact that the solutions obtained are highly accurate and do not suffer from the usual drawbacks of other domain type numerical schemes. *Also* complex reservoir geometries with multiple wells can **be** handled due to the good boundary conformance obtained with the elements.

Such desirable features are realized because the analytical nature of the solution is preserved due to the use of free space Green's function of the governing differential operator as the weighting function in the weighted residual approach. **A** collocation type method is used for the solution of the resulting integral equations. Also, since the method is a boundary procedure, the dimensions of the problem are reduced by one. This reduction in dimensionality is obtained in cases where there are no distributed sources/sinks in the problem domain and the initial conditions are homogeneous. The use of the Green's function restricts the application of the method to linear problems with constant coefficients.

A variety of problems have been considered herein. Generation of streamlines is demonstrated in odd shaped reservoirs with multiple wells. Two different formulations for transient flow of single phase fluids in homogeneous anisotropic porous media are also presented. Both real space and Laplace space formulations are derived and compared. Also the boundary element method is proposed to determine the pressure solution in a piecewise homogeneous reservoir with arbitrary geometry of each region. This formulation can solve fluid injection problems which show composite behavior (steam injection, CO_2 flooding, in-situ combustion). In addition impermeable barriers of any shape and orientation, as well as large pressure support sources (aquifers) can be included.

Both pressure and pressure derivative behavior of such systems are studied. Example solutions are verified against known analytic solutions. Numerical features of the boundary element method, such as accuracy, consistency, and the optimum number of nodal points, are investigated in detail.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Development of sound exploitation scheme for petroleum and geothermal reservoirs is contingent upon characterizing the reservoirs as well as possible. **A** synthesis of the conceptual reservoir model proposed by the geoscientists into the modeling efforts for fluid flow through the porous rocks can theoretically lead to a stage when accurate solutions to complex problems may be obtained and the interpretation of field tests could be performed with confidence. The major reason for this integration is that the reservoirs are normally of complex configuration, and the performance depends quite strongly on these factors. Also, numerous wells producing from the reservoirs alter the flow field significantly.

Traditionally, pressure transient tests have been one of the primary sources of information for reservoir rock transmissivity and storativity. The pressure data collected from the wells are matched against an assumed model whose behavior is known by analytical or numerical means. Once a reasonable match is obtained, the parameters are calculated based on the solution to the assumed model. These predictive models are also used to design well tests so that a particular feature can be seen during the test to help evaluate the reservoir better. These models have usually been generated by either analytical or numerical methods. In this work, a method which has a strong analytical basis but uses numerical schemes to evaluate the solutions is investigated.

1.1 Analytical Methods

There are only a few conventional analytic methods for solving the equations goveming the fluid transport exactly. The important methods are separation of variables, eigenfunction expansion, similarity transform, Laplace transform, Fourier transform and Green's functions. To solve a partial differential equation, both the equation and the boundary conditions should **be** expressed in a separable orthogonal coordinate system. This limits the types **of** problems that can be solved analytically. Thus, one of the major shortcomings of the analytic methods is that only a few simple geometric shapes lend themselves to closed form solutions. Another level of flexibility in handling geometric shapes and boundary conditions is obtained by the use **of** the linearity property of the differential operators, in the form of superposition. In this method, image wells are used to generate different types of boundary conditions and boundary shapes. *Larsen (1985)* discusses the possible reservoir geometries that are amenable to analytic solutions by the technique of superposition. He gave **a** general algorithm for the location of images for computing the pressure transient behavior at a well. *Earlougher et al (1968)* and *Earlougher (1977)* presented the solution and interpretation of pressure behavior in bounded rectangular systems. The superposition method is a standard technique extensively used in the field of pressure transient testing.

For problems governed by the Laplacian operator (steady state problems) the solutions are analytic functions which satisfy the Cauchy-Riemann equations *(Spiegel, 1964).* The technique of hodograph transformation, which is a form of conformal mapping can be used to solve problems of complex geometry. The equations obtained for bounded systems are often complex elliptic integrals which must be evaluated numerically.

1.2 Numerical Methods

The shortcomings of the analytical methods are alleviated by numerical schemes which have a greater flexibility in solving complex problems. However, this is at the expense of accuracy. In the classical numerical schemes the continuous character of the governing equation on the domain is compromised in terms of approximate solutions at a discrete number of locations. Finite difference and finite element methods belong to this category. Although complex non-linear problems can be solved by these methods, they suffer from a variety of errors which are sometimes difficult to characterize. A brief description and shortcomings of both the finite difference and finite element methods is presented next.

1.2.1 Finite Difference Method

This method is the most widely used in the oil and geotherrnal industries. In the finite difference method, the governing differential equations are discretized and solved approximately at each of the nodal locations. The matrix structure obtained for solving the equations simultaneously is sparse and banded. The special structure of the matrix makes the solution amenable to fast matrix inversion algorithms.

The source of errors in the finite difference method (FDM) are due to a variety of reasons. The solutions obtained with the FDM are affected, at times strongly, by the orientation of the grid blocks with respect to the flow field. This is particularly true when there are preferred flow channels or paths in the system. The fluid fronts are smeared due to the grid orientation effects. In addition, flexibility in gridding is restricted to mostly rectangular grids. Although, special techniques have been developed which use triangular grids. Other problems associated with the finite difference method include numerical dispersion and discretization errors. These errors affect the accuracy of solutions to a significant degree, too. For a description of the various aspects of finite difference methods see *Roache (1972).* In spite of all the drawbacks, finite difference methods are the most powerful general schemes available for solving complex nonlinear problems.

1.2.2 Finite Element Method

The finite element technique (FEM) is based on a weak formulation of the governing differential equations. This method has a strong theoretical basis and a number of properties, which are important in considerations of a sound numerical method, can be proved for a few differential operators (Hughes, 1987). Two sets of approximating functions known as shape functions and weighting functions respectively, are defined on each of the elements. The shape (or basis) functions defined piecewise on each element, are said to have local compact support. This means that the basis functions are zero on all the other elements except on the element they are defined for. Local compact support keeps the resulting matrix (called the stiffness matrix) sparse and banded as in the finite difference method. Enormous gains in flexibility of element shapes and interpolation functions are gained over the FDM by the use of isoparametric elements. A good comparison of the finite difference and finite element method is given by ***Russell and Wheeler** (1983).* One of the popular schemes in the finite element methods is the Galerkin method. The Galerkin formulation uses both the weighting and the interpolation functions from the same set of finite dimensional functions. Other techniques, such as Petrov-Galerkin method ***(Johnson,** 1987)* uses a different set of weighting and interpolation functions. The Petrov-Galerkin method is being used in fluid mechanics.

Though a better representation of boundary shape and boundary conditions is obtained with the finite element methods, the problems of numerical dispersion still remain. *An* additional disadvantage is the complexity and the amount of input data that must be provided. The requirement for a large amount of input data is primarily clue to the need to provide details of connectivity between the elements. An important limitation of all numerical schemes is the difficulty in handling the advective or convective terms in the differential equations. A variety of upwinding methods have been proposed to overcome the problems with the convective terms. Details of the finite element method and its shortcomings are given in ***Hughes** (1987).*

1.3 Boundary Element Method (BEM)

A method or technique which has the accuracy of the analytical methods and preserves the versatility of the numerical techniques to solve complex reservoir problems is highly desirable. With high speed computing possible with modern computers, the geological information can be included in well testing models with good precision. Such models can prove to be quite useful.

A method which preserves these desirable features is the boundary element method (BEM). The early development of this technique took place within the realm of potential theory. A good description of the mathematical foundations of the boundary element method is given by *Jialin (1986).* The horizons of the method expanded tremendously during the 1970's in various disciplines of engineering such as elastodynamics, heat transfer, wave propagation, fluid mechanics and groundwater hydrology. Unlike the FDM and FEM which are domain methods, the boundary element method is a surface method. This means internal grids need not be generated and only the bounding surface of the system is discretized. Being a surface method the drawbacks of grid orientation effects are avoided as no internal grids are prescribed. The analytical character of the solution is preserved in terms of the Green's function of the governing differential equation. The BEM is an integral method developed from a weak formulation of the problem statement and so, the numerical dispersion is minimal because of the smoothing property of the integral operators.

A brief discussion of the pertinent literature on boundary element method is presented next. The application of BEM to steady state problems is addressed first. Subsequently the transient problems are discussed.

The application of the boundary element method to steady state problems governed by Laplace's equation are numerous. This is because the BEM is particularly suited for Laplace type (potential flow) equations due to the symmetry properties this equation possesses.

***Liggett** (1977)* solved the problem of locating the free surface in groundwater flow using the boundary element method. He assumed steady state flow and used an iterative technique to update the surface location until convergence was achieved. ***Liggett** and Liu (1979)* extended the abovementioned work and considered the nonlinearity of the motion of the free surface parametrically using finite differences in time to propagate the free surface. They compared their results with experimental data.

Based on the work by *Liggett (1977)*, *Liu **and** Liggett (1978)* solved two groundwater problems of sudden drawdown and free surface flow through a dam with a toe drain. They used a hodograph transformation to the complex plane to reduce the problem to a mixed type boundary value problem on a regular domain. *Lennon et **al** (1979)* presented the formulation and solution to axisymmetric potential flow problems in groundwater hydrology. This formulation was used to calculate the free surface in pumping wells. Parallel applications for steady state heat transfer and equivalent applications in elastodynamics are discussed by *Brebbia and Walker (1980)*. *Brebbia (19843)* showed the use of BEM for solving potential flow problems external to a boundary. *Lafe et al (1980)* discussed the treatment of singularities during steady flow. The extension of boundary element procedure to three dimensions was shown by ***Lennon** et **al** (1980)*. The essential difference between two and three dimensional applications of BEM is that the boundary is a surface in three dimensions instead of a line in two dimensions. Other problems for steady state flow including tilting interfaces in a piston-like displacement experiment were solved by ***Liu** et. al. (1981)*. They compared their results with experiments conducted with a Hele-Shaw cell. Application to petroleum reservoir engineering problems was done by ***Masukawa** and Horne (1988)* and *Numbere and Tiab (1988)* who used the boundary element method for streamline generation in balanced waterflood patterns.

*Rizzo and **Shippy** (1970)* were the first to address the problems in transient heat conduction by the boundary element method. They formulated the problem in Laplace space. The resulting solution was applied to the problem of heated cylinder with convective boundary conditions. *Shaw (1974)* solved the same problem with different and arbitrary boundary conditions on the curved surface of the cylinder. *Shaw (1974)* solved the problem in real space rather than in the transformed domain. The nature of the diffusion equation is such that the solution **is** history dependent. In other words, the solution at any given time depends on the solutions at all previous times. They calculated the solution at any given time by starting the computations at time zero.

Groundwater hydrology applications of BEM for transient flow in aquifers was first presented by *Liggett and Liu (1979)*. They compared both the real space and Laplace transform solutions. Also, the time stepping procedure for the real space formulation was based on finite differences in time unlike those of *Shaw (1974)*. *Liggett and Liu (1979)* showed some simple examples using these formulations. They considered a mixed problem defined on a square block of porous medium. The top and bottom surfaces of the medium were sealed. The inlet and outlet surfaces were held at constant values of head. In order to start the solution procedure for **this** problem, an asymptotic expression for the flux at short times had to be used. This is clearly a disadvantage of the formulation. *Wrobel and Brebbia (1981)* and *Pina (1985)* suggested different **types** of time stepping schemes. *Pina (1985)* showed that if the time step size were assumed to be constant then one can avoid recalculation of all the matrices corresponding to the previous time steps (because of the history dependence). *Taigbenu and Liggett (1985)* proposed time stepping schemes based on finite difference in time. One of their schemes was based on a multi-substep procedure which requires iterative solution. In another related paper, *Taigbenu and Liggett (1986)* performed a stability analysis based on the solution to a simple problem.

In addition to these problems, techniques to include multizonal reservoirs have been introduced for steady state problems *(Cheng, 1984, Masukawa and Horne, 1988)*. Pressure transient behavior for composite and sectionally homogeneous reservoirs have been treated by *Kikani and Horne (19893)*. *Wrobel et al (1986)* proposed a dual reciprocity boundary element formulation for transient heat conduction problems. They assumed that the time derivative in the equation can be approximated by specific functions which satisfy the Laplace's equation and hence are harmonic. *Aral and Tang (1988)* extended the idea of *Wrobel et al (1986)* and proposed a secondary reduction process for transient problems governed by the diffusivity equation. This process utilizes a secondary interpolation process for temporal derivatives using specific coordinate functions which are problem independent. *Shapiro and*

Andersson (1983) considered steady state fluid response through fractured rock. The fractures were represented discreetly and continuum equations were used. ***Ellsworth (1986)*** solved the problem of hydraulic response of three-dimensional sparsely fractured rock masses. He considered disk shaped fractures which are simply or multiply connected to the other fractures. Use of regular perturbation to solve steady state problems with linearly and exponentially varying permeabilities was presented by ***Lafe and Cheng (1987).*** ***Van Kruysdijk and Dullaert, (1989)*** used the boundary element method to solve for the transient pressure response of multiply fractured horizontal wells. The horizontal well was treated as a line source producing with a uniform rate and the individual fractures were assumed to be planar with an averaging in the vertical plane through a skin factor. One-dimensional flow along the fractures was considered.

This work investigates the application of the boundary element method to complex problems, including considerations of accuracy and computational efficiency. The application to streamline generation is shown in balanced injection patterns with unit mobility ratio. The major thrust however, is to solve complex pressure transient problems. Modeling pressure transient behavior in sectionally homogeneous and composite reservoirs is also investigated.

Four separate computer programs have been developed in the course of this work. One of them is capable of generating streamlines in complex reservoirs with multiple producers/injectors, and has the ability to include singularities whose analytical behavior may be known from hydrodynamics. The second program solves the transient problem governed by the diffusivity equation in real space. The third computer program implements the transient problem in Laplace space. This has wider capabilities in teims of efficiency of implementation as well as capability of including important effects such as wellbore storage and skin, dual-porosity behavior and rate decline. This program can also handle an internal boundary of arbitrary shape and boundary condition, with a few basic: modifications to the program. The last program solves the pressure transient problem in a two region composite reservoir with arbitrary inner boundary and external boundary shapes. Composite models are needed to analyze field injection and falloff tests in a variety of processes including steam injection, water flooding, in-situ combustion, and CO_2 injection. This model assumes that there **is** a sharp discontinuity in properties across the injection fluid front. Since the velocity of the front is calculated as a part of the solution procedure, the front can be tracked approximately.

A wide variety of problems have been solved using the above computer programs and the versatility of the methodology is demonstrated through the investigation **of** accuracy **and** ease of solving complex problems. Such questions **as** the consistency of solutions, the required number of nodes, and the best nodal arrangement were also investigated.

Chapter 2

MATHEMATICAL PRELIMINARIES

The dimensionless form of the diffusion equation for the Row of single phase fluids through a homogeneous porous medium is derived in this chapter. In addition, a unified review of numerical techniques in the general context of weighted residual methods is presented. Subsequently, a step **by** step procedure for the development of the boundary element formulation is discussed. **A** review of the Green's functions and numerical considerations in the solution procedure for the boundary element method **are** also explained.

2.1 Fluid Flow Equations

A continuum representation of a porous medium allows one to write macroscopic scale fluid flow equations based on physical laws. Under the assumption of the flow of a single phase fluid in **a** homogeneous porous medium, the continuity equation is

$$-\nabla \bullet (\rho\vec{v}) = \frac{\partial(\phi\rho)}{\partial t} + Q^* \tag{2.1}$$

where Q^* is the strength of a sink in mass per unit volume per unit time. The velocity field can be determined by **a** rate equation or momentum equation. Darcy's law, which can be derived under certain assumptions from the Navier-Stokes equation, can be used for the purpose. Darcy's law in a differential form is

$$\vec{V} = -\frac{K}{\mu} VP \tag{2.2}$$

where K is the permeability tensor. Substituting **Eq.** (2.2) into (2.1) gives

$$V \blacksquare(\frac{K\rho}{\mu} \nabla p) = \frac{\partial(\phi\rho)}{\partial t} + Q^* \tag{2.3}$$

Derivation of equations of **this** type can be found in **Aziz** ***and*** *Settari* (1979). A typical reservoir geometry is shown in Fig. (2.1). The domain of the probleim is defined by Ω and the corresponding bounding surface is represented by Γ. Any of the three linear boundary conditions may be applied at a section of the reservoir. We use the equation of state for small and constant compressibility fluid and assume that the permeability tensor (K) can be diagonalized. Defining a Cartesian coordinate system with the coordinate axes aligned with the principal permeability directions, we obtain the following equation :

$$\frac{\partial}{\partial x'}\left[\frac{k_{xx}}{\mu}\frac{\partial p}{\partial x'}\right] + \frac{\partial}{\partial y'}\left[\frac{k_{yy}}{\mu}\frac{\partial p}{\partial y'}\right] = \phi c\frac{\partial p}{\partial t} + \frac{Q^*}{\rho} \tag{2.4}$$

Assuming k_{xx} **and** $k_{y'}$ to **be** constants, in addition to the viscosity of the fluid and the porosity of the medium and performing a coordinate transformation given by

$$x = x' \quad ; \qquad y = y'\left[\frac{k_{xx}}{k_{yy}}\right]^{1/2} \tag{2.5}$$

we obtain

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \frac{1}{\eta}\frac{\partial p}{\partial t} + \frac{Q^*\mu}{\rho k_{xx}} \tag{2.6}$$

where

$$\eta = \frac{k_{xx}}{\phi\mu c} \tag{2.7}$$

Since the system geometry can be **odd,** the above system **is** normalized with respect to the area (A) of the system. Thus, defining

$$t_{DA} = \frac{k_{xx}\, t}{\phi\mu c A} = \frac{\eta t}{A} \; ; \qquad x_D = \frac{x}{\sqrt{A}} \; ; \qquad y_D = \frac{y}{\sqrt{A}} \; ; \qquad p_D = \frac{p_0 - p}{\tilde{p}} \tag{2.8}$$

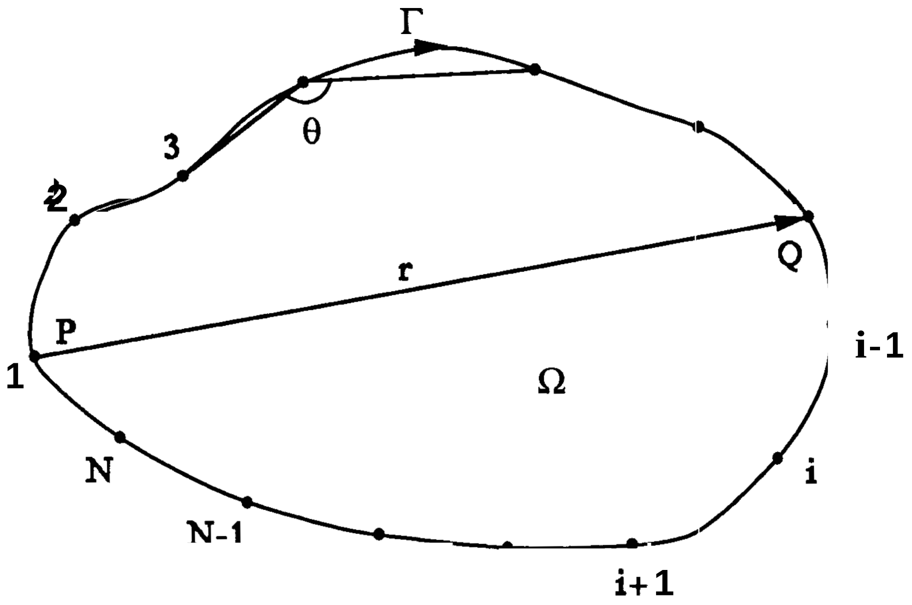


Fig. 2.1 A typical computational domain

gives

$$\frac{\partial^2 p_D}{\partial {x_D}^2} + \frac{\partial^2 p_D}{\partial {y_D}^2} = \frac{\partial p_D}{\partial t_{DA}} + Q_D \tag{2.9}$$

where

$$Q_D = \frac{Q^* \mu A}{\rho k_{xx} \tilde{p}} \tag{2.10}$$

and Q_D is dimensionless flow rate. $\tilde{p}$ is an arbitrary normalization pressure chosen to be equal to one and p_0 is the initial reservoir pressure. Rewriting Eq. (2.9) in a coordinate-system free format, that is in the operator form

$$\nabla^2 p_D = \frac{\partial p_D}{\partial t_{DA}} + Q_D \tag{2.11}$$

A linear operator L is defined such that

$$L(p_D) \equiv -\nabla^2 p_D + \frac{\partial p_D}{\partial t_{DA}} + Q_D = 0 \tag{2.12}$$

For fluid flow under steady state conditions the time derivative on the right hand side of Eq. (2.11) is zero. The governing equation is a Laplacian operator which is elliptic in nature. The velocity field is established instantaneously and does not change with time. For a finite time derivative in Eq. (2.11), the nature of the operator changes to parabolic **and** the problem becomes an initial-boundary value problem.

2.1.1 Boundary Conditions

Three types of boundary conditions can be imposed on different sections of the boundary, Γ.

$$p_D = p_1 \qquad on\ \Gamma_1\ \varepsilon\ \Gamma \qquad (Dirichlet) \tag{2.13a}$$

$$\frac{\partial p_D}{\partial n} = 41 \qquad on\ \Gamma_2\ \varepsilon\ \Gamma \qquad (Neumann) \tag{2.13b}$$

$$\alpha\, p_D + \beta \frac{\partial p_D}{\partial n} = \gamma \qquad on\ \Gamma_3\ \varepsilon\ \Gamma \qquad (Mixed) \tag{2.13c}$$

where n is the outward pointing normal on an element on the boundary. The initial condition is homogeneous. Symbolically

$$p_D(t_{DA} = 0) = 0 \tag{2.14}$$

2.2 Weighted Residual Method

It is informative to compare various numerical schemes from the point of view of the consistency of the method, its convergence properties and error estimates. This **task** becomes easier if common grounds for evaluating each of the numerical schemes can be established. If it can be shown, for example, that most of the numerical schemes **are** subsets of one formulation only, and that a particular numerical method **is** obtained by making certain assumptions in the general formulation, then the evaluation of a numerical method becomes simple. Such a formulation which unifies all the numerical schemes is known as the weighted residual method. Although, traditionally, none of the numerical methods are derived from the weighted residual formulation, it is another way of looking at the numerical methods. The idea here is to show an alternative viewpoint.

In the weighted residual approach, assumptions of particular functions for the unknown variables lead to different numerical schemes. The generality and the strength of each method can then **be** seen and compared on the same basis. The basis of the weighted residual approach is presented next. It is shown how some of the numerical methods are obtained from the weighted residual formulation. ***Brebbia*** (1984, *1984b)* give an overview of the weighted residual scheme, and show simple examples to illustrate the derivation of various numerical schemes.

Consider the solution **of** partial differential equations of the type

$$L(p_D) \equiv -\nabla^2 p_D + \frac{\partial p_D}{\partial t_{DA}} + Q_D = 0 \qquad in\ \ \Omega \tag{2.15}$$

given in **Eq.** (2.12), by the weighted residual scheme over the domain Ω. The external surface of the domain Ω is Γ and the exact solution to Eq. (2.15) is given by p_D. L is considered to be a linear operator for the purposes of this work.

The unknown variable p_D is approximated by a set of functions $\phi_k(x,t)$ such that

$$p_D = \sum_k B_k \phi_k \tag{2.16}$$

In **Eq.** (2.16), B_k are the undetermined parameters and ϕ_k ***are*** linearly independent functions of the independent variable, taken from a complete sequence of functions satisfying the admissibility conditions. These admissibility conditions are related to the essential boundary conditions and the degree of continuity of the functions. In essence, a particular form of the solution for **Eq.** (2.15), in terms of simple functions ϕ_k, has been assumed. For examp e, ϕ_k could be trigonometric *sine* and ***cosine*** functions.

Since **Eq.** (2.16) is an approximation of the true solution, substituting **Eq.** (2. 6) in (2.15) produces an error function ε' which is called the residual, symbolically

$$L(p_D) = 0 + \varepsilon' \tag{2.17}$$

When ε' is zero the solution is exact, but in general, ε' will be non-zero. One can choose to force **this** error to be zero in an average sense over the domain of the problem. The error ε' is forced to zero in an average sense by setting the weighted integral of the residual to zero

$$\int_\Omega \varepsilon' \psi_i \, dV = 0 \tag{2.18}$$

where ψ_i is a set of weighting functions. Physically **Eq.** (2.13) means that the error is orthogonalized with respect to a set of weighting functions. In other words, the inner product of the error with respect to the weighting functions is set to zem over the domain.

In order to relate the functions ϕ_k and ψ_k to physical variables, consider for example, the discussion about finite element method in Section 1.2.2. It was mentioned in Section 1.2.2 that two types of functions are assumed in the finite element method. The two sets of functions are ϕ_k and ψ_k which are shown in **Eqs.** (2.16) and (2.18) respectively. ϕ_k is known as the basis, trial or interpolation function and ψ_k is known as the weighting function.

Different numerical methods result on choosing a different set of weighting functions. When the set of weighting functions is a sequence $1, x, x^2, \ldots.$ the method that results is called the 'Method of Moments'. In the event that the error is forced to be zero at certain points in the domain, the resulting scheme is called the 'Point Collocation' technique. These points are usually, but not necessarily, evenly distributed in the domain. The Galerkin finite element method results if the sequence of weighting functions ψ_i is taken to be identical to ϕ_i. That is, the weighting functions are the same as trial functions.

All the above methods are restricted to self-adjoint operators (analogous to the symmetry of a matrix) and boundary conditions coinciding with the essential (Dirichlet type) boundary conditions. A simple but crude definition of a self-adjoint differential operator is one which consists of only even order derivatives. Existence of any odd derivative makes the operator adjoint (not self-adjoint). A more rigorous definition of a self-adjoint operator will be given later. More powerful techniques of solving partial differential equations emerge when the stipulation of self-adjointness and the restriction on the boundary conditions are relaxed. This can be achieved through weak or integral formulations of the differential operators. Both the finite element and the boundary element methods are based on the weak formulation. The essentials of the weak formulation and how this gives rise to the finite element and boundary element methods, is discussed next.

The weak formulations are based on the following considerations. The continuity requirements of the solution space are usually determined by the highest derivative in the differential operator. If some of these derivatives are transferred to the weighting function, by integrating by parts, the resulting method is known as the finite element method. Transfer of derivatives reduces the continuity requirement of the trial functions in the L_2 - sense. The L_2 - norm which defines the square integrability of a function over the domain of interest, is considered in the sub-space known as the Sobolev space. The Sobolev spaces are a subset of spaces which arise on imposing restrictions on the continuity and integrability requirements of the function and its derivatives. The continuity requirements are **an** important consideration, because they help one choose the mathematically feasible functions as approximations (the trial and weighting functions). For example, a C^o continuous function is simply a continuous function whereas a C^1 continuous function implies the continuity of both, the function and its first derivative. **A** C^1 continuity requirement thus restricts the trial or weighting function space from which a function can be chosen. Very interesting coinvergence properties can be proved for such weak formulations and with simple **basis** functions designed with respect to the above analysis, the results obtained are quite accurate ***(Hughes, 1987).***

It was seen in the previous paragraph that, when only a few derivatives are transferred to the weighting functions, the resulting method **is** the finite element method, but if the highest derivatives from the operator $L(p_D)$ are completely transferred to the weighting function (by integration by **parts),** sort **of** an 'inversion' process, **this** leads to the method known as the Boundary Integral Equation Method. Since **an** integral equation is difficult to solve, the boundary integral equations are implemented discretely on elements. Due to this discrete implementation, the boundary integral equation method is commonly known as boundary element method (BEM).

With the boundary element method weighting functions can be proposed which identically satisfy the goveming equations and only approximately the boundary conditions. In **this** regard, the boundary element method can be viewed as a special case of the weighted residual technique where the highest derivatives of the differential operator have been transferred to the weighting function. The details of the formulation and casting of the differential equation into **an** integral equation are discussed in the next section.

2.3 Boundary Element Method

Although the boundary element technique can be seen in the light of the weighted residual methods as shown in the previous section, the derivation that follows is based on more fundamental principles. Starting from the partial differential equation Eq. (2.12), an integral equation based on a boundary procedure is derived. The primary objective for the manipulation of Eq. (2.12) is to cast the governing differential equation into **an** exact differential. If the equation is exact, then a first integral of the differential equation is readily available. Rewriting Eq. (2.12)

$$L(p_D) \equiv -\nabla^2 p_D + \frac{\partial p_D}{\partial t_{DA}} + Q_D = 0 \tag{2.19}$$

Consider a function, ***G*,** which is in the same set of admissible functions **as** p_D. Admissible here means that G belongs to the same class of functions as p_D with regards to continuity **and** integrability. Taking a product of the function G with the differential operator $L(p_D)$ given by **Eq.** (2.19)

$$GL(p_D) = -G\nabla^2 p_D + G\frac{\partial p_D}{\partial t_{DA}} + GQ_D \tag{2.20}$$

Expressing Eq. (2.20) in a divergence form which effectively involves transferring derivatives to the function G. **As** a first step one of the derivatives from the ∇^2 operator and the only temporal derivative, is transferred over to G.

$$GL(p_D) = -[\nabla \bullet (G\nabla p) - \nabla G \bullet \nabla p_D] + [\frac{\partial(Gp_D)}{\partial t_{DA}} - p_D\frac{\partial G}{\partial t_{DA}}] + GQ_D \tag{2.21}$$

where the identity

$$\nabla \bullet \{x\,\vec{y}\} = x\nabla \bullet \vec{y} + (\nabla x) \bullet \vec{y} \tag{2.22}$$

has been used. The term x in Eq (2.22) is a scalar function and $\vec{y}$ is a vector function. Rewriting Eq. (2.22)

$$GL(p_D) \equiv -\nabla \bullet (G\nabla p_D) + \frac{\partial(Gp_D)}{\partial t_{DA}} + GQ_D + \nabla G \bullet \nabla p_D - p_D\frac{\partial G}{\partial t_{DA}} \tag{2.23}$$

At this point ∇p_D still appears in Eq. (2.21) hence, performing the operation of casting in divergence form again

$$GL(p_D) = -\nabla \bullet (G\nabla p_D) + \frac{\partial(Gp_D)}{\partial t_{DA}} + GQ_D + [\nabla \bullet (p_D\nabla G) - p_D\nabla^2 G] - p_D\frac{\partial G}{\partial t_{DA}} \tag{2.24}$$

Rewriting Eq. (2.24)

$$GL(p_D) = -\nabla \bullet [G\nabla p_D - p_D \nabla G] + \frac{\partial (Gp_D)}{\partial t_{DA}} + GQ_D - p_D[\nabla^2 G + \frac{\partial G}{\partial t_{DA}}] \tag{2.25}$$

The last term in Eq. (2.25) is identical to the diffusion equation without the source term and a positive sign on the temporal derivative instead of a negative sign. At this point all the highest derivatives from the original equation have been transfeirred to a function *G* which will be seen later to be a weighting function. The positive **sign** on the last term in Eq. (2.25) instead of a negative sign in the diffusion operator [Eq. (2.19)] is 'becausethe diffusion operator is not ***self-adjoint*** as it contains a first order (odd) derivative. More rigorously, an adjoint (or not self-adjoint) differential operator is similar to an adjoint of a matrix. ***An*** operator L is said to be ***self-adjoint*** **if** the inner product (G, Lp_D) is equal to (LG, p_D), where the inner product of two functions u and v is defined as

$$(u,v) = \int_\Omega uv \; d\Omega \tag{2.26}$$

The diffusion type operators are non self-adjoint and cannot be formally cast into a self-adjoint form. On the other hand for the steady state case in which the time derivative in Eq. (2.12) is zero, a Laplacian type operator results which is formally self-adjoint. Substituting In Eq. (2.25) an adjoint differential operator is defined as

$$L^*(G) \equiv -\nabla^2 G - \frac{\partial G}{\partial t_{DA}} \tag{2.27}$$

Substituting Eq. (2.27) into Eq. (2.25) and transposing the terms

$$GL(p_D) - p_D L^*(G) = -\nabla \bullet [G\nabla p_D - p_D \nabla G] + \frac{\partial (Gp_D)}{\partial t_{DA}} + GQ_D \tag{2.28}$$

Eq. (2.28) can be derived in an alternative way which is simple to follow. In the alternative derivation Eq. (2.26) is multiplied by p_D and subtracted from Eq. (2.20). Green's third identity is used to cast the equation in the form of Eq. (2.28).

Eq. (2.28) expresses the left hand side expression in terms of a divergence form. The right hand side of **Eq.** (2.28) is an exact differential, as we had set out to obtain. Integrating **Eq.** (2.28) over the domain of the problem results in

$$\int_{t_{DA}} d\tau \int_{\Omega} \left[GL(p_D) - p_D L^*(G) \right] dA \; = \; - \int_{t_{DA}} d\tau \int_{\Omega} \nabla \bullet (G \nabla p_D \; - p_D \nabla G) \; dA$$

$$+ \int_{t_{DA}} d\tau \int_{\Omega} \frac{\partial (G p_D)}{\partial t_{DA}} \, dA \; + \int_{t_{DA}} d\tau \int_{\Omega} (G Q_D) \; dA \qquad (2.29)$$

Eq. (2.29) holds on the entire space-time domain of the problem. Assuming that the function *G* is a response to an instantaneous line source of strength unity, we obtain on defining the adjoint problem in Cartesian coordinate system

$$L^*(G) \; \equiv \; -(G, \; + G_{yy} \; + G_{t}) \; = \; \delta(x - \xi) \; \delta(y \; - \zeta) \; \delta(t \; - \tau) \qquad (2.30)$$

where the subscripts denote partial derivatives and $\delta(z \; - z_{0)}$ is the Dirac - Delta function. The solution to **Eq.** (2.30) provides the function G, which is known as the fundamental solution or the Green's function for the operator $L(p_D)$. Substituting Eq. (2.30) in **Eq.** (2.29) and realizing that $L(p_D) \; = \; 0$ from **Eq.** (2.9)

$$- \int_{t_{DA}} d\tau \int_{\Omega} p_D(x_D, y_D, t_{DA}) \; \delta(x_D - \xi) \; \delta(y_D - \zeta) \; \delta(t_{DA} - \tau) \; dA \; = \; \int_{t_{DA}} d\tau \int_{\Omega} \nabla \bullet (G \nabla p_D$$

$$- p_D \nabla G) \; dA \; + \int_{t_{DA}} d\tau \int_{\Omega} \frac{\partial (G p_D)}{\partial t_{DA}} \, dA \; + \int_{t_{DA}} d\tau \int_{\Omega} (G Q_D) \; dA \qquad (2.31)$$

Using the shifting property of the delta function

$$\int \delta \; (x - \xi) \; \phi(x) \; dx \; = \; \phi(\xi) \qquad (2.32)$$

and the divergence theorem of Gauss in **Eq.** (2.31) we obtain, an equation for pressure at any location in the reservoir in terms of an integral over the boundary values **of** both pressure and fluxes. The pressure at any location within the reservoir is

$$p_D(\,\xi, \zeta, t_{DA}) \; = \; \int_{t_{DA}} d\tau \int_{\Gamma} (G \nabla p_D \; - p_D \nabla G) \bullet n \; dS + \int_{\Omega} G_0 \, p_{D0} \, dA + \int_{\Omega} G \; Q \; dA \qquad (2.33)$$

This is the integral equation we had set out to derive. At this point no approximations have been made. All that is required is that a function G be found which has smooth properties and decays uniformly in the **far** field. Eq. **(2.33)** shows that the free space Green's function appears as a weighting function in the integral equation.

p_{D0} represents the initial condition in the integral equation. For a homogeneous initial condition, the second term on the right hand side in Eq. (2.33) becomes zero. The third integral on the right hand side corresponds to the contribution due to the source/sink of strength Q_D. If we consider n_w continuous line sources or **sinks** starting at the initial time zero, and producing / injecting at locations $(x_{D\,i}, y_{D\,i})$ at strengths $Q_{D\,i}$ respectively, then Q_D in **Eq. (2.33)** can be replaced by $\sum_{i=1}^{n_w} Q_{D\,i}\ \delta(x_D - x_{D\,i})\ \delta(y_D - y_{D\,i})$. The integral equation then reduces to

$$p_D(\xi, \zeta, \tau) = \int_{t_{DA}} d\tau \int_{\Gamma} \left(G\frac{\partial p_D}{\partial n} - p_D\frac{\partial G}{\partial n}\right) dS + \sum_{i=1}^{n_w} G_i Q_{D\,i} \qquad \textbf{(2.34)}$$

where G_i is evaluated at the source locations $(x_{D\,i}, y_{D\,i})$. Again, the shifting property of the delta function has been used to transfer the point of application of Green's function to the source location i. In other words

$$\sum_{i=1}^{n_w} \int_{\Omega} G Q_{D\,i}\ \delta(x_D - x_{D\,i})\ \delta(y_D - y_{D\,i}) = \sum_{i=1}^{n_w} G_i\ Q_{D\,i} \qquad \textbf{(2.35)}$$

The integral equation Eq. **(2.34)** is **a** strict boundary integral and implies that the dimensionless pressure at any location in the reservoir (ξ, ζ) and at a **time** τ, is given by a boundary integral over the space **and** time domains. Eq. **(2.34)** forms the basis for the Boundary Element Method. Eq. **(2.34)** is of a lower dimension compared to the original problem. For the two-dimensional problems considered in this work, it becomes a contour integral around the boundary. Unlike the numerical schemes such as finite difference or finite element methods, the entire reservoir need not **be** discretized. It is only the surface of the reservoir volume that must be discretized. The reduction in dimensionality could save considerable computing effort.

In order to compute pressure within the reservoir, Eq. **(2.34)** requires that both the pressure and flux be known at every point on the boundary. In other words, the temporal or spatial variation of pressure at any point within the reservoir boundary requires one to know the variation of both the pressure (p_D) and the flux $(\frac{\partial p_D}{\partial n})$ on the boundary. However, either the pressure or the flux is prescribed on the boundary through the boundary conditions. That is, one of the two quantities (pressure or flux) is unknown on the boundary. Thus, the integral cannot be evaluated until the missing piece of information on the boundary is determined. It is to this end, that Eq. **(2.34)** becomes an integral equation as the unknowns are embedded in the boundary integrals. For example, if the reservoir boundary is at constant pressure then the flux across the boundary is unknown. The flux terms $(\frac{\partial p_D}{\partial n})$ in Eq. **(2.34)** appear within the integral only. This integral equation is known as a Fredholm integral equation of the first kind. On the other hand, if the reservoir boundary were closed, then the flux across the boundary is zero and the pressures are unknown. The terms containing pressure appear both inside and outside of the integral in Eq. **(2.34).** This integral equation is known as the Fredholm integral equation of the second kind.

Once the boundary unknowns are determined, Eq. **(2.34)** is used again to calculate the value of pressure at any interior location. Interior solutions may be calculated now, because the right hand side of **Eq. (2.34)** is fully determined. Pressure at an interior location is calculated **by** Eq. **(2.34)** but in order to calculate the flux at a point interior to the domain, derivative of Eq. **(2.34)** has to be taken.

2.3.1 Green's Functions

The variable G , used in the integral equation is a solution to the adjoint problem defined in Eq. **(2.30).** It is known as a kernel function, a Green's function, or a fundamental solution to the differential operator defined in **Eq.** (2.9). The Green's function is the response to an instantaneous source of Unit strength at a location in the flow field. The value of the function depends on both the location of the fictitious (also known as Green's function) source and the observation point. The Green's function also depends on the location of the boundaries. It is thus, a two point function with symmetry properties. That is, if ihe source and the observation locations are interchanged, the response remains the same.

For application to finite domains such as shown in Fig. (2.1), the Green's function is different for every shape and type of boundary condition. In addition it is very difficult to calculate the Green's function except for certain regular shapes. This limits the flexibility of the solution procedure. To avoid this problem, a free space Green's function is defined for the given adjoint differential operator. This function satisfies the boundary conditions at infinity. That **is,** the Green's function is bounded and decays uniformly in the far field. The free space Green's function is then used in Eq. (2.34) as a weighting function. Appendix A gives the derivations for the free space Green's functions for the problems discussed in later chapters.

The free space Green's function satisfies the material balance condition

$$\int_{\Omega} G \; dA \;=\; 2\,\pi \tag{2.36}$$

The value 2π is obtained because the source is completely surrounded by the domain of interest. For example, for the free space Green's function, an infinite reservoir is considered and the source is completely contained in it. Now, in order to solve **Eq.** (2.34) for the *unk*nown boundary conditions, the Green's function source (fictitious source) is moved to the boundary. On the boundary, the domain of interest subtends an 'internal angle θ as seen in Fig. 2.1. Thus, the angle 2π in the free space Green's function is replaced by the angle θ on the boundary. The final form of the integral equation (2.34) becomes

$$c\,\theta\, p_D(\,\xi,\,\zeta,\,\tau) \;=\; \int_{t_{DA}} d\tau \int_{\Gamma} (G\frac{\partial p_D}{\partial n} - p_D\,\frac{\partial G}{\partial n})\; dS \;+\; \sum_{i=1}^{n_w} G_i Q_{D\,i} \tag{2.37}$$

$$\theta = 2\pi \qquad \text{if } (x_D, y_D) \ \varepsilon \ \Omega \tag{2.37a}$$

$$\theta = \theta \qquad \text{if } (x_D, y_D) \ \varepsilon \ \Gamma \tag{2.37b}$$

where θ is the internal angle subtended between two adjacent elements. This is shown in Fig. (2.1). The value of the constant, c, is one or two depending upon ithe formulation used.

The integral equation obtained in Eq. (2.37) is impossible to solve analytically unless the behavior of p_D and $\frac{\partial p_D}{\partial n}$ is known on the boundary. One of these variables is known from the boundary condition but the other one **is** unknown, **as** discussed earlier. Thus, a trial or interpolation function, which is piecewise continuous, is assumed for the unknown variables. This leads to the numerical aspects of the procedure which are discussed next.

2.4 Numerical Considerations

The boundary integral equation obtained in Eq (2.37) can be solved explicitly for some special cases of infinite domains and it forms the basis for solution of partial differential equations by the Green's function method. Use of the Newman product theorem and superposition helps to find solutions for certain bounded systems with orthogonal boundaries ***(Carslaw and Jaegar, 1959).***

The boundary integral requires knowledge of either the pressure or the normal derivative of the pressure at **all** locations on the boundary. In a well-posed problem, the pressure, the normal derivative of pressure, or some combination of the two, is known at each point on the boundary. In order to calculate the solution at any location within the domain, it is imperative to first solve for the boundary unknowns. The boundary unknowns can be solved for by a collocation technique.

Implementation of the collocation technique requires discretization of both the space and time domains. The boundary of the computational (space) domain shown in Fig. 2.1 is discretized into N points referred to as nodes. The line segment joining two adjacent nodes will be referred to as an element. In this 'elemental' sense the Boundary Integral Equation Method (BIEM) is referred to as Boundary Element Method (BEM).

Equation *(2.37)* is solved for the unknown boundary condition by choosing a discrete number of elements on the boundary. An interpolating function is assumed **for** pressure and flux (or normal derivative of pressure) on a boundary element in both space and time dimensions. The integral equation can be solved by moving a fictitious source point (or Green's function source) to each of the nodes, in turn. One matrix equation is obtained for each position of the fictitious source. Moving the source to all the nodes generates enough equations so that they match the number of unknowns. The problem then reduces to one **of** solving a matrix equation. Once **both** pressure and normal derivative of pressure are known at all points on the boundary, **Eq.** *(2.37)* can be used again, to obtain the solution at any internal point (ξ, ζ, τ). Since an internal point is completely surrounded by the computational domain, the angle θ in Eq. *(2.37)* becomes 2π for calculations in the interior.

2.4.1 Interpolation Functions

Interpolation functions define the variation of the dependent variables between two nodes or equivalently over the element. Since only one variable [pressure or normal derivative of pressure or a combination of the two] is known on the boundary, the behavior of the unknown variable must be assumed in order to solve Eq. *(2.36)*. Various **types** of interpolation functions have been used. *Brebbia* (1984) discusses some of the commonly used functions. The two most commonly used interpolation functions are constarit elements and linear elements. These are described next.

For the constant elements the dependent variables are taken to be the average value of the nodal values and they act at the center of an element. Symbolically

$$p_D(\xi) = \frac{1}{2}(p_{D_i} + p_{D_{i+1}}) \qquad \xi_i < \xi < \xi_{i+1} \tag{2.38}$$

A similar function is defined for the flux. The advantage of using constant elements is its ease of implementation and use. Flow singularities due to sharp corners in the flow field are avoided because the center of gravity (point of action) of a constarit element is at the center of the element rather that at the edges (end points). In other words, the element value of the dependent variable is the average of the end point values of the two nodes constituting the element. Use of constant elements for time domain interpolation is common.

The linear interpolation function represents a continuously varying dependent variable on the element. The pressure and normal derivative of pressure at any point on the element is expressed in terms of the nodal values as follows :

$$p_D(\xi) = \left[(p_{D_{j+1}} - p_{D_j})\,\xi + (\xi_{j+1}p_{D_j} - \xi_j p_{D_{j+1}})\right]/(\xi_{j+1} - \xi_j) \qquad \xi_j < \xi < \xi_{j+1} \tag{2.39}$$

$$\begin{aligned}(\frac{\partial p_D}{\partial n})(\xi) &= \left\{\left[(\frac{\partial p_D}{\partial n})_{j+1} - (\frac{\partial p_D}{\partial n})_j\right]\xi + \left[\xi_{j+1}(\frac{\partial p_D}{\partial n})_j\right.\right. \\ &\quad \left.\left. - \xi_j(\frac{\partial p_D}{\partial n})_{j+1}\right]\right\} / \ (\xi_{j+1} - \xi_j) \qquad \xi_j < \xi < \xi_{j+1}\end{aligned} \tag{2.40}$$

where ξ is the local coordinate varying along the element. **Eq.** (2.38) and (2.39) imply that the pressure and flux at any point on an element which extends from the point ξ_j to ξ_{j+1} is a linear interpolation of its end point nodal values.

Throughout this work linear interpolation functions in space and constant elements in time have been used. Higher order functions can be used at the cost of complexity in implementation and numerical evaluation of resulting integrals. The extra time and cost produces only a small improvement in accuracy.

2.4.2 Local Coordinate System

The contour integral of **Eq.** (2.37) is still a double integral in space dimension. To avoid having to evaluate double integrals, a local coordinate system based on the fictitious source point and the element over which the integral is performed is defined. Thus, the coordinate system moves depending on which node is the source point and which element is the field element. An illustration of a local coordinate system is given in Fig. 2.2. In Fig. 2.2, the point p_i is the node at which the fictitious source point is acting and the element over which the contribution of the source is being calculated consists of nodes p_j and p_{j+1}. This element is called the field element, and a point on this element is referred to as a field point.

To define a local coordinate system, a perpendicular is drawn from the fictitious source point on the boundary onto the element over which the integral has to be performed. The perpendicular forms one of the axes of the local moving coordinate system (ζ) and is constant on the element. The other axis is along the element (ξ). The local coordinate system is different from the fixed global coordinate system in which the problem has been defined.

2.4.3 Matrix Equations

Using the interpolation functions presented in Section 2.4.2, the integral of Eq. (2.37) can be performed piecewise on each element. Equations (2.39) and (2.40) are substituted in Eq. (2.37) along with a transformation to a moving coordinate system. As mentioned earlier, the source point is moved around the boundary and is located at each of the nodes in turn. This generates N equations in N unknowns, where N is the number of boundary nodes. A geometry dependent coefficient matrix is thus obtained. The form of the matrix equation is

$$H^{\alpha\beta} u_\beta = G_\beta \tag{2.41}$$

where a and β denote an element in the matrix or location of the matrix in an array of matrices, depending on the formulation used. The vector of unknowns is represented by u and G_β represents the right hand side vector. Details of the matrix structure and the elements of the matrix will be described in greater detail in Chapters 3 and 4.

The matrix $H^{\alpha\beta}$ is full, that is there are no elements which are zero. The reason for this is explained by comparing the matrix in Eq. (2.41) with the matrices obtained in the finite element method,

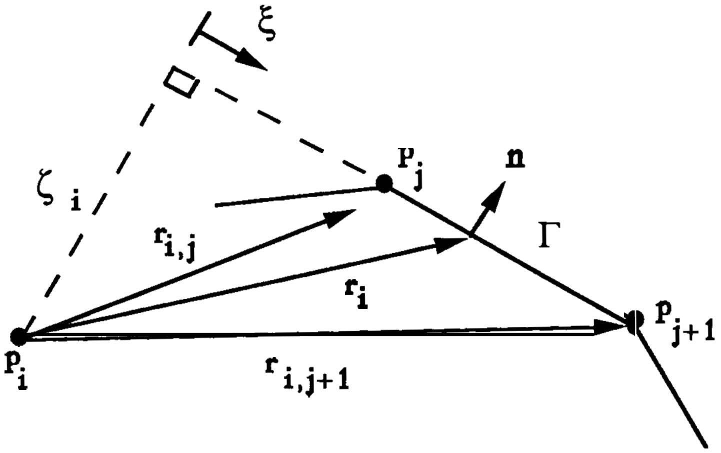


Fig. 2.2 Local coordinate system

In finite element methods the basis functions or weighting functions are usually chosen to be either isoparametric or piecewise polynomials. These functions have a compact local support *(Section* 2.2). That is, the functions have a finite value over an element but are zero over all the other elements. This leads to the well known sparse matrix structure. In the boundary element method the weighting functions are solutions to the adjoint problem and do not have to be chosen arbitrarily. This translates to the fact that they have global support. Owing to the global support, the contribution at one location is affected by all points in the discretized domain. Thus, the matrices that are obtained in the boundary element method are full, unlike the sparse structure of other numerical schemes. One redeeming feature is that the size of the matrices are much smaller compared to the domain type schemes. This is due to the boundary only character of the boundary element method.

2.5 Numerical Integration

The elements of the coefficient matrix consist of integrals over boundary elements. These integrals depend on the type of interpolation functions used and the complexity of the weighting functions. For development of a general purpose simulator where any type of interpolation function can be prescribed, the integrals have to be evaluated numerically. For this work, most of the integrals could be evaluated analytically. Nevertheless, some integrals could not be evaluated analytically and numerical means had to be used. The Green's function has a singularity at the origin. Other than that singularity, the Green's functions, used in the integral equation (2.37), are well behaved functions which decay rapidly in the far field. Highly accurate Gauss-Legendre type integration schemes were used to evaluate the integrals numerically. Gauss-Legendre quadrature has been shown to be optimum in the sense of maximum accuracy for a given number of function evaluations (*Stroud and Secrest, 1966).* The range of integration is divided into two panels and a six point integration scheme is used on each of the panels. This scheme was found in general to be the optimum between computing effort and accuracy. The accuracy criterion used on test cases was to eight significant digits.

A smaller number of function evaluations (four, for example) works well if the criterion for accuracy is relaxed slightly. The Gauss-Legendre quadrature requires specification of abscissae and the weights given to each of the abscissae points. The automatic generation of the abscissae and weights for the Gauss-Legendre quadrature for an N-point rule is provided in Appendix - F.

2.5.1 Singular Integral Evaluation

When the fictitious source point is on the same boundary element as the field point, integrals which are singular at one of the limits, are obtained. The integrals are singular due to the presence of a singularity in the Green's function at the origin. In Fig. 2.2, when the fictitious source point (p_i) is acting at p, or p_{j+1} the value of the local coordinate ζ is zero and ξ_j or ξ_{j+1} is also zero. The usual techniques for numerical integration are unsuitable. In, particular the integral

$$I_2 = \int_0^c E_1 (\frac{x^2}{4t})\, dx \tag{2.42}$$

has been performed in a very tedious way in the literature *(Liggett and Liu, 1979, Taigbenu and Liggett, 1985, Pina, 1985*). Various methods have been suggested to handle this, but all of them require either removing the singularity and integrating numerically or performing a term by term integration of an infinite series which converges slowly at large values of the argument. Eq. (2.42) can be evaluated in terms of smoothly behaved functions. The result is

$$I_2 = cE_1(\frac{c^2}{4t}) + \sqrt{4\pi t}\;\; erf\,(\frac{c}{\sqrt{4t}}) \tag{2.43}$$

and is derived in Appendix C. The functions in Eq. (2.43) are standard special functions and can be evaluated easily and quickly.

The mathematical development of the boundary element method was discussed in this chapter. In the next chapter the development of the boundary element method for steady state problems (Laplacian operator) is presented.

Chapter 3

STEADY STATE PROBLEMS

Steady state flow occurs in balanced waterflood schemes or at late times in bounded systems with constant pressure outer boundaries. The effect of pattern geometry and flow singularities **are** important because the reservoir configuration and sources/sinks govern the movement of injected fluid to the production wells. Though, the flow singularities affect the local flow field more strongly than other locations, ignoring them could lead to undesirable and unrealistic results. **A** number of studies involving analog and computer models have been conducted on the effect of various parameters. **It** is assumed that the reservoir fluid is incompressible and is being displaced by a similar fluid. The fluid and reservoir properties are homogeneous. Capillary and gravity effects are neglected.

As mentioned earlier, the conventional numerical schemes have difficulty in tracking moving fronts due to the nature of domain gridding whereas the **boundary** element method **is** most suited for potential flow applications. Examples are shown in this chapter which demonstrate the utility of Boundary Element calculations in a number of typical cases for unit mobility ratio displacement problems.

3.1 Implementation

The goveming equation for steady state flow becomes

$$\nabla^2 p_D = 0 \tag{3.1}$$

considering no sources/sinks present in the domain. The time dependent term in **Eq. (2.12)** is absent from **Eq. (3.1).** The operator of **Eq. (3.1)** is the Laplacian operator and is self-adjoint. This means that the adjoint operator which is solved with a concentrated (delta-function) source to give the Green's function is the same as the Laplacian operator. Following the derivation outlined in Chapter **2** the integral equation for pressure is

$$p_D\ (\xi, \zeta) = \int_{\Gamma} (G\frac{\partial p_D}{\partial n} - p_D\frac{\partial G}{\partial n})\ dS \tag{3.2}$$

The time integral is not present in **Eq. (3.2)** because of the steady Bow conditions. The effort required to compute the single integral around the contour is far less compared to the double integrals which arise in unsteady state problems. The treatment of unsteady problems will be shown later.

The weighting function, G, in **Eq. (3.2)** is the free space Green's function for the Laplacian operator and is

$$G(x_D, y_D, \xi, \zeta) = \frac{1}{2\pi}\ln r_D \tag{3.3}$$

where

$$r_D^2 = (x_D - \xi)^2 + (y_D - \zeta)^2 \tag{3.4}$$

Eq. (3.2) can be written in a discrete form over the element nodes on the boundary between ξ_j and ξ_{j+1} in the local coordinate system as

$$\theta_i\ p_{D_i} = \sum_{j=1}^{N} \int_{\xi_j}^{\xi_{j+1}} (G\nabla p_D - p_D\nabla G) \bullet n\ d\xi \qquad i = 1, N \tag{3.5}$$

Eq. (3.5) suggests that the dimensionless pressure at a location in the reservoir is given by the sum of the contributions of each of the boundary elements. The angle θ_i in Eq. (3.5) is given by

$$\theta_i = 2\pi \qquad \text{if } (x_D, y_D) \in \Omega \tag{3.5a}$$

$$\theta_i = \theta \qquad \text{if } (x_D, y_D) \in \Gamma \tag{3.5b}$$

If the fictitious source is completely surrounded by the reservoir boundary, θ_i is equal to 2π. If the source is on the boundary, θ_i is equal to the internal angle between two adjacent elements.

An important step comes when the fictitious (or the Green's function) source is moved to the boundary and the contribution of this source is evaluated at each of the nodal points on the boundary. As mentioned earlier, one of the quantities under the integral is unknown and the only way to solve Eq. (3.5) for any interior point is to first determine the boundary unknowns. It is at this step that Eq. (3.5) is an integral equation. The fictitious source point is moved to each of the boundary nodes in turn. Eq. (3.5) then gives the pressure at a boundary node in terms of the pressure and fluxes at other boundary nodes. This gives N equations in as many unknowns on the boundary. Using the interpolation functions defined in Eq. (2.39) and (2.40) and the free space Green's function defined in Eq. ('3.2) the discretized integral equation for pressure at a nodal location i in a local coordinate system is

$$\theta_i\, p_{D_i} = \sum_j \left[\frac{1}{(\xi_{j+1} - \xi_j)} \left\{ (p_{D_{j+1}} - p_{D_j}) \int_{\xi_j}^{\xi_{j+1}} \xi \ln r_{D_i}\, d\xi - (\xi_{j+1}\, p_{D_j} - p_{D_{j+1}}\, \xi_j) \int_{\xi_j}^{\xi_{j+1}} \ln r_{D_i}\, d\xi \right.\right.$$

$$\left.\left. + (p_{Dn_{j+1}} - p_{Dn_j}) \int_{\xi_j}^{\xi_{j+1}} \xi \frac{1}{r_{D_i}} \frac{\partial r_{D_i}}{\partial n} d\xi + (\xi_{j+1}\, p_{Dn_j} - p_{Dn_{j+1}}\, \xi_j) \int_{\xi_j}^{\xi_{j+1}} \frac{1}{r_{D_i}} \frac{\partial r_{D_i}}{\partial n} d\xi \right\} \right] \tag{3.6}$$

where, $(\frac{\partial p_D}{\partial n})_j$ is replaced by $p_{D_{n_j}}$ for convenience. The relation tetween the local (ξ, ζ) and the global (x, y) coordinates is evident from Fig. 2.2, and the following equations hold

$$r_{D_i}{}^2 = \xi^2 + \zeta_i^2 \tag{3.7}$$

and

$$\frac{\partial r_{D_i}}{\partial n} = \frac{\zeta}{r_{D_i}} \tag{3.8}$$

The integrals in **Eq.** (3.6) are given in Appendix **B.** Rewriting Eq. (3.6) in matrix notation

$$\sum_{i} \left[F_{ij} - \delta_{ij}\, \theta_i \right] p_{D_j} = \sum_{j} L_{ij}\, p_{Dn_j} \qquad i = 1, N \tag{3.9}$$

where

$$F_{ij} = \left[\frac{\zeta_i}{2} \ln\left[\frac{\xi_{j+1}^2 + \zeta_i^2}{\xi_j^2 + \zeta_i^2} \right] - \xi_{j+1}\left\{ \tan^{-1}\frac{\xi_{j+1}}{\zeta_i} - \tan^{-1}\frac{\xi_j}{\zeta_i} \right\} \right] / (\xi_{j+1} - \xi_j) \tag{3.9a}$$

and

$$\begin{aligned} L_{ij} = \Bigg[& \frac{1}{4}(\xi_{j+1}^2 + \zeta_i^2)\left[\ln(\xi_{j+1}^2 + \zeta_i^2) - 1 \right] - \frac{1}{4}(\xi_j^2 + \zeta_i^2)\left[\ln(\xi_j^2 + \zeta_i^2) - 1 \right] \\ & - \frac{\xi_{j+1}}{2}\Bigg\{ \left[\xi_{j+1} \ln (\xi_{j+1}^2 + \zeta_i^2) - 2\xi_{j+1} + 2\zeta_i \tan^{-1}\frac{\xi_{j+1}}{\zeta_i} \right] - \\ & \left[\xi_j \ln(\xi_j^2 + \zeta_i^2) - 2\xi_j + 2\zeta_i \tan^{-1}\frac{\xi_j}{\zeta_i} \right] \Bigg\} \Bigg] / (\xi_{j+1} - \xi_j) \end{aligned} \tag{3.9b}$$

The boundary conditions in **Eq.** (2.13a, b, c) are used in **Eq.** (3.9). These boundary conditions are multiplied by the proper coefficients **of** the matrix in **Eq.** (3.9). Separating the vector of unknowns from the vector of knowns results in a matrix equation of the form

$$H_{ij}\, \phi_j = G_j \qquad i = 1, N \tag{3.10}$$

where ϕ_j is the vector of unknowns and G_j is the right hand **side** vector obtained by using the boundary conditions.

3.1.1 Interior Solutions

Once the boundary unknowns are obtained, pressure at any internal location is calculated by using **Eq. (3.6)** again. **For** front tracking purposes, velocities at internal points must be calculated. The velocity of a point within the domain is

$$v^2 = v_x^2 + v_y^2 \qquad (3.11a)$$

$$\alpha = \tan^{-1}\frac{v_y}{v_x} \qquad (3.11b)$$

The velocities in the x and y directions are obtained by taking the directional derivatives of the integral equation **(3.6).**

$$v_x = -\frac{k}{\mu}\frac{\partial p}{\partial x} = -\frac{k}{\mu}\frac{1}{2\pi}\sum_j\left[\frac{\partial}{\partial x}\int_{\xi_j}^{\xi_{j+1}}(\ln r_{D_i}\frac{\partial p_D}{\partial n} - p_D\frac{1}{r_{D_i}}\frac{\partial r_{D_i}}{\partial n})\,d\xi\right] \qquad (3.12)$$

where, r_D is the distance from the internal point to a boundary point. **All** the integrals in Eq. (3.12) can be evaluated exactly since, both the flux and the pressure at each point on the boundary are known. The evaluation **of** the integrals and the resulting expression for the velocities is given in Appendix **B.**

3.2 Treatment of Singularities

Singularities in potential flow situations occur when the flux or the pressure at a certain location goes to a mathematical infinity. Sources and **sinks** with diminishing radii are examples of such singularities. Also sharp bends and comers in the flow geometry cause the flow behavior to become singular in the vicinity of the singularities. 'These singularities can be treated in two different ways. Well singularities can be treated as inhomogeneities in the flow equation as shown in Eq. (2.19) and **(2.37).** Being Dirac-delta functions for line source conditions, they appear only as an additive term as in **Eq.** (2.37). Another way to treat the singularities is **by** singularity programming, which is discussed next.

3.2.1 Singularity Programming

If the behavior of the singularity is known from hydrodynamics or elsewhere, then the contribution due to the singularity is subtracted from the goveming equation and the boundary conditions. The singularity free problem is solved with the conventional boundary element procedure. Once the solution is obtained, the contribution of the singularity is added. This simple procedure improves the accuracy of solutions greatly. This procedure can be represented as

$$p_D = p_{D_s} + p_{D_{ns}} \qquad (3.13)$$

$$\Longrightarrow \qquad p_{D_{ns}} = p_D - p_{D_s} \qquad (3.14)$$

where, $p_{D_{ns}}$ is the non-singular part of the solution and p_{D_s} is the singular solution. For a line source well singularity, p_{D_s} is given by

$$p_{D_s} = \sum_{i=1}^{n_w} \frac{Q_i}{2\pi} \ln r_{D_i} \qquad (3.15)$$

Both the singular and the non-singular solutions satisfy Laplace’:; equation. The boundary conditions should be adjusted for the singular behavior as follows,

$$p_{D_{ns}\Gamma_1} = p_{D\Gamma_1} - \sum_{i=1}^{n_w} \frac{Q_i}{2\pi} \ln r_{D_i} \qquad (3.16)$$

$$\left[\frac{\partial p_D}{\partial n}\right]_{ns\Gamma_2} = (\frac{\partial p_D}{\partial n})_{\Gamma_2} - \sum_{i=1}^{n_w} \frac{Q_i}{2\pi} \frac{1}{r_{D_i}} \left[\frac{\partial r_{D_i}}{\partial n}\right] \qquad (3.17)$$

Other known singularities can be treated in a similar manner. The effect of singularities on the potential and the velocity fields depends on the strength of the singularities and the distance from the points at which the contribution is being calculated. For weak singularities, the far field solutions are not affected strongly. Only in the vicinity of the singular points is the effect felt. Sometimes weak singularities can be treated either by approximating the flow field, or by not providing special elements and using the interpolation function used in the other parts of the flow field.

An example of approximation of the flow field such that the singularity is avoided is given in Fig. 3.1. The problem considered is the steady irrotational flow of fluid in a comer. Each side of the square is of length π. The top boundary is maintained at a dimensionless pressure of ten and the adjacent right side boundary at a pressun: of five. The upper right hand comer is discontinuous in pressure **i.e.,** the pressure changes From ten to five at a single point, thus the flux at that location goes to a mathematical infinity. Fig. 3.2 shows a diagram (not to scale) of a modified boundary where the upper right hand comer is chopped slightly (1 % of the total length) and is replaced by a no flow boundary. **This** provides the fluid enough non-equipotential surface so that the pressure gradient is established in a finite way. Table 3.1 shows the comparison of the boundary solutions with the analytical solution at various locations. The analytical solution is given in ***Numbere*** *(1982).* Also investigated is the effect of the number of nodal points on the results. Each side of the square boundary was divided into three elements (4 nodes) for the case with 13 nodal points. The nodes are equidistant from one another. Each of the elements was further subdivided into two for the case with 25 nodal points. In **both** cases, the chopped comer consisted of only two nodes at each of the ends. The maximum error is less **than** 1% for both 13 and 25 boundary nodes. Element refinement with 25 nodes gives accurate results, the maximum percent error being 0.0612%. The maximum error for the case with 25 nodes is near the chopped element (coordinates 3.0, 3.0), since the effect of singularity is felt more strongly in the vicinity of its occurrence. **As** no further subdivision of the chopped element was made when refining the elements from 13 to 25 nodes, the errors at this location for both the cases are the same. This is because the effect of the singularity overpowers the effect of nodal subdivision in its vicinity. **As** the points at which the solution is evaluated get closer to one of the boundaries, the **results** with 25 nodes **are** better than those of 13 nodes. This is seen from the results at nodal locations (3.0, 1.571) and (3.0, 0.5).

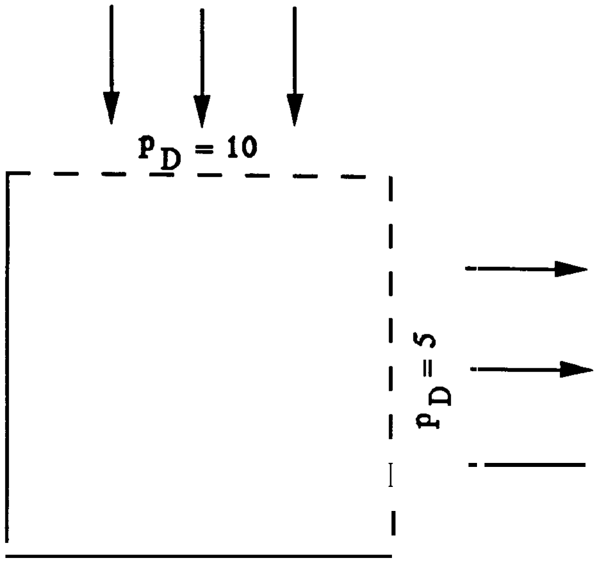


Fig. 3.1 : Steady irrotational flow in a corner

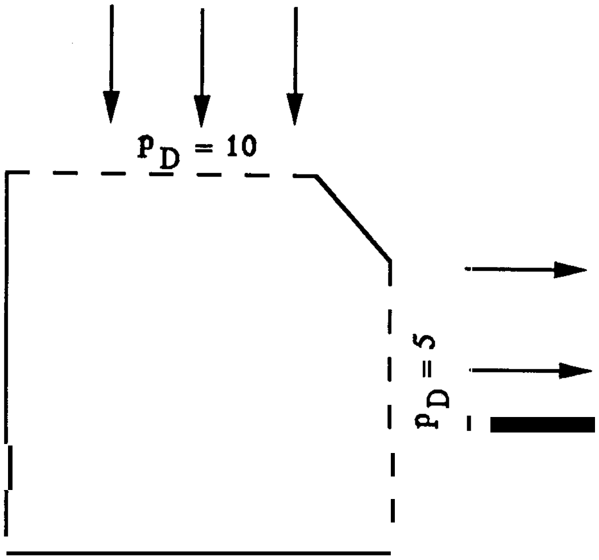


Fig. 3.2 : Modified boundary for steady irrotational flow

Table 3.1 Effect of Element Refinement on BEM Soln.

Y-Coord.	Y-coord.	Analytical Soln.	BEM			
			13 Nodes		25 Nodes	
			Soh.	% Error	Soh.	% Error
1.000	1.000	7.50076	7.5000	0.0101	7.5000	0.0101
1.500	1.500	7.50080	7.5000	0.0107	7.5000	0.0107
2.000	2.000	7.50060	7.5000	0.0080	7.5000	0.0080
2.500	2.500	7.50089	7.5000	0.0118	7.5000	0.0118
3.000	3.000	7.50459	7.5000	0.06 12	7.5000	0.0612
1.000	1.571	7.91911	7.9204	0.0163	7.9190	0.0014
1.500	1.57 1	7.56539	7.5648	0.0078	7.5648	0.0078
2.000	1.571	7.021 17	7.0212	0.0004	7.0205	0.0096
2.500	1.571	6.25481	6.261 1	0.1006	6.2543	0.0081
3.000	1.571	5.29154	5.3416	0.9461	5.2908	0.0139
1.000	0.500	7.29231	7.2877	0.0632	7.2905	0.0248
1.500	0.500	6.94476	6.9372	0.1089	6.9426	0.03 11
2.000	0.500	6.46369	6.4575	0.0957	6.4618	0.0292
2.500	0.500	5.86787	5.8620	0.1001	5.8667	0.0200
3.000	0.500	5.19627	5.1850	0.2 169	5.1961	0.0033

A comparison is made in Table **3.2** between the analytical solution, solution with constant boundary elements and solution with linear elements. 'With constant elements, the location of the singularity is avoided because the pressure or flux is evaluated at the midpoint of an element. Thus, the solution procedure does not know about the presence of a comer. In fact, for this reason the flow field need not be approximated by chopping off the comer, as in this example. The results for constant elements were given by ***Numbere*** (2982). The result with linear elements is far superior to constant elements as seen in Table **3.2.**

Figure **3.3** is a schematic of flow through a rectangular porous medium with a 90° wedge in the flow field ***(Liggett and Liu, 1983*)**.** The origin of the Cartesian coordinate system is at the tip of the wedge, which is at the center of the geometry. The boundary nodes are shown by solid circles in Fig. **3.3.** The behavior in the vicinity of this singularity is known from theoretical hydrodynamics to be

$$p_D \approx r^{\frac{2}{3}} \qquad \textbf{(3.18)}$$

This behavior can be incorporated in the boundary element procedure, in the neighborhood of the singular point by defining special elements around the singularity. Instead of linear variation of pressure and flux, used for the regular elements, the variation, in the vicinity of the singularity point is defined using Eq. **(3.18).** Table **3.3** show!; the dimensionless pressure and velocity at the boundary nodes with and without considering the wedge singularity with special elements. The errors due to the use of regular elements are small. This can also be seen in Table **3.4** which compares the boundary element solution at selected boundary points with the exact solution. The improvement is not great because the singularity is relatively weak, and the linear boundary elements match the results reasonably well.

3.3 Streamline Generation and Front Tracking

Since sources and sinks are simple to include and the handling of complex geometries is simplified, the boundary element method can be used effectively for generating streamlines in

Table 3.2 Steady Irrotational Flow in a Corner

Elements	Node Points	Max. % Error
Constant	12	7.9
Linear	13	**0.94**
Linear	*25*	0.061

* Numbere(1982)

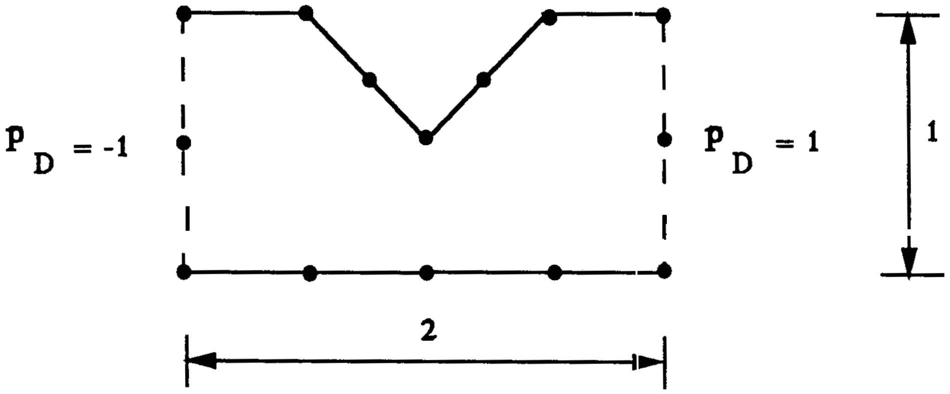


Fig. 3.3 : Schematic of flow across a 90° wedge

Table 3.3 Boundary Solutions for a 90° 'Wedge

X-Coord.	Y-coord.	Regular Elements		Singularity Prog.	
		p_D	**Flux**	p_D	**Flux**
-1.00	-0.50	-1.000	-0.892	-1.000	-0.876
-1.00	+0.00	-1.000	-0.796	-1.000	-0.778
-1.00	+0.50	-1.000	-0.655	-1.000	-0.635
-0.50	+0.50	-0.720	+0.000	-0.728	+0.000
-0.25	+0.25	-0.505	+0.000	-0.520	+0.000
+0.00	+0.00	-0.000	+0.000	+0.000	+0.000
+0.25	+0.25	-0.505	+0.000	+0.520	+0.000
+0.50	+0.50	-0.720	+0.000	+0.728	+0.000
+1.00	+0.50	+1.000	+0.655	+1.000	+0.635
+1.00	+0.00	+1.000	+0.796	+1.000	+0.778
+1.00	-0.50	+1.000	+0.892	+1.000	+0.876
+0.25	-0.50	+0.282	+0.000	4.288	+0.000
+0.00	-0.50	+0.000	+0.000	+0.000	+0.000
-0.25	-0.50	-0.282	+0.000	-0.288	+0.000

X-Coord.	Y-Coord.	p_D w/o Singular El.	p_D with Singular El.	Exact
+0.00	+0.00	-0.000	+0.000	+0.000
+0.25	+0.25	**-0.505**	+0.520	+0.521
+0.50	+0.50	*-0,720*	+0.728	+0.729

multiwell complex geometries. The method is fast and accurate because the dimensionality of the problem is reduced by one.

There are two different ways of computing the streamlines. After computing the pressure at an interior point, the velocity of the point is determined from Eqs. (3.11a) and (3.11b). The particle is then moved in time in a particle tracking sense. The location of a particle at a time $t + At$ is given by

$$x_{t+\Delta t} = x_t + v_x \Delta t \tag{3.19}$$

and

$$y_{t+AI} = y_t + v_y \Delta t \tag{3.20}$$

This implies that once the velocity field is computed at an interior **point,** the particle can be tracked to the producing well by using Eq. (3.19) and (3.20). In other words the coefficient matrix in Eq. (3.10) must be solved only once.

Another way **is** to pose the problem in the stream-function format instead of using pressure. The goveming equation remains the same; only the boundary conditions are modified. The stream function solution at any interior location gives the velocity field directly, although the pressure field must be calculated by integrating the velocity field. In **this** work, the velocity field is computed in a particle tracking scheme using the pressure solution as shown in Eq. (3.12).

Figure 3.4 shows the streamlines for the symmetric element (one-eighth) of a five-spot pattern at breakthrough. Sixteen points were selected radially around the injection well. Each of these particles, as they are referred to, were tracked until the first one reached the production well. The particles were tracked using **Eqs.** (3.19) and (3.20) once the velocity field was determined by Eq. (3.12). The results compare well with the analytical solution for the breakthrough sweep efficiency presented by *Morel-Seytoux* ***(1965).*** Figure 3.5 depicts the isochrones or the location of fluid particles at particular times after the start of injection for the symmetric element of a five-spot pattern. This is essentially the location of the front. Figures

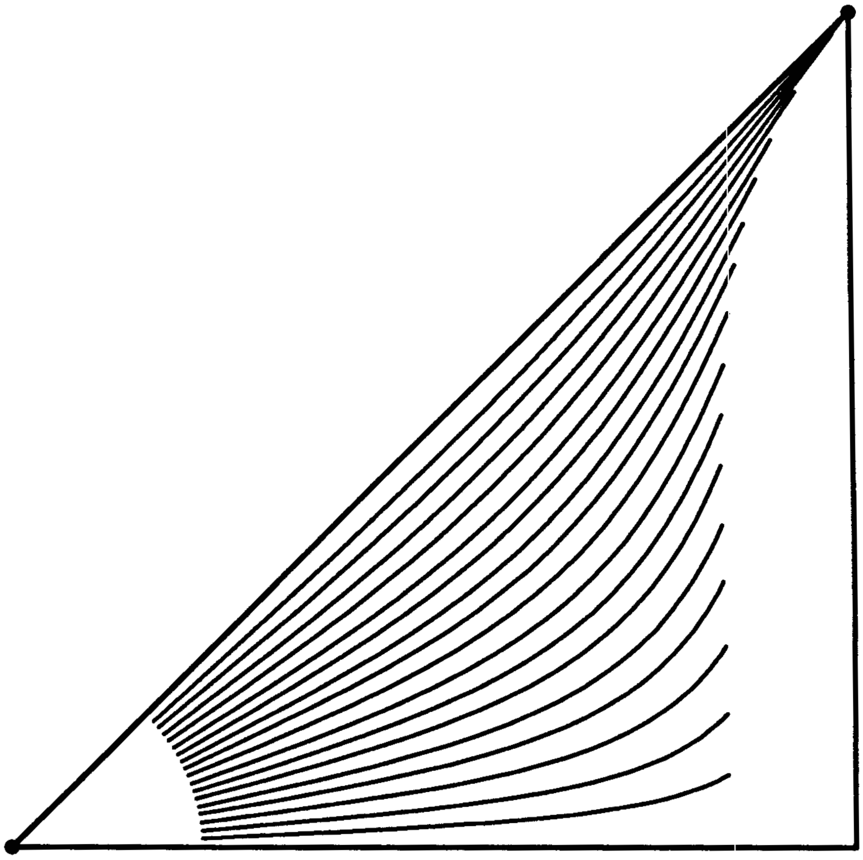


Figure **3.4** Streamlines at breakthrough in a 5-spot pattern

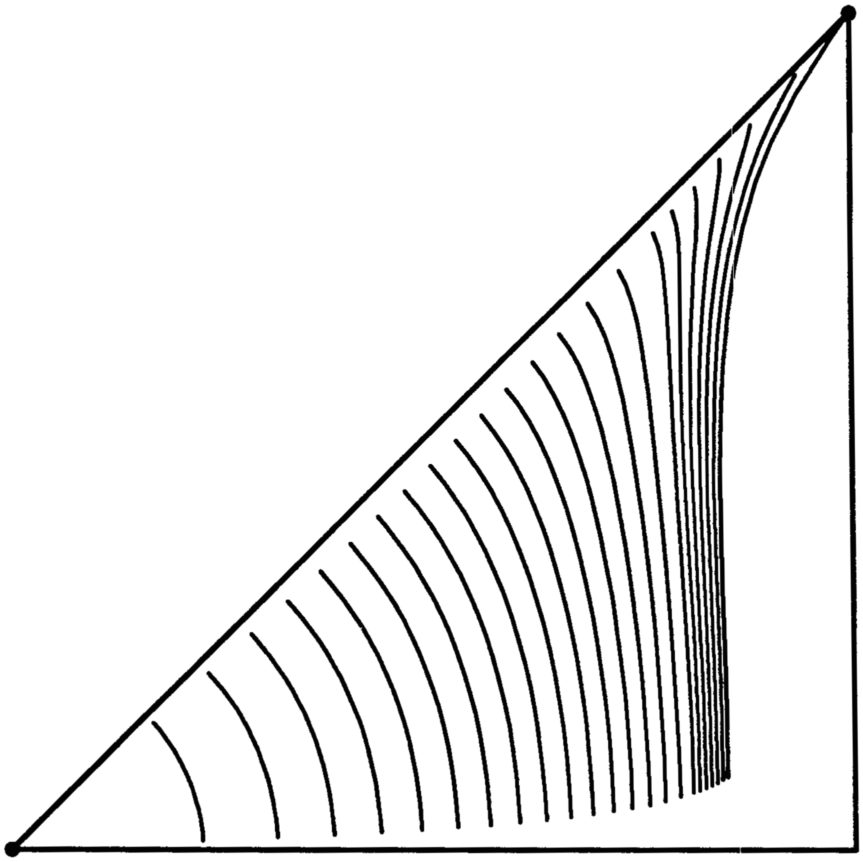


Figure 3.5 Isochrones at breakthrough in a 5-spot pattern

3.6 and **3.7** show the streamlines and isochrones for the symmetric element of a staggered line drive pattern with a d/a ratio of 1.0.

As the fluid particles approach the pressure sink at the top right comer, they rapidly move toward it. In numerical schemes, such as finite differences, distinct preferential flow paths are not well represented owing to the smearing of the front due to grid orientation effects. Fine gridding or other action must be taken to reduce the effect. The BEM, on the other hand, retains the **sharp** front. The location of the front is not affected by preferential flow paths since there is no gridding of the interior of the reservoir.

Cumulative recovery until breakthrough is calculated by material balance. A schematic of an oil reservoir undergoing a balanced waterflood with two injectors and four producers is shown in Fig. **3.8.** Figure **3.9** shows the streamlines in the flow field at breakthrough at individual wells. As a streamline breaks through at a well, it is stopped but the others are continued until all the wells have been broken into. A couple of the streamlines in Fig. 3.9 are seen to have a kink or a knot at a point. This is because the computational procedure was stopped when a streamline broke through at a well and subsequently restarted. During this restart procedure a couple of the streamlines were replaced inadvertantly by different streamlines which were very close to the original one, but were not the same. As the internal computation point approaches within half-a-boundary element length from the boundary, the solutions are affected due to the nature of the boundary approximations. Smaller elements must be used to preserve or increase accuracy of these streamlines. This layer close to the boundary where the solutions start to be affected has been termed the boundary layer by some researchers.

Figure **3.10** shows the streamlines in a two-well water-drive system in a 2:1 rectangle. The short side of the rectangle (the dashed line) is a constant pressure boundary or an active water support. Both wells produce at equal rates, and are placed :symmetrically in the reservoir. The coordinate locations **of** Wells 1 and 2 are (0.5, 0.5) and (1.5, 0.5) respectively. Point O on Fig. 3.10 is a limiting streamline. A particle to the left of point O goes to well 1 and a particle infinitesimally to the right of point O goes to the well 2. ***Ramey et al (2973)***

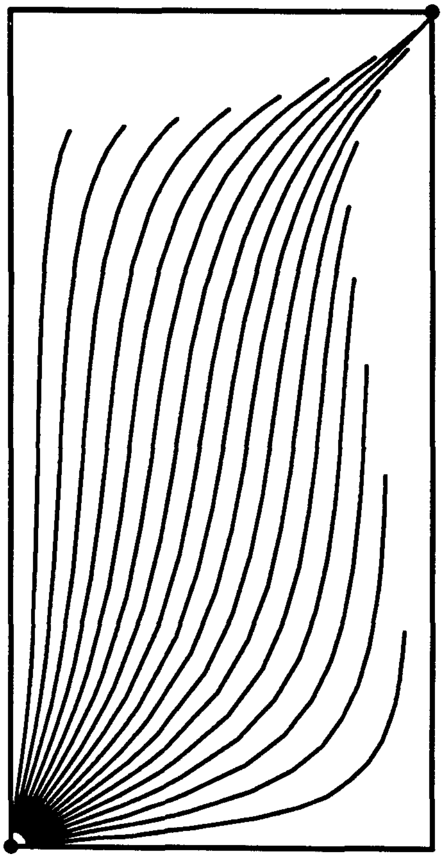


Figure **3.6** Streamlines at breakthrough in a staggered line drive

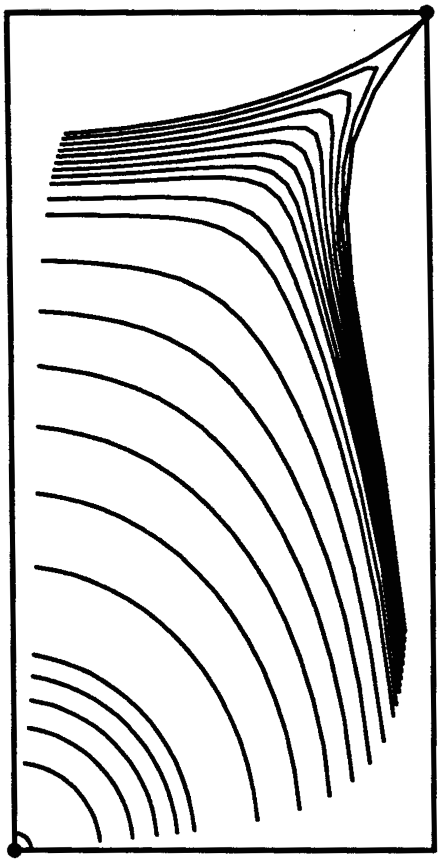


Figure **3.7** Isochrones at breakthrough in a staggered line drive

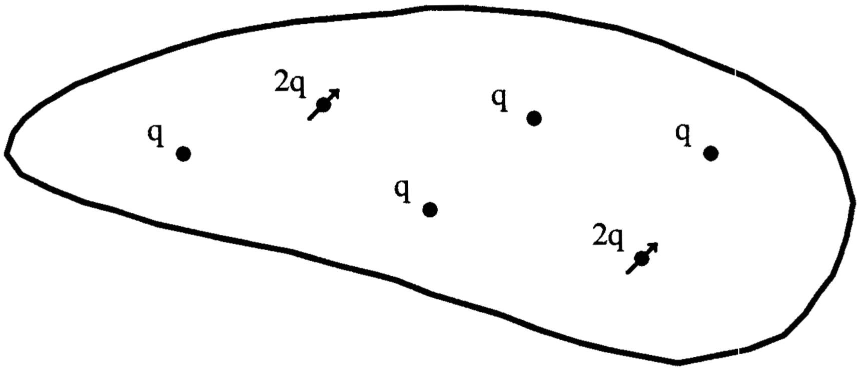


Figure **3.8** Schematic of the production scheme in a simulated reservoir

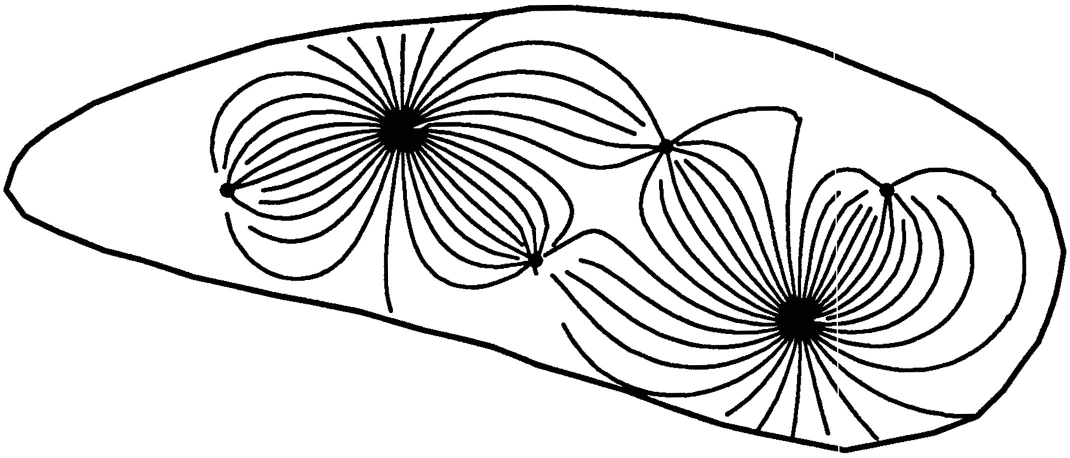


Figure **3.9** Streamlines at breakthrough in the simulated reservoir

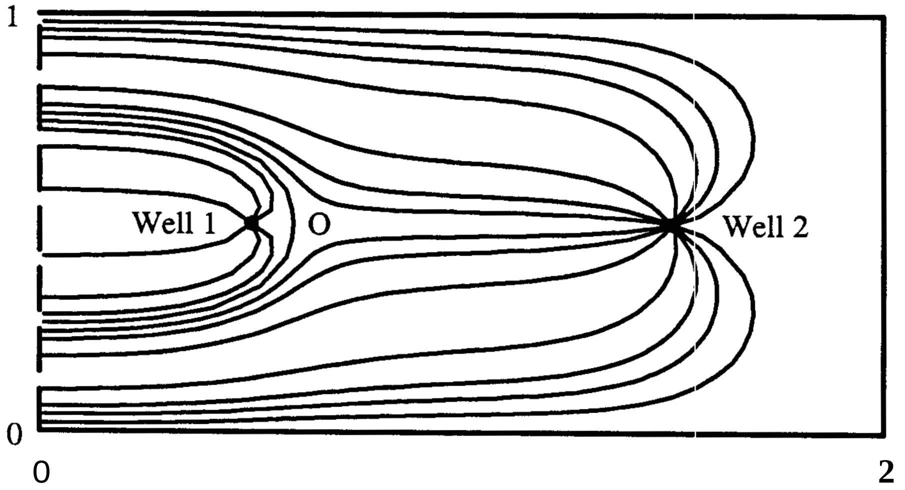


Figure 3.10 Streamlines in a 2:1 rectangle with one short side at constant pressure

report oscillations in numerically tracking this stagnation point. 'With the boundary element method, two particles were chosen close to the constant pressure boundary. These particles were positioned equidistant from the line of symmetry joining the two producing wells. Reverse tracking both the particles in time led to both of them converging to the location of the stagnation point, where they remained for all subsequent times. Table 3.5 shows the approach of the two particles to the stagnation point. The lower-half streamline in Table 3.5 shows the coordinate locations of the streamline which starts at the coordinates (0.01, 0.26) and is tracked until it reaches the stagnation point (0.6, 0.5). Similarly, the upper-half streamline begins at the coordinates (0.01, 0.74) and approaches the stagnation point from above. This demonstrates that the location of drainage boundaries in steady state flow fields could be done accurately with this technique.

Table 3.5 Approach of Limiting Streamlines to Stagnation Point

Lower-half Streamline		Upper-half Streamline	
X-Coord.	Y-Coord.	X-Coord.	Y-Coord.
0.01000	0.26000	0.01000	0.74000
0.07730	0.26000	0.07730	0.73900
0.14470	0.26300	0.14470	0.73600
0.21200	0.26600	0.21200	0.73300
0.27900	0.27300	0.27900	0.72600
0.40950	0.29800	0.40950	0.70100
0.52090	0.35000	0.52090	0.64900
0.58370	0.41800	0.58370	0.58100
0.59700	0.46600	0.59700	0.53300
0.60000	0.50000	0.60000	0.50000

Chapter 4

UNSTEADY STATE FLOW

The equation governing unsteady flow of single phase fluids through homogeneous, anisotropic porous media is given by Eq **(2.12).** This is the diffusivity equation and is used widely for pressure transient problems. **A** time dimension must be added to the previous considerations for steady state flow. Two different formulations for the transient problems are proposed in this chapter. They are :

- Real Space
- Laplace Space

Both these techniques **are** investigated in detail and the respective merits of each is analyzed. Verification and new solutions are presented for both formulations in the results section in this chapter.

4.1 Transient Real Space

The goveming equation is given by Eq. (2.19). Time appears as a first derivative term in the diffusivity equation. The presence of the first derivative term destroys the formal self-adjoint character of the Laplacian operator found in steady state problems. The diffusion operator is not **a** self-adjoint operator and cannot be cast into one. The integral equation is given by Eq. (2.37)

$$p_D(\, x_D\, ,\, y_D\, ,\, t_{DA}\,) \;=\; \int\limits_{t_{DA}} d\tau \int\limits_{\Gamma} (G \frac{\partial p_D}{\partial n} - p_D \frac{\partial G}{\partial n})\; dS \;+\; \sum_{i=1}^{n_w} G_i\; Q_{D\,i} \qquad (4.1)$$

where the source term in Eq. (2.19) has been treated as a line source. G_i is evaluated at the source locations (x_{Di}, y_{Di}). Since line sources are singularities in the flow field, the source term can be treated by singularity programming primarily because the differential equation is linear.

The free space Green's function 'G' in the above equation is defined as the pressure response at (x_D, y_D) at a time t_{DA} due to an instantaneous line source of strength unity generated at coordinates (ξ, ζ) at the time τ (*Carslaw and Jaegar,* 19.59).The medium is initially at zero pressure and infinite in extent. The free space Green's function is

$$G(x_D, y_D, t_{DA}, \xi, \zeta, \tau) = \frac{1}{4\pi(t_{DA}-\tau)} H(t_{DA}-\tau) \exp\left[-\frac{r_D^2}{4(t_{DA}-\tau)}\right]^{1}. \tag{4.2}$$

where $H(t_{DA} - 2)$ is the Heaviside step function. The derivation of Eq. *(4.2)* by the method of Fourier transforms is given in Appendix A. Substituting Eq. *(4.2)* in Eq. *(4.1)*

$$2\theta\, p_D(\xi, \zeta, \tau) = \int_{t_{DA}} d\tau \int_{\Gamma} dS \left[\frac{1}{(t_{DA}-\tau)} \exp\left[-\frac{r_D^2}{4(t_{DA}-\tau)}\right] \frac{\partial p_D}{\partial n} - \right.$$

$$\left. p_D \frac{\partial}{\partial n}\left\{\frac{1}{(t_{DA}-\tau)} \exp\left[-\frac{r_D^2}{4(t_{DA}-\tau)}\right]\right\}\right] \tag{4.3}$$

where the source term has been dropped for simplicity. θ in Eq. *(4.3)* corresponds to the angle subtended by any two adjacent boundary elements. If the fictitious Green's function source is completely enclosed by the boundary of the reservoir, θ becomes 2π. Symbolically

$$\theta = 2\pi \qquad \text{if } (x_D, y_D) \in \Omega \tag{4.3a}$$

$$\theta = \theta \qquad \text{if } (x_D, y_D) \in \Gamma \tag{4.3b}$$

On evaluating the normal derivative of the Green's function, Eq. *(4.3)* becomes

$$2\theta\, p_D(\xi, \zeta, \tau) = \int_{t_{DA}} \frac{d\tau}{(t_{DA}-\tau)} \int_{\Gamma} dS \exp\left[-\frac{r_D^2}{4(t_{DA}-\tau)}\right] \frac{\partial p_D}{\partial n} +$$

$$\int_{t_{DA}} \frac{d\tau}{2(t_{DA}-\tau)^2} \int_{\Gamma} p_D\, r_D \frac{\partial r_D}{\partial n} \exp\left[-\frac{r_D^2}{4(t_{DA}-\tau)}\right] dS \tag{4.4}$$

where

$$r_D^2 = (x_D - \xi)^2 + (y_D - \zeta)^2 \tag{4.5}$$

4.1.1 Time Stepping

The time t_{DA} at which the solution is desired appears both in the integrand and the limits of integration in a convolution fashion, that is the pressun: and the fluxes are functions of time in the forward direction but the basis functions depend on time coming backward from total time. This is not intuitive, but can be explained by the concept of energy integrals.

The energy integrals are based on the fact that the total system energy is conserved and the system tends to a minimum energy state. For the diffusion operator to conserve energy, the adjoint problem is considered reversed in time. The product of the forward and adjoint problem conserves energy. In fact the variational formulation of the problem is based on this principle.

The convolution character of the integrals which is evident from above suggests a history dependence of the diffusion operator. The solution at a time t_{DA} depends on the solution at all previous times.

Wrobel and Brebbia (1981) outlined a number of time stepping schemes. The merits and problems of several methods are discussed next. The solution at a particular time step could be treated as an initial condition for later time steps. The solution can then be advanced in time, stepwise. This destroys the boundary only character of the solution, as a domain integral over the initial condition appears as an extra term in Eq. **(4.4).** To evaluate the domain integral, the internal region must be discretized in element!;. Thus, at every time step one domain or volume integral must be evaluated. The discretization of the reservoir diminishes the appeal of the procedure as being a boundary only method. Another way to handle the integrals is to start the solution at zero time. Domain integrals do not arise in this case but the computations can become tedious, especially for solution at long times because at long times the time slot is divided into a large number of segments. Excessive repetitive computations must be performed in this case.

In this work, the convolution character is considered, by storing the matrices from all previous times levels to construct solutions at subsequent rimes. If the time step size is assumed constant, only one new matrix must be generated at each time level. This was suggested by *Pina (1984).* The computational inefficiency of the previous procedure is replaced by increased computer storage requirements. The matrix structure is shown in the next section.

Once the time step size requirements have been decided, the representation of pressure and fluxes in each time slot must be dealt with. In this work constant elements in time are prescribed.

4.1.2 Matrix Form of the Integral Equation

The interpolation functions between the boundary nodes for pressure and fluxes are represented as in Eqs. **(2.38)** and **(2.39).** Substituting Eqs. **(2.38)** and **(2.39)** in **Eq. (4.4)** and using a transformation to a moving coordinate system, the following system of equations is obtained

$$\sum_{j=1}^{N} 2\theta_i \, \delta_{ij} \, p_{D_j} = \sum_{j=1}^{N} \left[(AA)_{I_{ij}} + (AA)_{II_{ij}} \right] \qquad i = 1, N \tag{4.6}$$

where

$$(AA)_{I_{ij}} = \frac{\zeta_i}{(\xi_{j+1} - \xi_j)}(p_{D_{j+1}} - p_{D_j}) \left[E_1\left[\frac{\xi_j{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\right] - E_1\left[\frac{\xi_{j+1}{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\right]\right] + \frac{2\zeta_i}{(\xi_{j+1} - \xi_j})$$
$$\exp\left[-\frac{\zeta_i{}^2}{4(t_{DA}-\tau)}\right](\xi_{j+1}\, p_{D_j} - \xi_j\, p_{D_{j+1}}) \left[\int_{\xi_j}^{\xi_{j+1}} \frac{d\xi}{(\xi^2 + \zeta_i{}^2)} \exp\left[-\frac{\xi^2}{4(t_{DA}-\tau)}\right]\right] \tag{4.7}$$

and

$$\begin{aligned}(AA)_{II_{ij}} \;=\; & \frac{(p_{D_{n_{j+1}}} - p_{D_{n_j}})}{(\xi_{j+1} - \xi_j)} \Bigg[\frac{1}{2} \Bigg\{ (\xi_{j+1}{}^2 + \zeta_i{}^2)\, E_1\Bigg[\frac{\xi_{j+1}{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] - (\xi_j{}^2 + \zeta_i{}^2)\, E_1\Bigg[\frac{\xi_j{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\} \\ & + 2(t_{DA}-\tau)\, \exp\Bigg[-\frac{\zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\{\exp\Bigg[-\frac{\xi_j{}^2}{4(t_{DA}-\tau)}\Bigg] - \exp\Bigg[-\frac{\xi_{j+1}{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\}\Bigg] \\ & + \frac{(\xi_{j+1}\, p_{D_{n_j}} - \xi_j\, p_{D_{n_{j+1}}})}{(\xi_{j+1} - \xi_j)} \Bigg[\int_{\xi_j}^{\xi_{j+1}} E_1\Bigg[\frac{\xi^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] d\xi \Bigg] \end{aligned} \tag{4.8}$$

where

$$E_1(z) \;=\; \int_z^{\infty} \frac{e^{-x}}{x}\, dx \tag{4.9}$$

is the exponential integral. Rewriting Eq. (4.6) in a compact form

$$\sum_{j=1}^{N} a_{ij}\, p_{D_j} + b_{ij}\, p_{D_{n_j}} \;=\; 0 \qquad\qquad i = 1, N \tag{4.10}$$

where

$$\begin{aligned} a_{ij} \;=\; & \frac{\zeta_i}{(\xi_{j+1} - \xi_j)} \Bigg[E_1\Bigg[\frac{\xi_{j+1}{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] - E_1\Bigg[\frac{\xi_j{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg] + \frac{2\zeta_i\ \xi_{j+1}}{(\xi_{j+1} - \xi_j}) \\ & \exp\Bigg[-\frac{\zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] \Bigg| \int_{\xi_j}^{\xi_{j+1}} \frac{d\xi}{(\xi^2 + \zeta_i{}^2)} \exp\Bigg[-\frac{\xi^2}{4(t_{DA}-\tau)}\Bigg] \Bigg| - 2\theta_i \delta_{ij} \end{aligned} \tag{4.10a}$$

and

$$\begin{aligned} b_{ij} \;=\; & -\frac{1}{(\xi_{j+1} - \xi_j)} \Bigg[\frac{1}{2} \Bigg\{ (\xi_{j+1}{}^2 + \zeta_i{}^2)\, E_1\Bigg[\frac{\xi_{j+1}{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] - (\xi_j{}^2 + \zeta_i{}^2)\, E_1\Bigg[\frac{\xi_j{}^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\} \\ & + 2(t_{DA}-\tau)\, \exp\Bigg[-\frac{\zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\{\exp\Bigg[-\frac{\xi_j{}^2}{4(t_{DA}-\tau)}\Bigg] - \exp\Bigg[-\frac{\xi_{j+1}{}^2}{4(t_{DA}-\tau)}\Bigg]\Bigg\}\Bigg] \\ & + \frac{1}{(\xi_{j+1} - \xi_j)} \Bigg[\int_{\xi_j}^{\xi_{j+1}} E_1\Bigg[\frac{\xi^2 + \zeta_i{}^2}{4(t_{DA}-\tau)}\Bigg] d\xi \Bigg] \end{aligned} \tag{4.10b}$$

The boundary conditions are used in Eq. (4.10) to create a right hand side vector. The matrix of unknowns is the coefficient matrix. The normal derivatives in the preceding equations have been represented by the subscript n for convenience.

All but two of the integrals in Eq. (4.3) were evaluated analytically. The rest of the integrals in Eqs. (4.7) and (4.8) were evaluated with highly accurate Gauss-Legendre panel type integration schemes (*Stroud and Secrest,* 1966). The Gauss-Legendre integration scheme gives the most accurate results for a given order of polynornial representation of the given function. The disadvantage of this scheme is that for a different number of integration points, a new set of abscissae and weights must be generated. The range of integration was divided into two panels and a six point integration scheme was used on each of the panels. The abscissae and weights for each integration point were generated automatically depending on the number of points and the number of panels. This was found. in general, to be the optimum between computing effort and accuracy.

When the collocation point is on the same boundary element as the field point, integrals are obtained which are singular at one of the limits. This can be seen from Fig. 2.2. This happens when ζ is zero and ξ_j or ξ_{j+1} is zero. The usual techniques for numerical integration are unsuitable. There are several ways to handle this. One of the ways is to integrate analytically from the singularity to a value δ, then numerically from *6* onwards. Another method could be to use the Gaussian integration scheme with points spaced to avoid the singularity. Yet another way could be to perform a term by term integration of an infinite series obtained by expanding the integrand. This infinite series is slowly convergent at large values of the argument.

All of the methods described above require either removing the singularity and integrating numerically or performing a term by term integration of an infinite series. This integral can be evaluated exactly and a simple closed form integration scheme in terms of smoothly behaved functions is described in Appendix C.

4.1.3 Solution of Matrix Equations

If we assume a constant time step size, $t_{DA} = t_0 + k\Delta t_{DA}$, where Δt_{DA} is the step size. The matrix equation generated can be rewritten in the following form for convenience

$$\sum_{\alpha=1}^{k} H^{k\alpha} u^{\alpha} = G^{\alpha} \tag{4.11}$$

where H is the coefficient matrix, G is the right hand side vector and u is the unknown vector. The unknown vector u can consist of either pressures or fluxes on the boundary nodes. It can also be a mixture of the pressures and fluxes, if the boundary conditions change at different places in the reservoir. The coefficient matrix has solutioin time, both as the limit of the sum and in the elements of the matrix. This arises from integrals of the form

$$\int_0^{t_{DA}} f(\tau)\, G(t_{DA}-\tau)\, d\tau$$

The convolution character in the above expression is evident. Appendix D.1 considers a specific example of interpolation function and boundary conditions to demonstrate the origin of the convolution structure in Eq. (4.11) using the Green's function for the diffusion equation. Appendix D.1 also shows [using *Eq.* (4.11)] that the solution at any given time is the sum of the results at all the previous time slots. If the solution up to $(k-1)^{th}$ time step is known, the solution at the k^{th} time level may be found from Eq. (4.11). Rewriting *Eq.* (4.11) in expanded form for the k^{th} time level

$$H^{k1}u^1 + H^{k2}u^2 + ... + H^{kk}u^k = G^k \tag{4.12a}$$

which on transposing becomes

$$H^{kk}u^k = G^k - \sum_{\alpha=1}^{k-1} H^{k\alpha}u^{\alpha} \tag{4.12b}$$

The matrix $H^{k\alpha}$ depends entirely on the geometry of the system and the step size. At the k^{th} time level for example, Eq. (4.12b) suggests that the unknown vector u^k at time *step* k is multiplied by a coefficient matrix H^{kk}. The right hand vector is the sum of the products of matrices obtained at earlier time levels with the solution vectors at previous times. The relation between matrices generated at different time steps is shown in an array form in Fig. **4.1.**

$$\left[\begin{array}{llllll} H^{11} & & & & & \\ H^{21} & H^{22} & & & & \\ H^{31} & H^{32} & H^{33} & & & \\ | & | & | & | & & \\ | & | & | & | & & \\ H^{k1} & H^{k2} & H^{k3} & H^{k\alpha} & - \; - & H^{kk} \end{array}\right|$$

Fig. **4.1** : Relation between matrices in convolution boundary element method

Figure **4.1** shows the coefficient matrices which are generated at each time level. For example, at the first time level, the only matrix that must be generated is H^{11}. At the second time level, two matrices are created - one each for the time slots Δt and $2\Delta t$, respectively. This is represented as row **2** in the matrix structure in Fig. **4.1.** At k^{th} time level, k number of matrices are generated as shown in row k of Fig. **4.1. As** mentioned earlier, if the time step size is assumed to be constant, a relationship between each of these matrices is found. The matrices on each of the diagonal lines are the same, that is $H^{32} = H^{21} = H^{43}$ etc. Similarly, the matrices on the principal diagonal of the array in Fig. **4.1** (i.e. $H^{11} = H^{22} = H^{kk}$) are identical. H^{kk} also happens to be the left hand side coefficient matrix in the matrix equation **Eq. (4.12b). This** coefficient matrix stays the same for the entire solution time due to the relation between these matrices seen from Fig. **(4.1).** Thus, **this** matrix can be inverted and stored, and at every time step only the right hand vector is created by summing the previously stored matrices in a convolution fashion.

At the k^{th} time level the right hand vector consists of (k - 1) matrices multiplied by their respective solution vectors. But at any given time level only one of these matrices must be generated anew. If we look at the third row in Fig. **(4.11,** for example, there are three matrices that are created because it is the third time level. H^{33} is the same as H^{11}, also H^{32} is the same as H^{21} and has been generated at the second time level already. The only matrix to be created is H^{31}. Appendix D.2 shows the matrix structure and creation of the right hand side vector **involving** convolution of matrices up to the third time step to demonstrate the idea behind the convolution method.

4.1.4 Computational Details and Algorithm

The program structure (algorithm) is outlined in Fig. 4.2. After initialization of vectors and arrays, the area of the reservoir is computed by triangulation of the domain. The area of individual triangles are computed and summed. Subsequently, the matrix coefficients are computed by integration over individual elements in a local coordinate system based on the fictitious source point. This was outlined in Chapter 2. The boundary conditions are multiplied by the corresponding matrix elements to create a right hand side vector. The manner in which this is handled is by assigning node identification numbers to each of the nodes. The identification number specifies the type of boundary condition on the particular nodal location. Three quantities are needed at each of the nodes, two of which are known from the boundary condition. This is illustrated in Fig. 4.3. The unknown information is entered as a zero (0) in the datafile. The following are the pieces of information :

- Pressure, p_D
- Normal derivative of pressure leaving the node, $p_{D_{nl}}$
- Normal derivative of pressure approaching the node, $p_{D_{na}}$

If the node is not a comer, then it is assumed that $p_{D_{nl}} = p_{D_{na}}$ and there is no discontinuity **in** fluxes. The combination of various boundary conditions and their associated identification numbers are given in Table 4.1.

The matrix generated is inverted by Gauss-Jordan elimination with pivoting, after which it is stored. The reason for storing the inverted matrix has been explained earlier in this

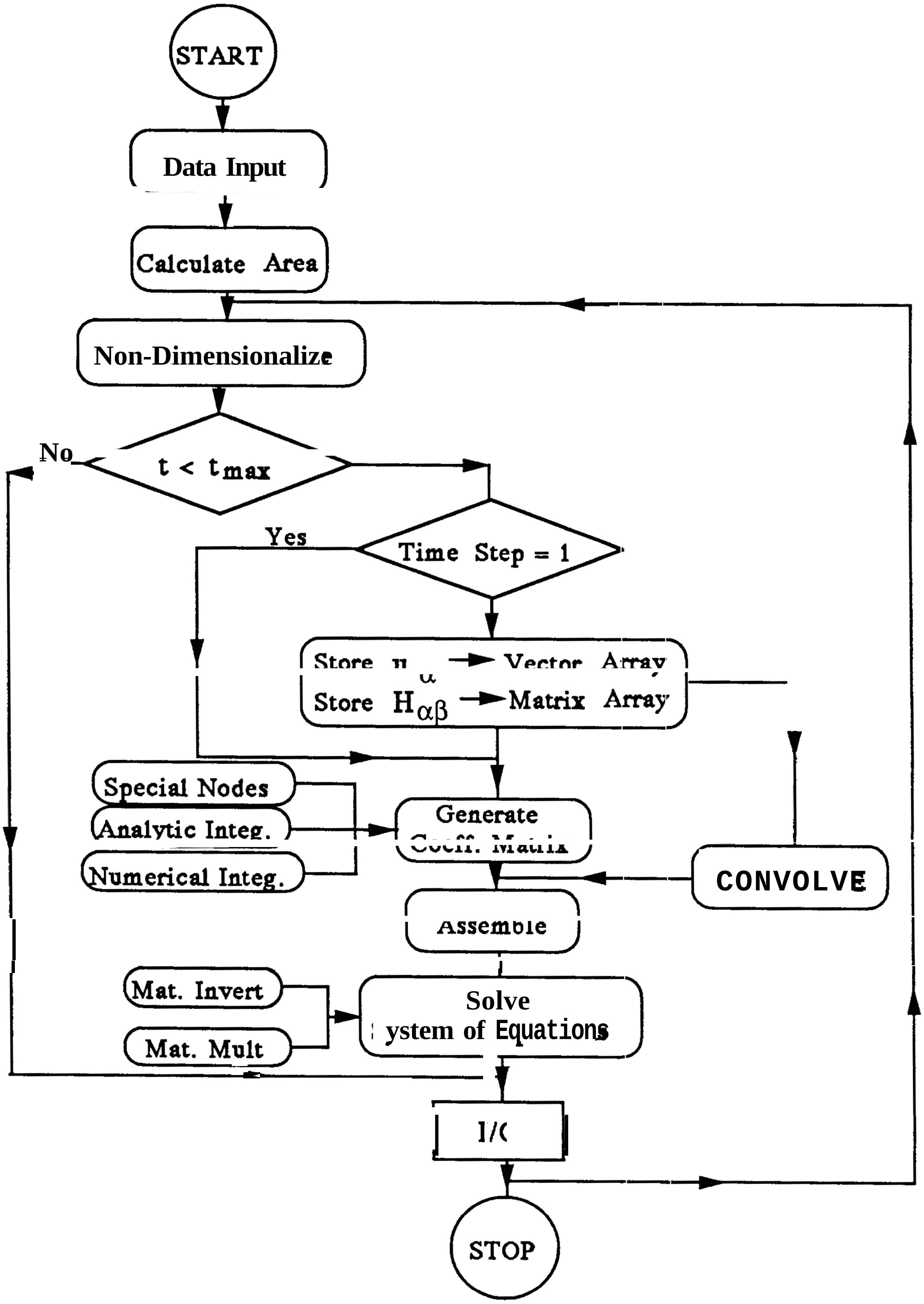


Fig. **4.2** Flow chart for the convolution BEM

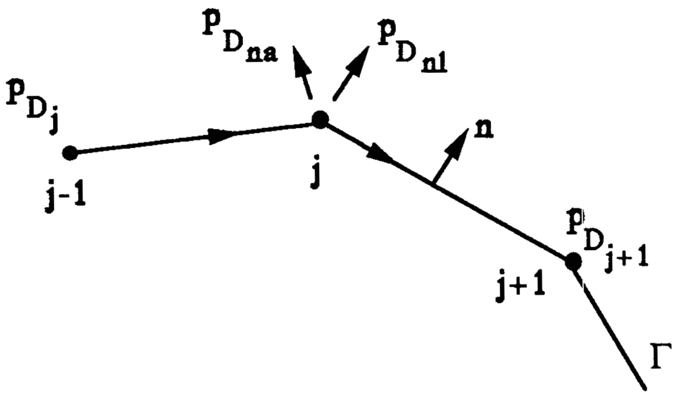


Fig. **4.3** Illustration of dependent variables at boundary nodes

Table 4.1 Identification of element nodes

Node ID	Known BC	Unknown BC	Equation
1	p_D $p_{D_{nl}}$	$p_{D_{na}}$	$p_D = c_1$ $p_{D_{nl}} = c_2$
2	p_D $p_{D_{na}}$	$p_{D_{nl}}$	$p_D = c_1$ $p_{D_{na}} = c_2$
3	p_D	$p_{D_{nl}}$ $p_{D_{na}}$	$p_D = c_1$ $p_{D_{nl}} = p_{D_{na}}$
4	$p_{D_{nl}}$ $p_{D_{na}}$	p_D	$p_{D_{nl}} = c_1$ $p_{D_{na}} = c_2$
5	$p_{D_{nl}}$	p_D $p_{D_{na}}$	$\alpha p_D + \beta p_{D_{na}} = c_1$ $p_{D_{nl}} = c_2$
6	$p_{D_{na}}$	p_D $p_{D_{nl}}$	$\alpha p_D + \beta p_{D_{nl}} = c_1$ $p_{D_{na}} = c_2$
7	$p_{D_{nl}} = p_{D_{na}}$	p_D $p_{D_{nl}}$ or $p_{D_{na}}$	$p_D + \alpha p_{D_{nl}} = c_1$ $p_D + \alpha p_{D_{na}} = c_1$
8	-	p_D $p_{D_{nl}}$ $p_{D_{na}}$	$p_D + \alpha p_{D_{nl}} = c_1$ $p_D + \beta p_{D_{na}} = c_2$

section and is because the left hand side unknown coefficient matrix remains the same throughout the computations. Only the right hand vector is modified every time step as shown in Eq. (4.11b).

Once the solution at a particular time **step** is obtained, the solution vector and the coefficient matrices **are** stored in **an** array. The coefficient matrix is stored **as** a two dimensional array of matrices and the solution vector in a one dimensional vector array. These arrays **are** convolved at subsequent time levels to create the right hand side vector.

4.1.5 Limitations of Convolution BEM

The elegance **of** the boundary only nature of the solutions **is** preserved by treating the problem in real space and considering the convolution nature of the integrals. The use of constant time step size requires that the coefficient mamx and the solution vectors for **all** times be stored. In modem computers, high speed auxiliary memory access and large swap spaces provide large and quickly accessible memory, but due to the dyriamic growth in the number of matrices that must **be** stored, the solution becomes a tedious and expensive **task** for large times. Another limitation is that small errors created at any time due to roundoff errors can accumulate. Examples will be shown later in **this** chapter which suggest that **this** limitation is not prohibitive in that good solutions can **be** obtained if the boundary geometry is represented well.

4.2 Laplace Space Formulation

Since robust numerical inversion routines are available fca inverting Laplace space solutions for the diffusivity equation, the boundary element proceclure can be performed in Laplace space and the solution inverted to real space. The advantage that can be anticipated is that time appears as **a** parameter in Laplace space thereby removing the complication arising from the convolution nature of the integrals. The disadvantage is that the kernel (Green's) function is more complex and most **of** the integrals arising from the boundary discretization must be done numerically.

Equation (2.1) is transformed to Laplace space. The Laplace transform is

$$L\{f(t)\} = \bar{f}(s) = \int_0^\infty e^{-st} f(t)\,dt \tag{4.13}$$

where the function f(t) is of exponential order. The transformed equation with the initial conditions is

$$\frac{\partial^2 \bar{p}_D}{\partial x_D^2} + \frac{\partial^2 \bar{p}_D}{\partial y_D^2} = s\,\bar{p}_D + \frac{1}{s}\sum_{i=1}^{n_w} Q_{D_i}\,\delta(x_D - x_{D_i})\,\delta(y_D - y_{D_i}) \tag{4.14}$$

where **s is** the Laplace transform parameter. The boundary conditions become

$$\bar{p}_D = \frac{p_1}{s} \qquad on\ \Gamma_1 \in \Gamma \tag{4.15a}$$

$$\frac{\partial \bar{p}_D}{\partial n} = \frac{q_1}{s} \qquad on\ \Gamma_2 \in \Gamma \tag{4.15b}$$

$$\alpha\,\bar{p}_D + \beta\,\frac{\partial \bar{p}_D}{\partial n} = \frac{\gamma}{s} \qquad on\ \Gamma_3 \in \Gamma \tag{4.1\%}$$

The differential operator in Eq. (4.14) can be defined as

$$L(\bar{p}_D) \equiv (\nabla^2 - s)\,\bar{p}_D = 0 \tag{4.16}$$

The operator L is known as the modified Helmholtz operator. L is now self-adjoint as the first order time derivative in the diffusivity equation has been replaced by the zeroth order derivative. Casting Eq. (4.16) into divergence form as described in Section **2.3,** and integrating over the domain of the problem, results in

$$\int_\Omega \bar{G}L(\bar{p}_D) \equiv \int_\Omega \nabla\bullet(\bar{G}\,\nabla\bar{p}_D - \bar{p}_D\,\nabla\bar{G})\,dA + \int_\Omega \bar{p}_D(\nabla^2\bar{G} - s\bar{G})\,dA +$$

$$\frac{1}{s}\int_\Omega \bar{G}\sum_{i=1}^{n_w} Q_{D\,i}\,\delta(x_D - x_{D_i})\,\delta(y_D - y_{D_i}) = 0 \tag{4.17}$$

Defining the adjoint operator $L^*(G)$ as

$$L^*(\bar{G}) \equiv \nabla^2 \bar{G} - s\,\bar{G} = -\,\delta(x_D - x_{D_i})\,\delta(y_D - y_{D_i}) \tag{4.18}$$

and using the shifting property of the Dirac delta function, and the divergence theorem of Gauss, we obtain

$$\theta\,\bar{p}_D(x_D, y_D; s) = \int_\Gamma \left(\bar{G}\,\frac{\partial \bar{p}_D}{\partial n} - \bar{p}_D\,\frac{\partial \bar{G}}{\partial n} \right) dS + \frac{1}{s} \sum_{i=1}^{n_w} \bar{G}_i\, Q_{D_i} \tag{4.19}$$

where θ has the same meaning as Eqs. (4.3a) and (4.3b). $\bar{G}_i$ in Eq. (4.19) denotes the contribution of the Green's function from the source point (well). The solution to the adjoint problem of Eq. (4.18) gives the free space Green's function for the modified Helmholtz operator and is

$$\bar{G}(x_D, y_D, \xi, \zeta; s) = -\,\frac{1}{2\pi} K_0(\,|r_D|\sqrt{s}\,) \tag{4.20}$$

where $K_0(z)$ is the modified Bessel function of second kind of order zero, and r_D is given by Eq. (4.5). The value 2π in Eq. (4.20) has been replaced by θ in the equation (4.19) because the Green's function source is restricted to the domain of interest and does not act in free space. The derivation of Eq. (4.20) is given in Appendix A.

The solution procedure is analogous to the real space formulation described in the previous section. Eq. (4.20) is substituted in Eq. (4.19) and using Eq. (2.33) and (2.34) as space interpolation functions, matrix equations are generated. The discretized equation is

$$\sum_{j=1}^{N} \theta_i\,\delta_{ij}\,\bar{p}_{D_j} = \sum_{j=1}^{N} \left[(\overline{AA})_{I_{ij}} + (\overline{AA})_{II_{ij}} \right] + \frac{1}{s} \sum_{i=1}^{n_w} \bar{G}_i\, Q_{D_i} \qquad i = 1, N \tag{4.21}$$

where

$$\begin{aligned}(\overline{AA})_{I_{ij}} = {} & \frac{\zeta_i}{(\xi_{j+1} - \xi_j)} (\bar{p}_{D_{j+1}} - \bar{p}_{D_j}) \left[K_0 \left[\sqrt{s\,(\xi_j^2 + \zeta_i^2)} \right] - K_0 \left[\sqrt{s\,(\xi_{j+1}^2 + \zeta_i^2)} \right] \right] \\ & + \frac{\zeta_i \sqrt{s}}{(\xi_{j+1} - \xi_j)} (\xi_{j+1}\,\bar{p}_{D_j} - \xi_j\,\bar{p}_{D_{j+1}}) \left[\int_{\xi_j}^{\xi_{j+1}} \frac{d\xi}{\sqrt{(\xi^2 + \zeta_i^2)}}\, K_1 \left[\sqrt{s\,(\xi^2 + \zeta_i^2)} \right] \right]\end{aligned} \tag{4.22}$$

$$(\overline{AA})_{II_{ij}} = \frac{(\bar{p}_{D_{n_{j+1}}} - \bar{p}_{D_{n_j}})}{(\xi_{j+1} - \xi_j)} \left[\int_{\xi_j}^{\xi_{j+1}} \xi \, K_0\left[\sqrt{s \ (\xi^2 + \zeta_i^2)}\right] \, d\xi \right] + \frac{(\xi_{j+1} \, \bar{p}_{D_{n_j}} - \xi_j \, \bar{p}_{D_{n_{j+1}}})}{(\xi_{j+1} - \xi_j)}$$
$$\left[\int_{\xi_j}^{\xi_{j+1}} K_0\left[\sqrt{s (\xi^2 + \zeta_i^2)}\right] d\xi \right] \tag{4.23}$$

In a more compact notation

$$\sum_{j=1}^{N} [\, a_{ij} \, \bar{p}_{D_j} + b_{ij} \, \bar{p}_{D_{n_j}} \,] \; = \; 0 \qquad i = 1, N \tag{4.24}$$

where

$$a_{ij} = \frac{\zeta_i}{(\xi_{j+1} - \xi_j)} \left[K_0 \left[\sqrt{s \ (\xi_{j+1}^2 + \zeta_i^2)}\right] - K_0 \left[\sqrt{s \ (\xi_j^2 + \zeta_i^2)}\right]\right] + \frac{\zeta_i \ \xi_{j+1} \sqrt{s}}{(\xi_{j+1} - \xi_j)}$$
$$\left[\int_{\xi_j}^{\xi_{j+1}} \frac{d\xi}{\sqrt{(\xi^2 + \zeta_i^2)}} \, K_1\left[\sqrt{s \ (\xi^2 + \zeta_i^2)}\right] \right] - \theta_i \ \delta_{ij} \tag{4.24a}$$

and

$$b_{ij} = - \frac{1}{(\xi_{j+1} - \xi_j)} \left[\int_{\xi_j}^{\xi_{j+1}} \xi \, K_0\left[\sqrt{s \ (\xi^2 + \zeta_i^2)}\right] \, d\xi + \xi_{j+1} \int_{\xi_j}^{\xi_{j+1}} K_0\left[\sqrt{s (\xi^2 + \zeta_i^2)}\right] d\xi \right] \quad \text{(4.24t}$$

The boundary conditions are used in the above equation to create a right hand vector. The coefficient matrix is inverted to obtain the solution in Laplace space. Once the solution at a particular time is obtained in Laplace space, it is inverted to real space by means of the *Stehfest* (1970) algorithm.

Appendix E.1 gives a detailed explanation of the datafile needed to run the Laplace space BEM computer program which is given in Appendix **F.** A sample datafile is included in Appendix E.2 and a sample output file for the output at any interior location so that the for the the wellbore pressure and semilog pressure derivative as a function of dimensionless time is given in Appendix E.3.

4.3 Results and Comparison

Both convolution and Laplace space formulations are applied to a variety of problems with increasing level of difficulty. The primary goal being to evaluate the methods for accuracy, robustness, speed, storage requirements and generality of application.

On one hand, the convolution formulation has the advantage of being in real space, and most of the integrals may be performed analytically. On the other hand, the Laplace transform solutions do not have a history dependent nature, as the time domain is replaced by a parameter. Preliminary discussions concern solutions of problems which have closed form analytic solutions. The rest of the examples will be solved with the best method.

4.3.1 Comparison of the Two Formulations

Figure 4.4 shows a square box of porous media with impermeable upper and lower boundaries. The inner and outer boundaries are maintained at different conditions and both convolution and Laplace space BEM solutions are compared with their analytical counterparts. Figure 4.5 shows the pressure behavior at several locations for a mixed problem in which a step change in pressure is imposed at the inlet end [$p_D(x_D = 0) = 1$] and the other boundary is maintained at a constant pressure [$p_D(x_D = 1) = 0$]. The top and bottom surfaces of the porous medium are sealed. The solution is given by (*Liggett and Liu, 1979)*

$$p_D = (1 - x_D) + \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} e^{-k^2\pi^2 t_{DA}} \sin\{k\pi(1 - x_D)\} \tag{4.25}$$

The circles represent the convolution BEM with a time step size of 0.025, and the solid lines are the analytical solutions. For the convolution BEM, at early times the solutions are not good because of a rapid variation of Green's function. On reducing the time step size, the solution improves as shown in Fig. 4.6. Eight (8) equidistant nodal points on the boundary of

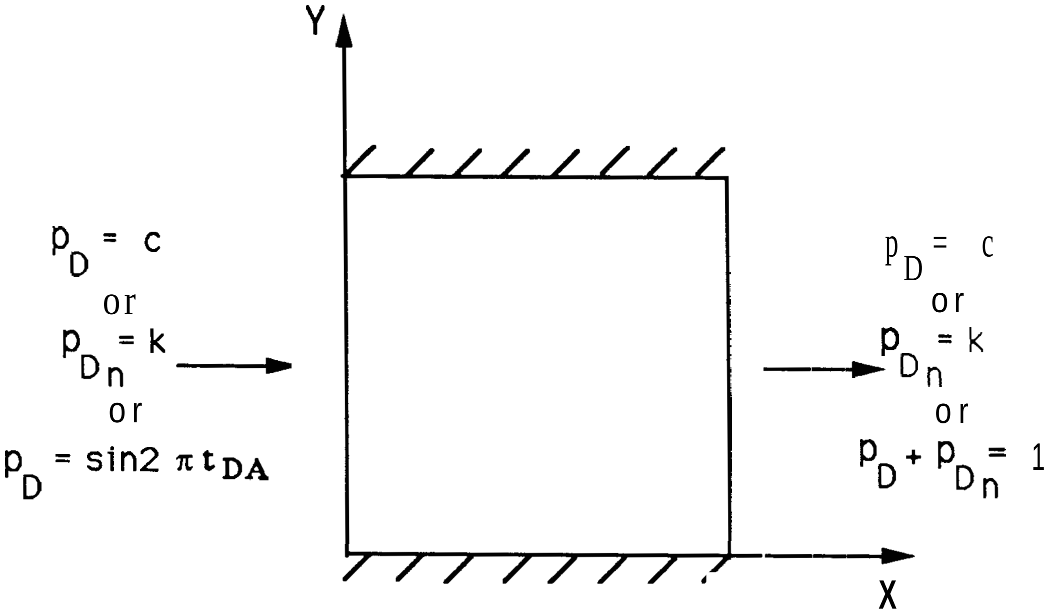


Fig. **4.4** Schematic of the reservoir for figures **4.5** through 4.10

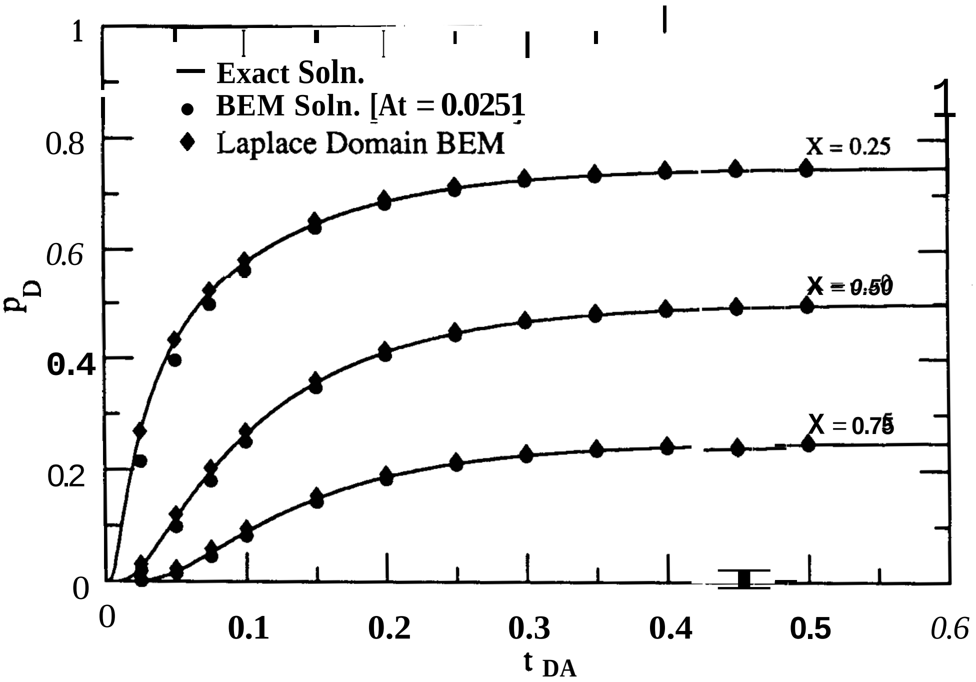


Fig. 4.5 Pressure behavior at various cross sections for a mixed type problem

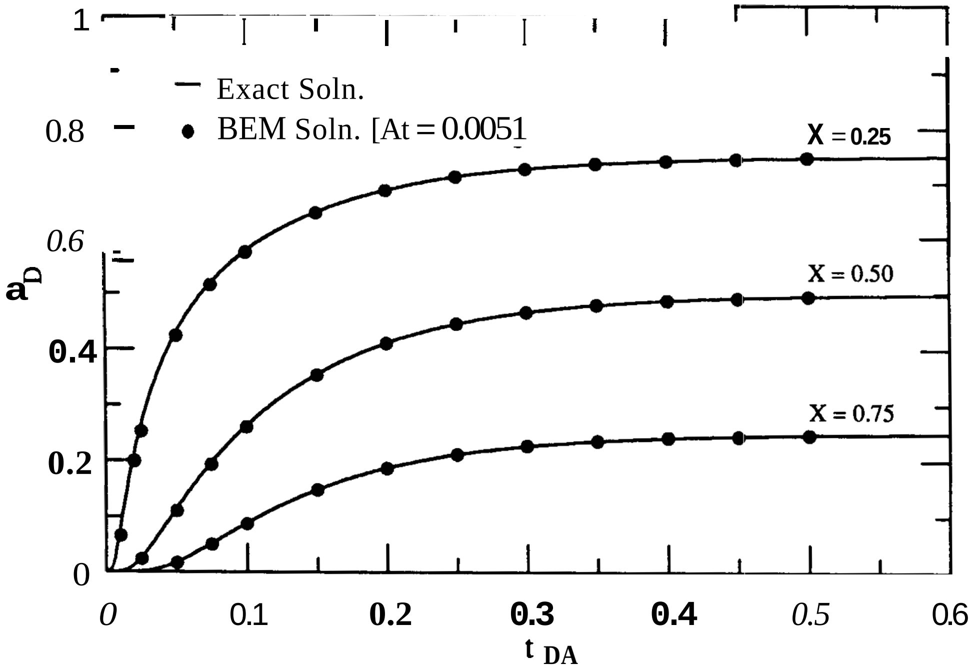


Fig. 4.6 Effect of time step refinement on the solution of a mixed problem

the problem domain were used for calculations. Figure 4.5 also shows the solution in Laplace space. The results are accurate. A finite time step is not required in Laplace space, since the solution can be determined at any required time point.

An interesting feature of the problem is the calculation of fluxes at the inlet end. Since a step change in pressure was imposed at time zero, there is a flux singularity at the inlet at early times, Figure 4.7 shows the flux as a function of time at the inlet end. Since the convolution solution considers the history dependence, small time step sizes have to be used to account for the flux singularity at early times. Taking smaller time steps improves the flux behavior. The Laplace space solution performs better, partly because it uses less computations and hence has less roundoff error. Also, since the results for the convolution method depend on all previous time solutions, error caused at a time step is carried through to the solution at subsequent time levels. Figure 4.8 shows the flux calculations with the convolution **BEM** assuming a refined time step size of 0.005. The error in flux calculations with the convolution **BEM** at a time (t_{DA}) of 0.05 with a time step size of 0.025 was about 30% and with the reduction in step size ($\Delta t_{DA} = 0.005$) the percentage error was reduced to 5% whereas the Laplace transform solution results gave error of less than 0.1% for the same time ($t_{DA} = 0.05$).

The mixed problem discussed in the preceding example was used to perform a stability analysis for the Convolution **BEM.** The stability analysis was done to evaluate the propagation of a small error introduced in one of the elements of a convolution matrix. For this purpose, a 5% error was introduced in an off-diagonal element of the matrix generated at the second time-step. Since this mamx is used for calculation of solutions at all the subsequent time steps due to the history dependent nature of the diffusion operator, it is important to determine whether the error attenuates or not. The maximum error in the solution at the third time step was 2.4%. For the solution at the eighth time step this error had reduced to 0.03% indicating that the contribution of a particular convolution matrix at later times diminished and that an error introduced in any matrix attenuates quite rapidly. Introduction of a 5% error in the diagonal element of the convolution matrix affected more damage on the solutions. A

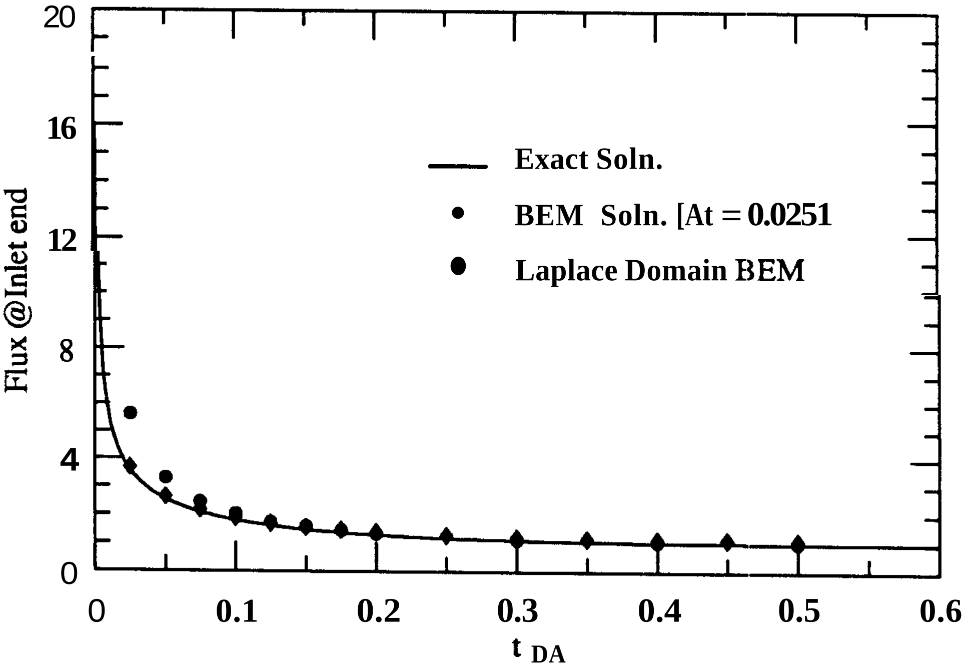


Fig. 4.7 Flux at the inlet end for the mixed problem

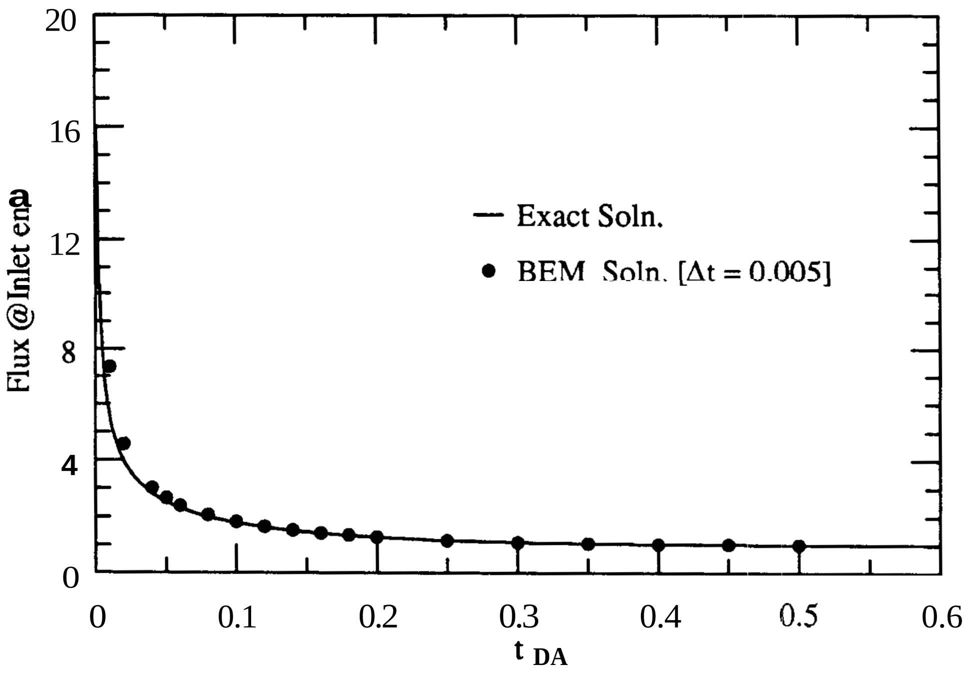


Fig. **4.8** Effect of time step refinement on the **flux** at the inlet end

19% error was seen at the third time step at the node where the error was introduced. It took much longer for the error to attenuate and at around the 19^{th} time step the error attenuated to about 0.015%.

The above analysis indicates that errors introduced in the matrices due to roundoff die out quite rapidly and that the error accumulation may not **be** a cause for concern. On the positive side, the solutions obtained are accurate.

The pressure response both at the inlet **and** outlet ends is shown in Fig. 4.9 **for** a constant flux inner boundary condition and a radiation type outer boundary condition. The boundary conditions **are**

$$\frac{\partial p_D}{\partial n}(0, t_{DA}) = 1 \tag{4.26a}$$

$$p_D(1, t_{DA}) + \frac{\partial p_D}{\partial n}(1, t_{DA}) = 0 \tag{4.26b}$$

The other two boundaries **are** closed. The exact solution for this problem is

$$p_D = (2 - x_D) - \sum_{n=1}^{\infty} \frac{2(\alpha_n^2 + 1)}{\alpha_n^2(2 + \alpha_n)} \cos(\alpha_n x_D)\, e^{-\alpha_n^2 t_{DA}} \tag{4.27}$$

where a,, are the roots of

$$\alpha \tan\alpha = 1 \tag{4.27a}$$

With a time step size of 0.1, the results matched the analytical solution well. If the inner boundary is at constant pressure, the fluxes **are** singular at early times at the inlet end, **and** are difficult to match with numerical schemes. Figure 4.10 shows the matches obtained with both the convolution and Laplace space **BEM.** The Laplace space solutions match the fluxes well, even at very short times.

A more practical example **is** shown in Fig. 4.1 1. Both convolution and Laplace space **BEM** were used to generate the linear aquifer influx functions presented by *Nabor and Barham* ***(1964).*** The convolution method shows a slight deviation at late times for closed outer boundaries. At early times when the outer boundaries have not been felt, the wellbore

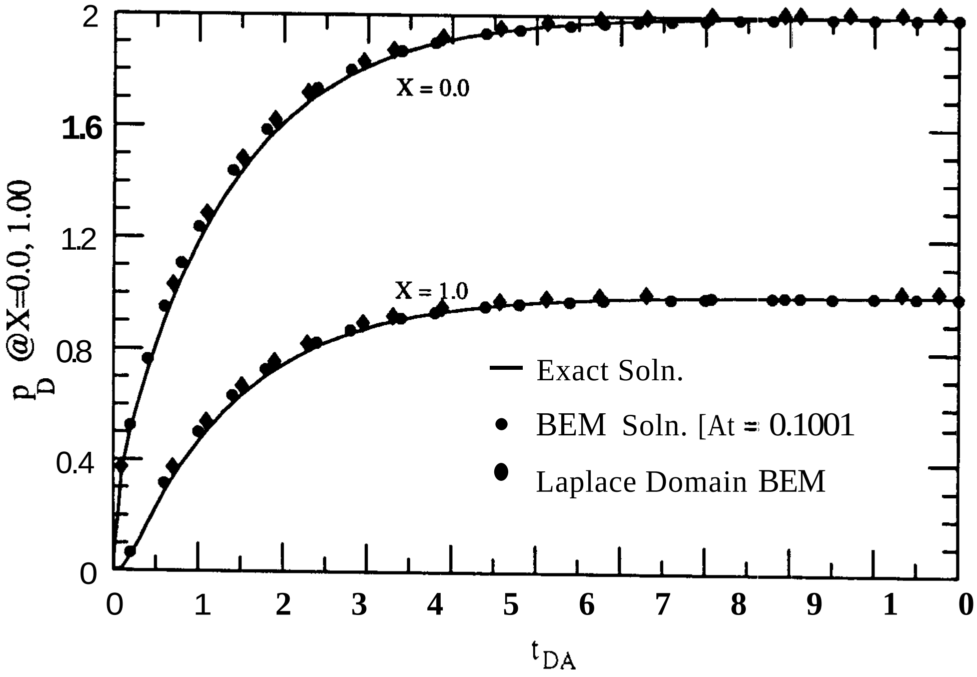


Fig. **4.9** Constant **flux** inner boundary and radiation outer boundary condition

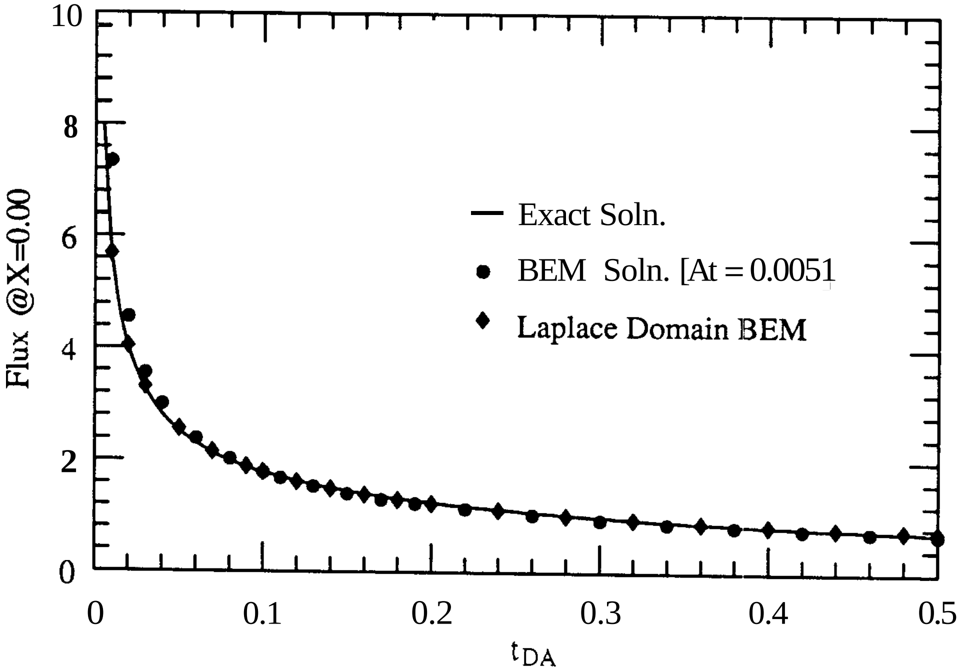


Fig. 4.10 Flux singularity at the inlet end for radiation-type **boundary** condition

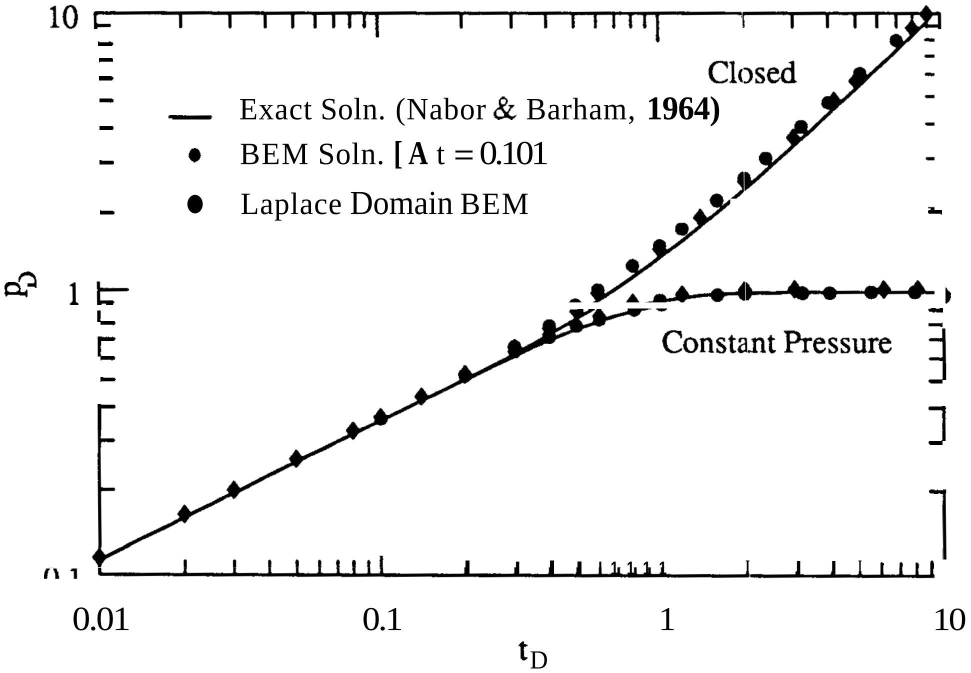


Fig. **4.1**1 Water influx functions for bounded linear aquifers pressure shows infinite acting behavior given by a half slope straight 'line on a log-log plot of dimensionless pressure versus time. Depending on the type of boundary condition the pressure deviates from infinite acting behavior to a boundary dominated regime.

Figure **4.12** is **an** example of solving the diffisivity equation with a time dependent inner boundary condition. **A** sinusoidally varying pressure pulse is imposed at the inner boundary at time zero. In other words

$$p_D \;=\; \sin 2\pi t_{DA} \qquad at \;\; x_D = 0 \tag{4.28}$$

The outer boundary is maintained at zero pressure $[p_D(x_D = 1) = 0]$. 'The pressures **are** monitored at three downstream **points** and matched to the analytical solution. The analytical solution is given by ***(Taigbenu and Liggett, 1985)***

$$p_D \;=\; \Big[\sin 2\pi t_{DA}\; (\sinh\alpha \;\cos a \,\sinh\beta \,\cos\beta + \sin\alpha \;\cosh a \,\sin\beta \,\cosh\beta)$$

$$+ \cos 2\pi t_{DA}\; (\sin\alpha \;\cosh\alpha \;\sinh\beta \;\cos\beta - \sinh a \;\cos a \,\sin\beta \;\cosh\beta)\Big]/D$$

$$- 4\sum_{k=1}^{\infty} (-1)^k \frac{k}{(k^2\pi^2 + 4)} \sin k\pi(1 - x_D)\; e^{-k^2\pi^2 t_{DA}} \tag{4.29}$$

and

$$D \;=\; \sinh^2\beta\cos^2\beta + \sin^2\beta\cosh^2\beta\;, \qquad \alpha \;=\; \sqrt{\pi}\,(1 - x_D)\;, \qquad \beta \;=\; \sqrt{\pi} \tag{4.29a}$$

The convolution BEM was used to solve this problem. **A** time step size of 0.025 was used. The results obtained are accurate **as** seen in Fig. **4.12. As** the pressure signal propagates in the medium the sinusoidal pulse diffuses and becomes out of phase with the original pulse. **This** example demonstrates the feasibility of handling time dependent inner boundary conditions by the boundary element method. ***Taigbenu and Liggett (1985)*** solved this problem by treating the solution at time, say t_1 **as** an initial condition for the subsequent time step. This requires a discretization of the domain to evaluate one area integral that is present. For the evaluation of the area integral the domain must be discretized in finite element type meshes. The solution presented here is by a purely boundary procedure.

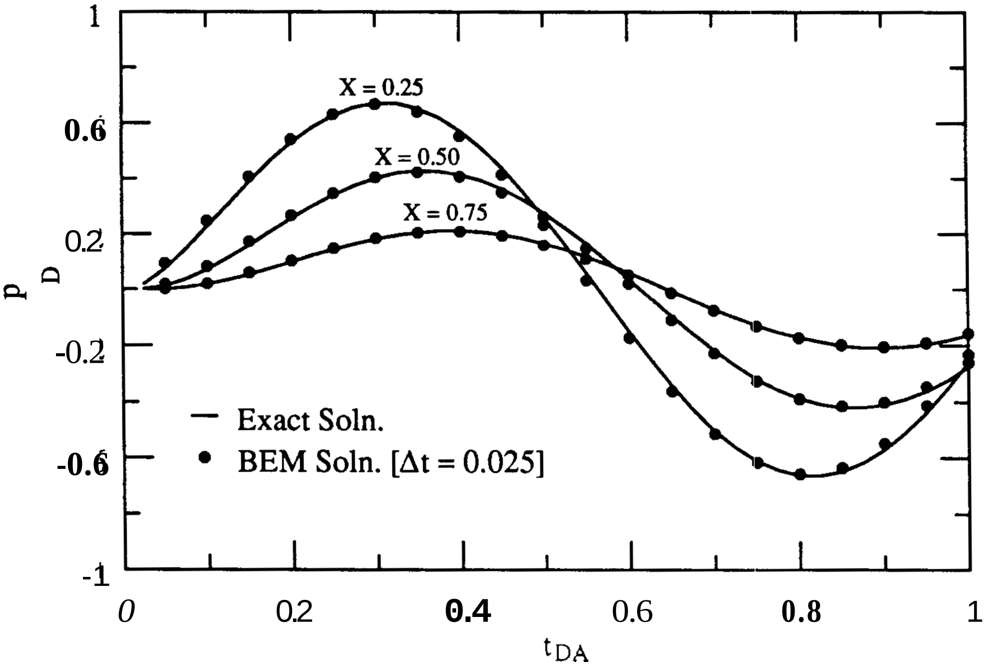


Fig. **4.12** Pressure transients due to sinusoidally varying inner boundary condition (Exact **Soln.,** after Taigbenu & Liggett,.1985)

Preliminary conclusions based on the above examples are as follows :

- The Laplace domain **BEM** is more accurate
- Small time step sizes are needed for the convolution **BEM** to match flux singularities and late time depletion type systems.
- In pressure transient solution where up to five or six log cycles of pressure data are usually needed, the requirement of a constant time step size in the convolution BEM poses a great computational burden.

Based on these observations, only the Laplace domain BEM was investigated further.

4.3.2 Application to Well Testing Problems

Figure 4.13 is a log-log plot of dimensionless pressure vs. time for a well producing at a constant rate from the center of a closed square of dimension $\sqrt{A}/r_w = 2000$. The solid lines are the analytical solution generated by superposition of an infinite array of wells ***(Earlougher et al, 1968).*** The solid circles are the boundary element solution with eight nodes on the boundary. A good match is obtained. Pressure at the wellbore and various interference locations are calculated. The error characteristics of the method with respect to the number of nodes are shown in Fig. 4.14. The maximum percentage error for four nodes, one each at the comer points of the square is about 3%. The absolute minimum number of nodes required to represent the geometry is four. As the number of nodal points are increased the solution becomes more accurate. The maximum error for 16 nodes is only about 0.3%.

Figures 4.15 and 4.16 show this effect for a well in the center of a circle. The BEM solution is compared with the ***van Everdingen and Hurst (1949)*** analytic solutions, for both constant pressure and closed external boundaries. The size of the reservoir is given by the ratio r_e/r_w. Representing the circle with eight nodes (i.e. an equivalent octagon) gives good results for the constant pressure external boundary where at late times steady state is reached (Fig. 4.15). For closed external boundaries however, eight nodes fail to give accurate solutions at late time (Fig. 4.16). Representing the circle with a hexa-decagon (16 nodes) gives

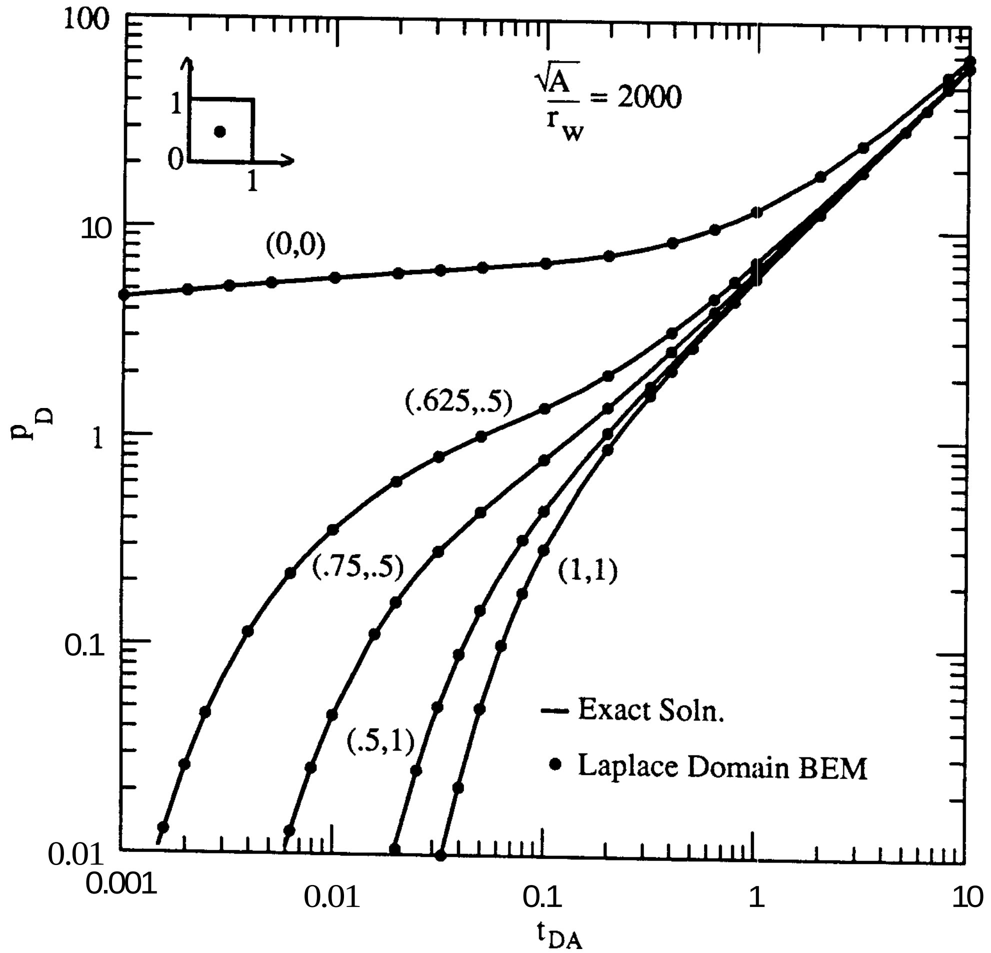


Fig. 4.13 Interference **and** wellbore pressure response for a well in the center of a closed square (Exact Soln., after Earlougher et al., **1968)**

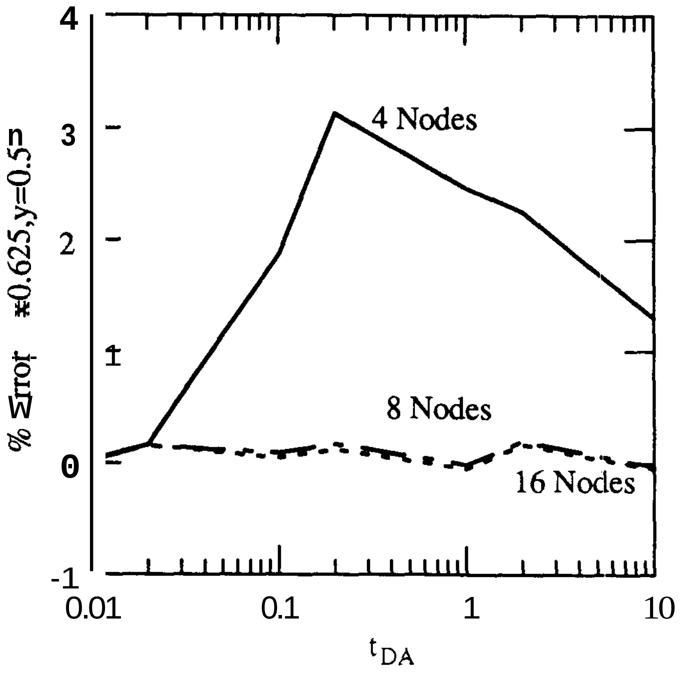


Fig. 4.14 Error in computations by Laplace domain boundary element method

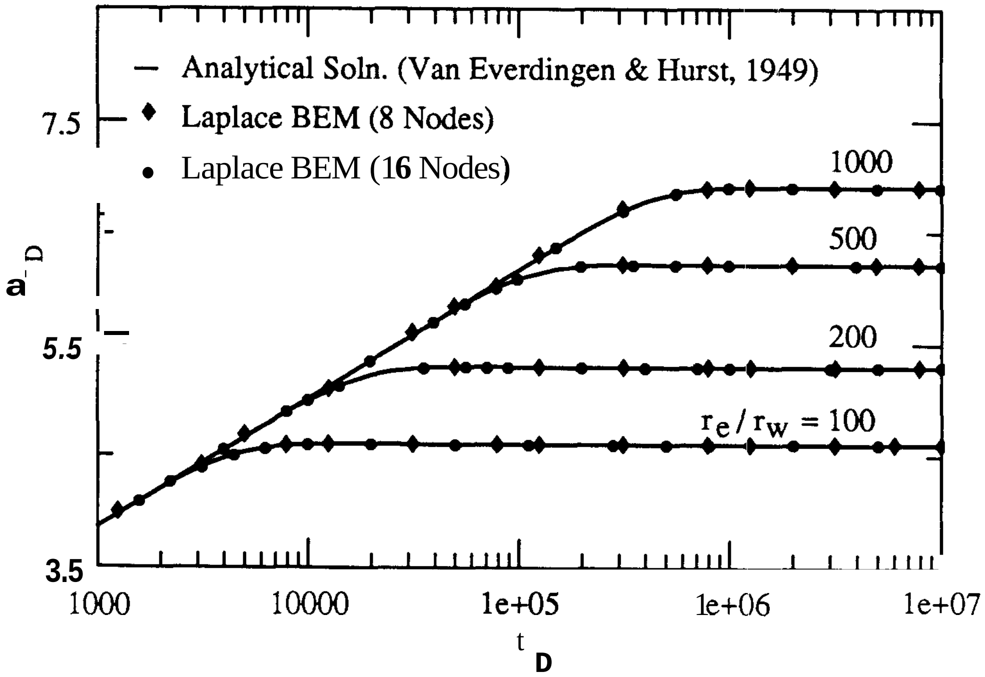


Fig. **4.15** Well pressure behavior for a constant pressure outer boundary in a circular reservoir

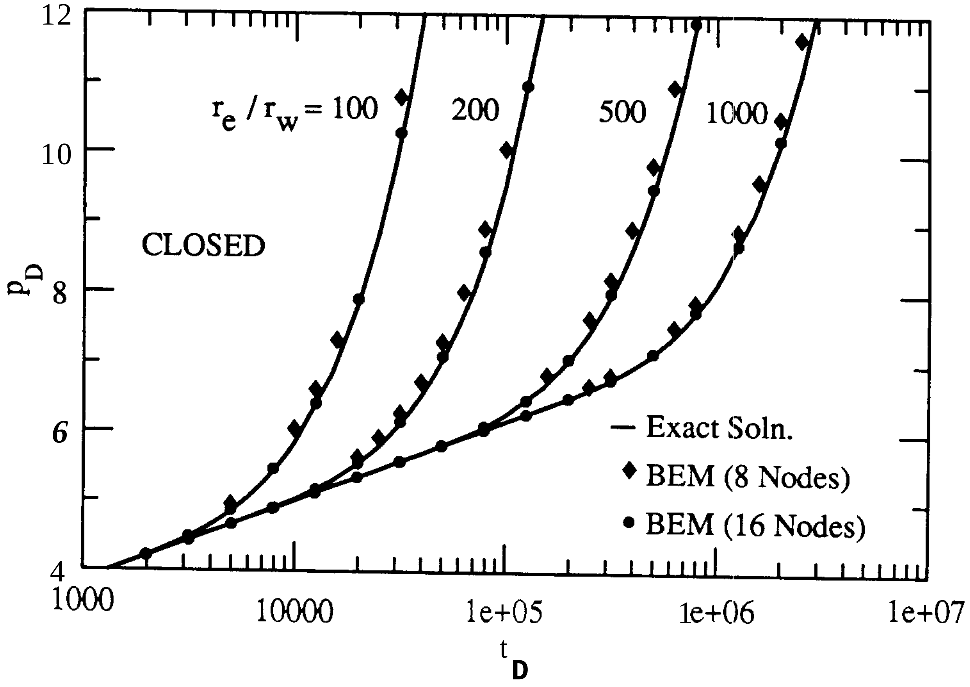


Fig. 4.16 Importance of accurate boundary representation for depletion systems (closed circular reservoir), [Exact Soln., after Van Everdingen & Hurst, 19491

results close to the analytic solution. This emphasizes the earlier observation that for depletion type systems the boundary mesh should be refined. This, in part, is due to the assumption of linear interpolation of pressure and fluxes between two adjacent boundary nodes. A higher order interpolation function (for example a quadratic) would alleviate this problem,

4.3.3 Use of Pressure Derivative

Since the advent of accurate pressure gauges for pressure measurements in well tests, the pressure derivative technique for analysis of the data has become prominent. The semilog pressure derivative is given by

$$p_D' = \frac{dp_D}{d\ \ln t_{DA}} = t_{DA}\frac{dp_D}{dt_{DA}} \tag{4.30}$$

The pressure derivative accentuates each of the flow regimes and thus aids in identification of flow regimes and parameters. Figure 4.17 shows a log-log plot of pressure and the semilog pressure derivative for a well producing at a constant rate from the center of a closed equilateral triangle of size $A/r_w^2 = 4 x\ 10^6$. The BEM solution for pressure matches well with the analytical solution (*Ramey* et *al,* 2973). The pressure derivative shows a value of 0.5 at early times, indicating infinite acting behavior. At late times the pressure goes into boundary dominated behavior (pseudosteady state) which is seen as a unit slope straight line on the plot. The results are shown in tabular form in Table 4.2. *Also* shown in the table are the values of the pressure derivative group.

Figure 4.18 shows a multiwell situation with two wells producing from within a 2:1 rectangle with one short side at constant pressure. The pressure is evaluated at the well closer to the constant pressure boundary. The pressure derivative graph shows that the system approaches pseudosteady state, but then the constant pressure boundary dominates and the derivative eventually goes to zero indicating steady state. The dimensionless pressures are compared to the analytical solutions given by ***Ramey*** et *al* (1973). Excellent agreement is obtained.

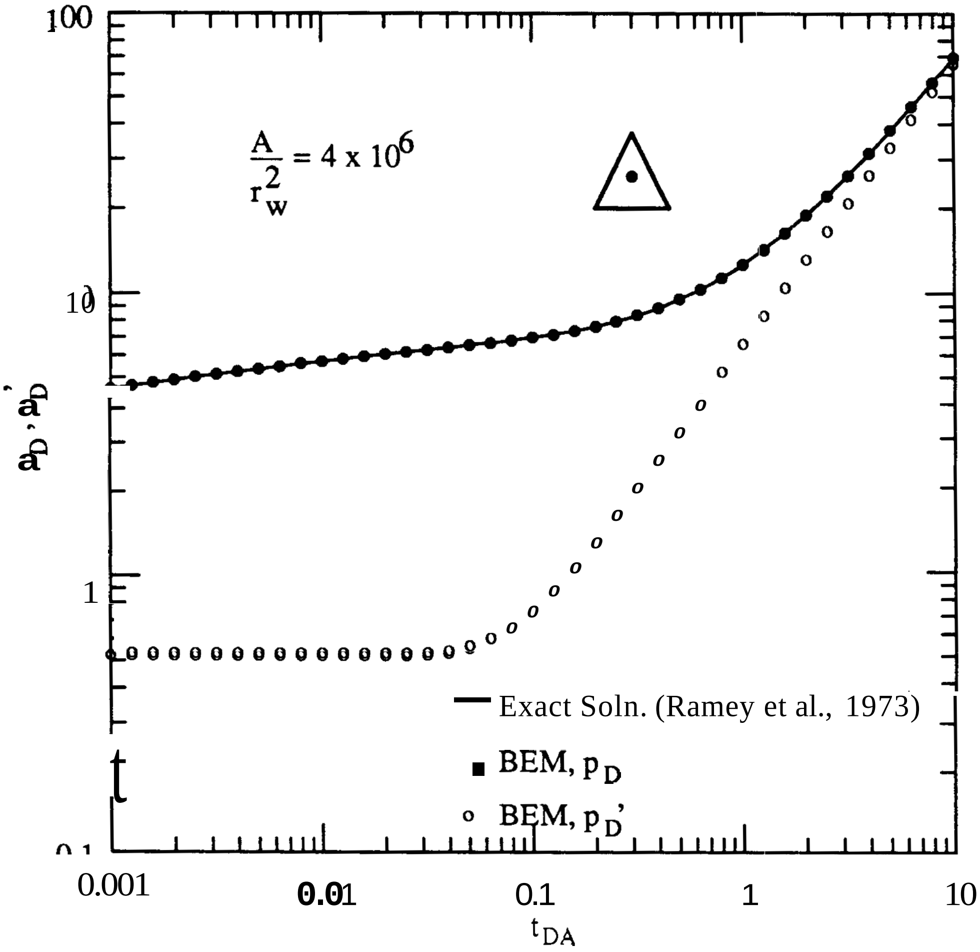


Fig. 4.17 Pressure **and** derivative response for a well in the center of **an** equilateral triangle

Table 4.2 Comparison of BEM and Analytic Solution for Well in the Center of an Equilateral Triangle

Dimensionless Time (t_{DA})	Anal. **Soln.** *Ramey et al*(1973)	Boundary Element	$t_{DA}\frac{dp_{wD}}{dt_{DA}}$
0.001	4.5516	4.5515	0.5000
0.002	4.8980	4.8969	0.5000
0.004	5.2447	5.2423	0.5000
0.008	5.5913	5.5876	0.5000
0.010	5.7029	5.7028	0.5001
0.020	6.0494	6.0483	0.4998
0.040	6.3981	6.3952	0.5119
0.080	6.7852	6.7780	0.6279
0.100	6.9371	6.9326	0.7188
0.200	7.6040	7.5916	1.2685
0.400	8.8649	8.8428	2.5002
0.800	11.3783	11.3327	4.9932
1.000	12.6349	12.6254	6.2861
2.000	18.9181	18.8819	12.5398
4.000	31.4844	31.3623	25.0179
8.000	56.6172	56.2652	49.9161
10.000	69.1836	69.1918	62.8404

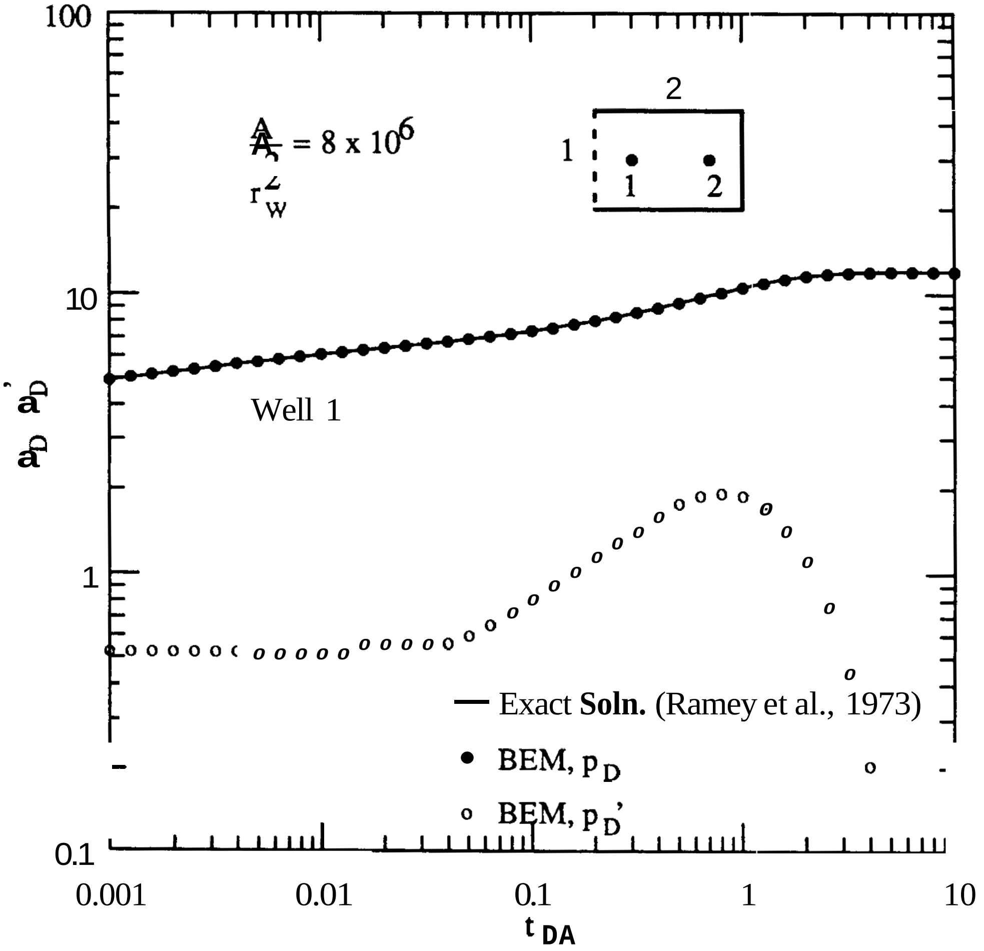


Fig. 4.18 Two-wells in a 2:1 rectangle with one short side at constant pressure

4.3.4 Special problems

The preceding examples were designed to show the efficacy of the BEM method by solving complex, but regular domain problems whose exact solutions can be found. The following examples illustrate the use of boundary element method for calculating shape factors for regular but complex geometries. Results highlight the advantages of using the Laplace space boundary element method.

Figure 4.19 is an illustration of a well in the center of a closed elliptical shaped reservoir. Though of the same area, the eccentricities of the two ellipses shown are different. Figure 4.20 shows the pressure and pressure derivative response at the well for the elliptical system of Fig. 4.19. As expected, after the early infinite acting period characterized by a slope of 0.5, there is a short transition before the external boundaries are felt and the system goes to pseudosteady state. The early and late time pressure derivative behavior for both eccentricities are the same. At intermediate transition times, the system with larger eccentricity deviates quicker from the semilog straight line and has a longer transition before reaching pseudosteady state. The reason for this is that radial streamlines are the most effective way of fluid transport to the wellbore. Any other geometry leads to longer flow paths and thus less efficient transport.

For regular geometries, a shape factor is defined based on the pseudosteady state response for closed reservoirs. The shape factor is a geometric factor characteristic of the system shape and the well location. During pseudosteady state the dimensionless pressure is given by

$$p_D = 2\pi t_{DA} + \frac{1}{2}\ln\left[\frac{A}{r_w^2}\right] + \frac{1}{2}\ln\left[\frac{2.2458}{C_A}\right] \tag{4.31}$$

where C_A in Eq. (4.31) is the shape factor. A shape factor for any regular shape may be determined from Eq. (4.31) if the pressure response and the area of the reservoir are known. Since the geometric variables in Eq. (4.31) are embedded within *log* terms, the shape factors are sensitive to p_D. Taking the semilog derivative of pressure in Eq. (4.31) we obtain,

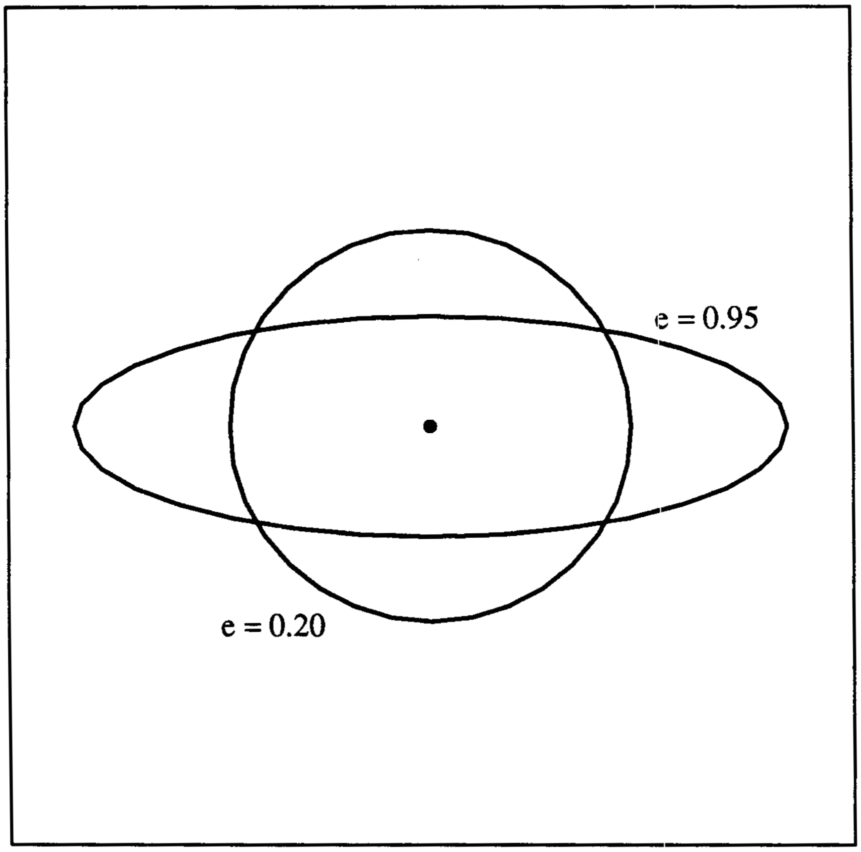


Fig. 4.19 Schematic of elliptical shaped reservoirs with different eccentricities

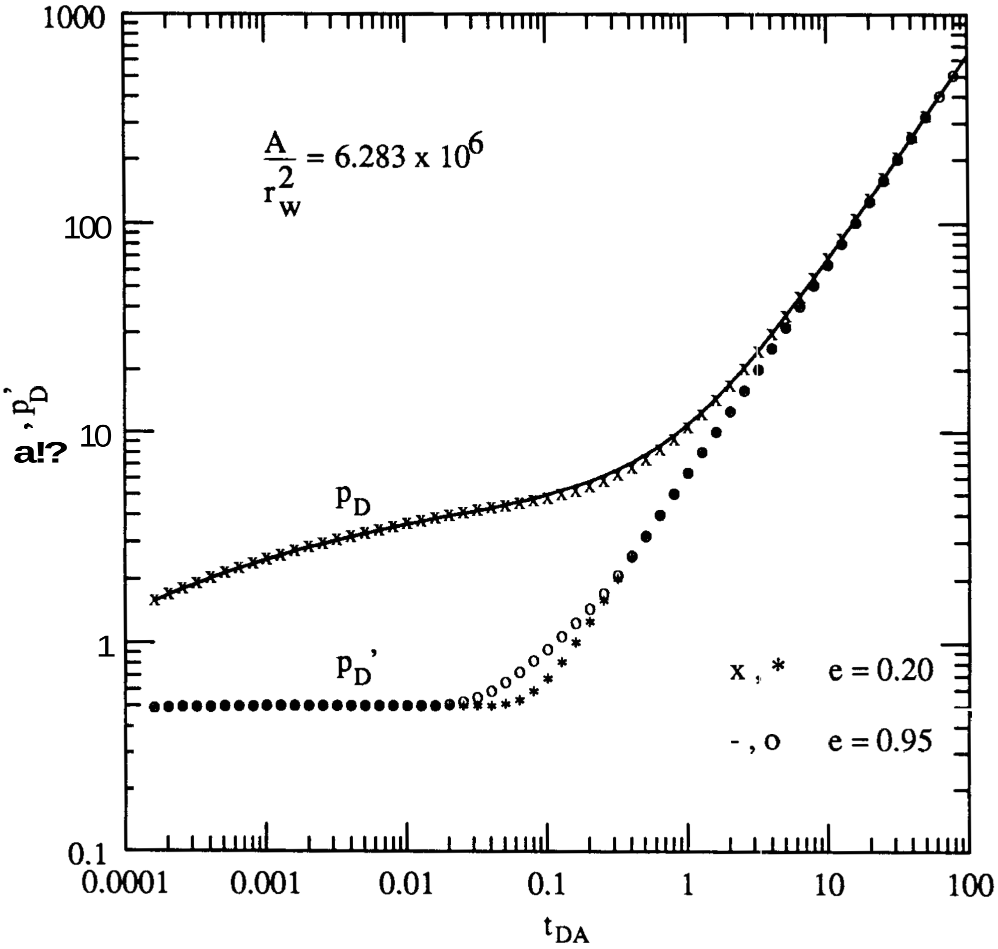


Fig. 4.20 Pressure and pressure derivative response for **closed elliptical reservoirs**

$$\frac{dp_D}{d(\ln t_{DA})} = 2\pi t_{DA} \tag{4.32}$$

The geometric terms are not present in Eq. (4.32). Replacing; $2\pi t_{DA}$ in Eq. (4.31) by Eq. (4.32), the following is obtained for the shape factor

$$\frac{1}{2}\ln\left[\frac{2.2458}{C_A}\right] = p_D - \frac{dp_D}{d(\ln t_{DA})} - \frac{1}{2}\ln\left[\frac{A}{r_w^2}\right] \tag{4.33}$$

Theoretically, determination of the shape factor from Eq. (4.33) is no different from that of Eq. (4.31). But, the results show that the accuracy is tremendously increased by using Eq. (4.33). This can be explained by considering the way numerical computations are performed. In Laplace space, $\bar{p}_D$ is calculated at each of the sampling points and summed. The pressure derivative in Laplace space is

$$\mathrm{L}\ \{\frac{dp_D}{dt_{DA}}\} = s\ \bar{p}_D \tag{4.34}$$

Thus, once $\bar{p}_D$ is calculated it is multiplied by the value of the Laplace parameter s. The derivative has similar computational errors as p_D. For this reason, subtracting the pressure derivative from pressure [Eq. (4.33)] to compute the shape factor leads to better results. Table 4.3 presents the shape factors determined both in the conventional way [Eq. (4.31)] and by Eq. (4.33). Results are shown for a well in the center of closed square, circle, equilateral triangle and ellipse. The calculated shape factors are closer to the analytical results when Eq. (4.33) is used.

Figure 4.21 shows the schematic of a kidney shaped reservoir. The well location is shown by a solid circle. Pressures are monitored at the wellbore and at two interference points shown in the figure by cross marks. The external boundary of the reservoir is closed. The results are shown in Fig. 4.22. The qualitative features lend credence to the correctness of the solution. The amount of computational effort is proportional to the number of elements required to represent the geometry adequately and does not depend on the complexity of the

Table 4.3 Comparison of shape factors for bounded reservoirs

Drainage **Shape**	Dietz Factor		Boundary Element Method			
			Eq. (4.29)		**Eq.** (4.32)	
	$\frac{1}{2}\ln(\frac{2.2458}{\hat{C}_A})$ (*)	C_A	*	C_A	*	C_A
Square	-1.3106	30.883	-1.2923	29.78	-1.3002	30.25
Circle	-1.3224	31.620	-1.3301	32.11	-1.3162	31.23
Eq. Triangle	-1.2544	27.600	-1.2410	26.8'7	-1.2495	27.33
Ellipse [Eccentricity = **0.8661**						
24 nodes			-1.2138	25.45	-1.1839	23.97
32 nodes		-	-1.2256	26.06	-1.1848	24.02

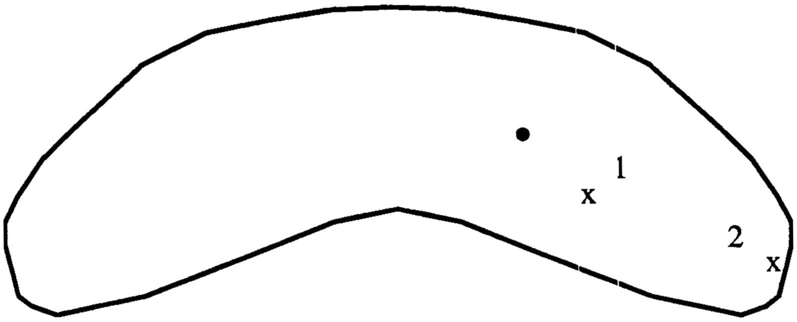


Fig. 4.21 Schematic of a kidney shaped reservoir

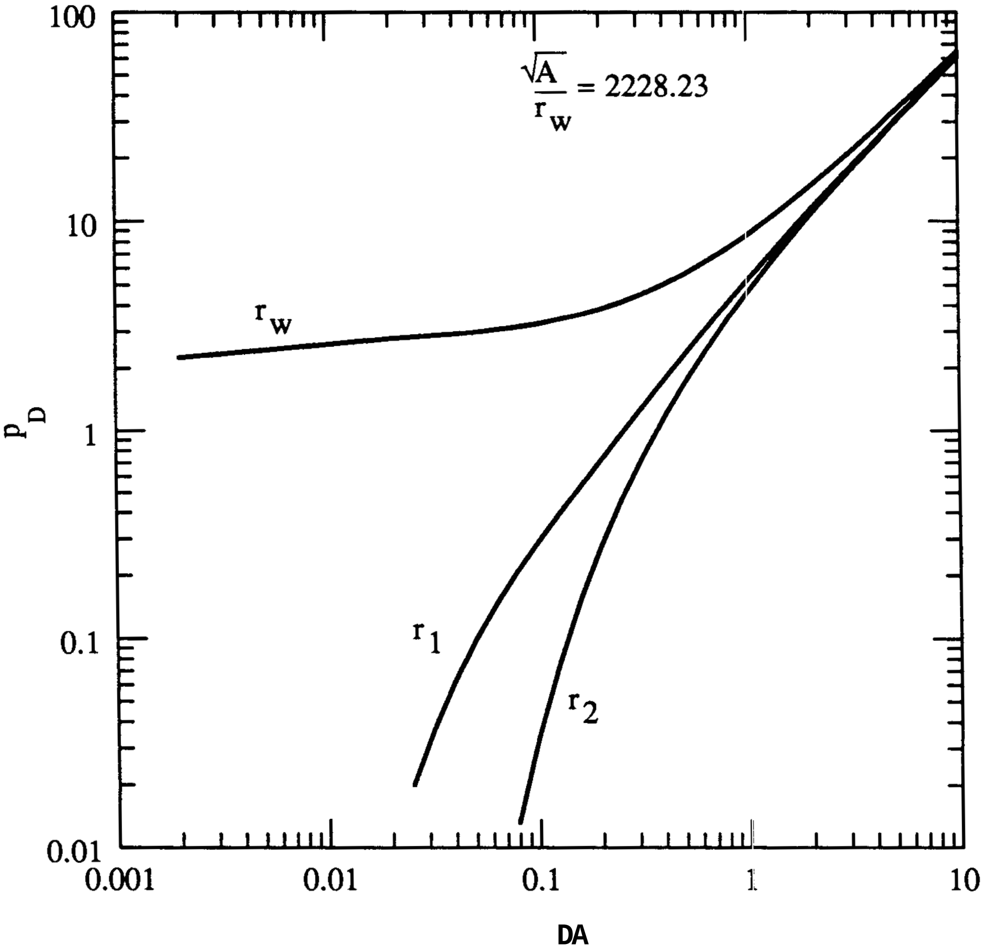


Fig. 4.22 Interference and **wellbore pressure response in a kidney-shaped reservoir**

geometry. In other words the computational effort in solving the kidney shaped reservoir would be the same as that for a circle if the number of boundary elements are the same.

A schematic of a multiwell production scheme in an irregularly shaped reservoir with mixed impermeable and constant pressure boundaries is shown in Fig. 4.23. There are four wells producing in the reservoir. The production rates of the wells are given in Table 4.4. Figure 4.24 shows the drawdown pressure transient and pressure derivative response in wells 2 and 3 respectively. The pressure behavior at the two wells is affected by the presence of a background trend owing to the production in rest of the wells. The presence of a constant pressure boundary makes the solution go to steady state at late times, because the Dirichlet type boundary condition dominates the behavior. At intermediate times, well 2 deviates from infinite acting behavior and attempts to go to pseudosteady state due to the presence of the closed boundaries around it, but at late times when the pressure support from the constant pressure boundary is felt the response goes to steady state, the derivative approaching zero. Well 3, on the other hand is closer to the pressure support and thus does not deviate as much as well 2 from the infinite acting behavior before going to steady state.

Another advantage of solving problems in Laplace space is the convenience of including wellbore storage and skin. The dimensionless wellbore pressure drop for a constant rate well with wellbore storage and skin is

$$p_{wD} = L^{-1}\left[\frac{1}{C_D s^2 + \dfrac{s}{S + s\bar{p}_D}}\right] \tag{4.35a}$$

where L^{-1} is the inverse Laplace transform operator. $\bar{p}_D$ in Eq. (4.35a) refers to the dimensionless wellbore pressure drop in Laplace space without storage and **skin.** The pressure derivative group with respect to time is

$$\frac{dp_{wD}}{d\ \ln t_{DA}} = p_{wD}' = L^{-1}\left[\frac{1}{C_D s + \dfrac{1}{S + s\bar{p}_D}}\right] \tag{4.35b}$$

Comparison of Eqs. (4.35a) and (4.35b) shows that a simple manipulation of the wellbore

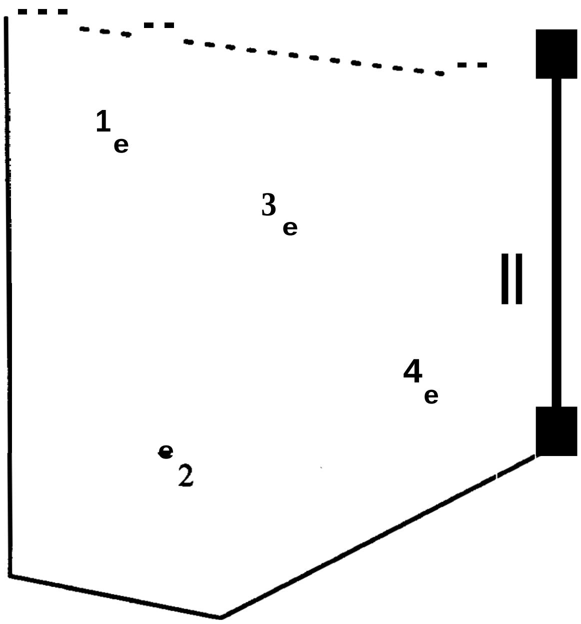


Fig. 4.23 Schematic of a simulated multiwell reservoir

Table 4.4 Well Flow Rates for the Simulated Reservoir

Well Number	Flow Rate (Q_D bpd)
1	150
2	**300**
3	*300*
4	100

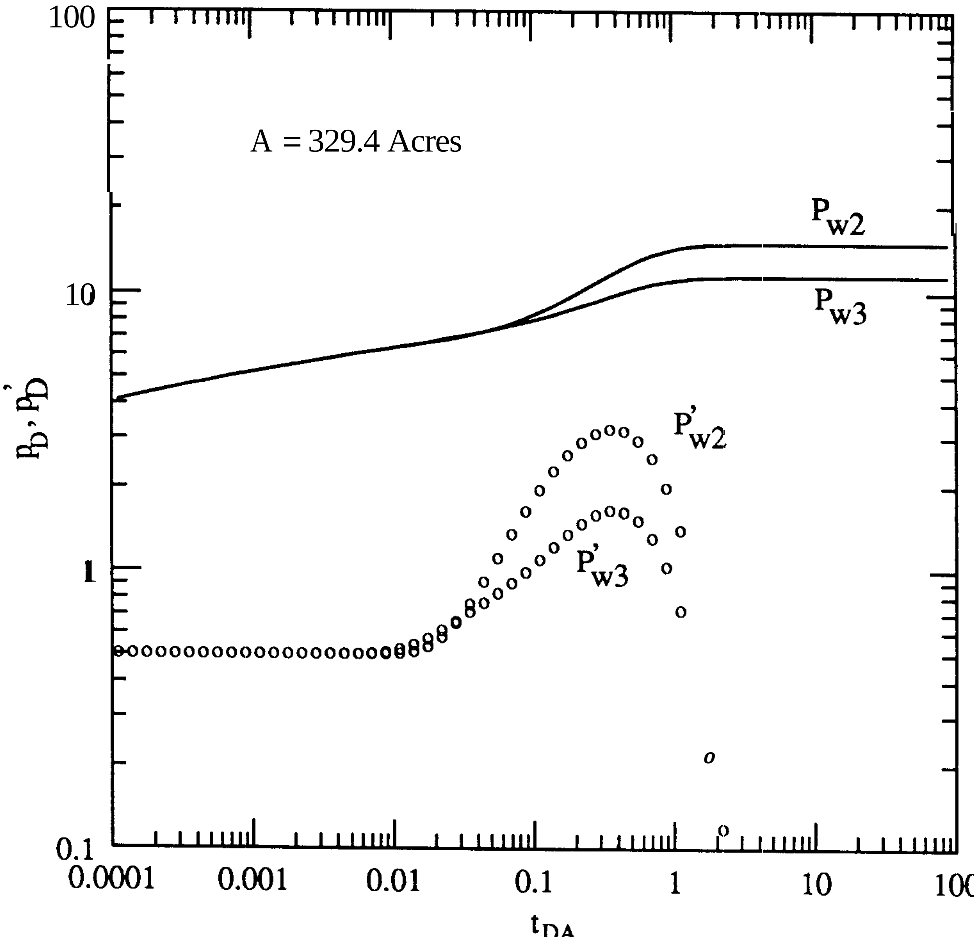


Fig. 4.24 Pressure and pressure derivative response ai: wells 2 and 3 for the simulated reservoir

pressure solution in Laplace space gives the pressure derivative solution and allows the inclusion of wellbore storage and skin.

Figure **4.25** shows the pressure derivative behavior of a well with wellbore storage producing from the center of a closed square reservoir. The solid lines are analytical solutions generated by superposition. The early time pressure behavior is masked by wellbore storage. **A** long transition follows before the **infinite** acting behavior is reached. Subsequently the outer boundary effects dominate the **response.** This is seen **as** a unit slope straight line on a log-log plot.

Production rate for constant pressure production can also be achieved by a simple manipulation of the pressure solution in Laplace space. **This** follows from an identity presented by Van ***Everdingen and Hurst, (1949).***

$$\bar{q}_D = \frac{1}{s^2 \bar{p}_D} \qquad (4.36a)$$

and the rate decline is

$$\frac{dq_D}{d\ln t_{DA}} = q_D' = t_{DA}\, \mathbf{L}^{-1}\left[\frac{1}{s\,\bar{p}_D}\right] \qquad (4.35b)$$

Figure 4.26 shows the rate and rate decline at a well producing at a constant pressure in a square reservoir with closed outer boundaries.

Simulation of pressure response for naturally fractured reservoirs can be accomplished easily by defining a resistance function. This resistance function represented by $f(s)$ is unity for homogeneous reservoirs. For naturally fractured reservoirs with pseudosteady state interporosity flow this function is

$$f(s) = \omega + \frac{(1-\omega)\lambda}{\lambda + (1-0)s} \qquad (4.37)$$

For transient interporosity flow with slab shaped matrix blocks and an infinitesimally thin skin (S_m) surrounding the matrix blocks, the resistance function is

$$f(s) = \omega + (\frac{\lambda}{3s})\,[\frac{\psi\sinh(\psi)}{\cosh(\psi) + \psi S_m \sinh(\psi)}] \qquad (4.38)$$

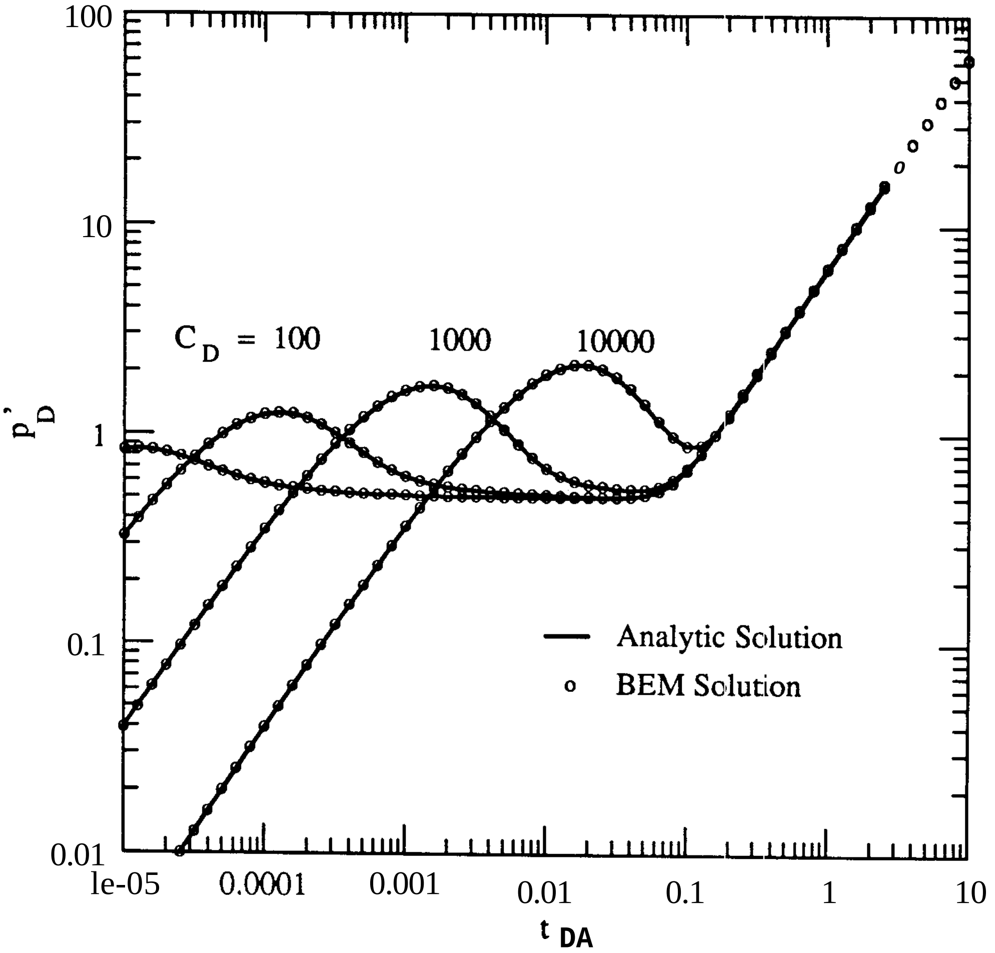


Fig. 4.25 Comparison of BEM and analytic solutions for pressure derivative at the wellbore in a closed square reservoir with wellbore storage

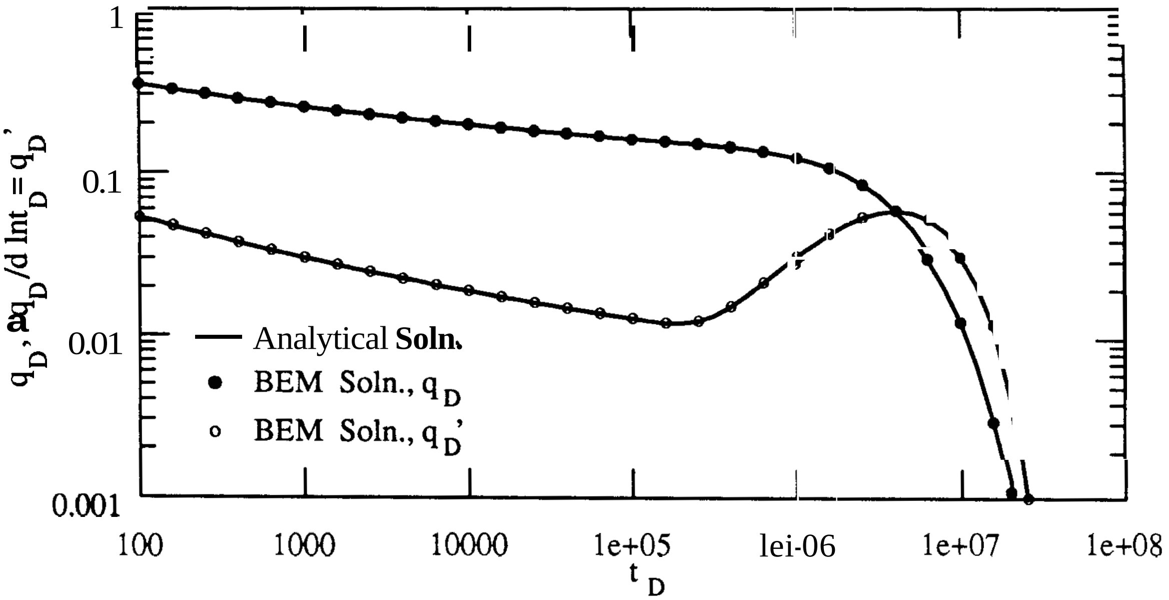


Fig. **4.26** Dimensionless rate and derivative group for a well in the center of **a** closed square where

$$\psi = \left[\frac{3(1-\omega)s}{\lambda}\right]^{1/2} \qquad (4.38a)$$

The parameters that govern the system behavior are the matrix to fracture interporosity flow coefficient (λ) and matrix to fracture storativity ratio (ω). The wellbore pressure for a naturally fractured reservoir is given by

$$p_{wD} = L^{-1}\{\bar{p}_D[sf(s)]\} \qquad (4.39)$$

where $\bar{p}_D$ is the Laplace space solution for a homogeneous reservoir and $f(s)$ is the resistance function defined in Eqs. (4.37) and (4.38).

From the preceding examples, it is evident that BEM is a powerful tool for evaluating solutions to arbitrary - shaped reservoirs with multiple sources and sinks. The Laplace domain boundary element simulator can be used to solve a wide variety of reservoir problems.

Chapter 5

SECTIONALLY HOMOGENEOUS RESERVOIRS

This chapter extends the application of boundary element method to problems which consist of simple heterogeneities in the flow field. In particular, the presence of internal boundaries and two-zone composite reservoirs are treated. The internal subregion can be an impermeable barrier or a constant pressure "hole". After a short introduction, the solution methodology for sectionally homogeneous reservoirs with the boundary element method is presented. Detailed implementation of the procedure is also shown. Next, the results are compared with the known analytical solutions and the important features of the method are discussed. Similarities and differences with homogeneous reservoirs are highlighted.

5.1 Introduction

Large scale features in reservoirs such as finite length faults, permeability barriers or different properties in front of and behind a flood zone significantly affect pressure transient response in the wells. Detection of such features or design of tests to include heterogeneities could be important in predicting long term performance. The parameters obtained by the analysis may also serve as input to a reservoir simulator. With better geological description of reservoirs becoming available, it is of relevance to be able to model the features in the flow field which may significantly affect the reservoir performance.

Two different types of problems are considered in this chapter. The Boundary Element method has been used to solve both problems. The first category of problems consists of wells producing **from** reservoirs which have permeability barriers and/or constant pressure "holes" of arbitrary size, shape **and** orientation. Permeability barriers occur as shale lenses or as structural features in a reservoir. These can also be created artificially due to the injection of low mobility and compressibility fluids. Constant pressure subregions ("holes") such as gas caps or limited extent recharge zones may occur in reservoirs. **Gas** injection for a number of EOR methods could create similar effects.

These problems have received limited treatment because of the difficulty of handling them analytically. The simplest manifestation of heterogeneity may be idealized as sealing boundaries in a flow field. The method of images has been used to study drawdown, buildup and interference tests for a well in a homogeneous reservoir containing single or multiple linear, sealing boundaries ***(Davis and Hawkins, 1963, Tiab and Crichlow, 1979, Tiab and Kumar, 1980, Streltsova and McKinley, 1984, Larsen, 1985).*** The pressure and pressure derivative behavior has been used to determine the distance to the sealing boundaries and the type of boundaries present ***(Davis and Hawkins, 1963, Prasad, 1'975). Prasad (1975)*** and ***Wong et al (1986)*** used a Green's function approach to generate multiple sealing boundaries. Multiple boundaries of finite dimensions have not been treated analytically because of the difficulty of posing the boundary conditions in an otherwise homogeneous flow field. ***Hantush and Jacob (1966)*** considered an eccentric well within a bounded aquifer with a leaky caprock. A single well producing external to a circular boundary has been solved by ***Sageev (1983).*** Computational complexities were encountered due to the nature of the solutions. ***Britto and Grader (1988)*** presented type curves for a well producing external to a circular or elliptic, no-flow or constant-pressure boundary. The well was located in an infinite reservoir. The superposition technique used is tedious and requires one to fix the strength of sources used for superposition such that the requisite boundary conditions are satisfied. This involves an iterative procedure. Domains with multiple barriers are difficult to treat with this method.

With the boundary element method, any number of sectional heterogeneities can be present in the flow field consisting of a number of line source wells (injectors/producers).

The second category consists of fluid injection problems which create mobility and storativity contrasts in the reservoir. Injection falloff tests performed in wells show distinct characteristics of both the injection and in-situ fluid provided the test is run long enough and that wellbore storage does not dominate the early time pressure behavior. Such systems may be treated as radial or elliptical composite reservoirs with different properties in the inner and outer regions. The interface between the two regions is assumed to be fixed in time and space. *Ramey (1970)* showed the validity of such an assumption. Radial composite models are used extensively to analyze well tests in water injection, steam injection, in-situ combustion and miscible injection projects. Extensive work has been done for such problems assuming both the inner and outer regions as perfectly radial and treating only one well producing from the center of the reservoir (*Bixel and van Poollen, 1967, Eggenschwiler et al, 1979, Ambastha and Ramey, 1987, van Poollen, 1965).* Analysis methods based on the pressure derivative have been shown to provide useful results. A number of methods have been proposed to calculate the location of the injection front from pressure transient data. A survey of the literature on this subject is given in *Ambastha (1988).*

In reality, injection fronts are never truly radial due to a variety of reasons. Background drift caused by injection and production wells nearby can affect the: geometry of the propagating front, and so can outer boundary effects. Also, fractured injection wells could cause the fluid front to move in an elliptical or a rectangular fashion for a certain distance in the reservoir before the effect of inner boundary geometry is dissipated, and the front becomes approximately radial. Gravity ovemde in steam injection projects also smear the radial geometry of the propagating front.

These two categories of problems are considered in the following, by the Boundary Element Method.

5.2 Mathematical Considerations

Development of the mathematical formulation of reservoirs with internal sub-regions and composite reservoirs is presented next.

5.2.1 Sectionally Homogeneous Reservoirs

Figure 5.1 shows a schematic of a reservoir with small scale features. These features tend to alter the flow characteristics of the fluids and consequently the pressure behavior. Examples are impermeable barriers or small recharge zones acting as constant pressure boundaries, or local concentration of injected gas near a wellbore. The constant pressure boundary is shown in Fig. 5.1 as a checkered region. The other two hatched regions represent impermeable bamers.

Mathematically, the problem domain is a multiply-connected surface, the flow through which is governed by the diffisivity equation for single-phase flow.

As mentioned earlier for a simply connected region (as in hcimogeneous reservoirs) the diffusivity equation is cast in an integral form in Laplace space. The boundary surface is discretized in a number of linear segments such that the boundary geometry is reasonably well represented. The linear segments or boundary elements, are numbered such that on traversing the boundary in a clockwise direction, the domain is always to the right. Clockwise numbering of nodes is not essential but a consistent method should be used. The procedure for solving the diffusivity equation on a multiply-connected surface follows the same concept, except that the integrals must be performed on the internal surfaces also. The interior sub-regions are numbered in a counter-clockwise direction to preserve the sense of the outward normal. The path integration scheme for one impermeable barrier in an otherwise: homogeneous reservoir is shown in Fig. 5.2. It demonstrates the fact that the reservoir on which the flow problem is solved is the annular region between the two surfaces. Both the internal sub-regions and the outer boundary are discretized in a number of boundary elements. The number of nodes on a boundary (Γ_i) is N_i.

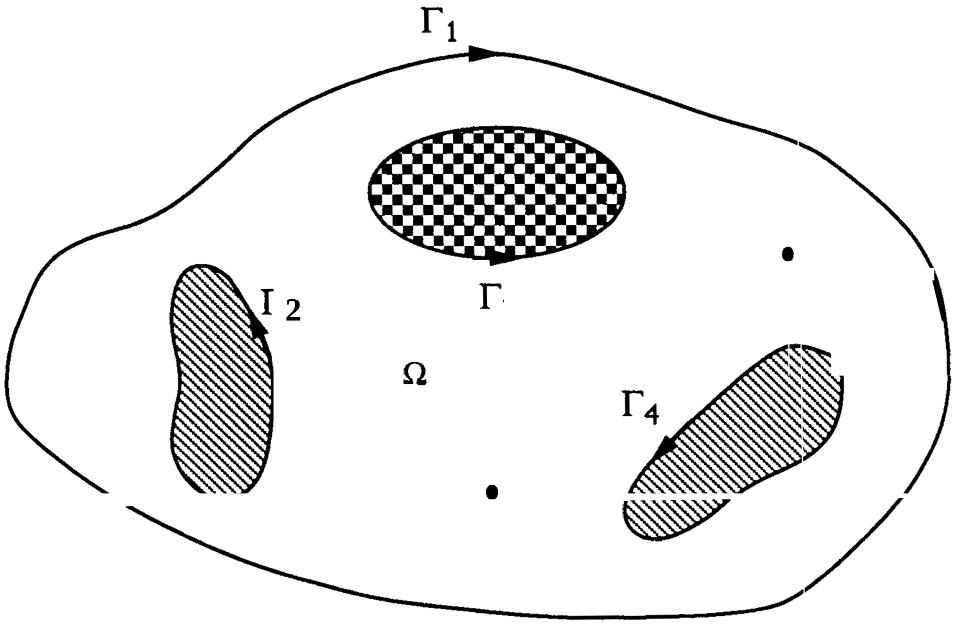


Fig. 5.1 Schematic of a reservoir with heterogeneities

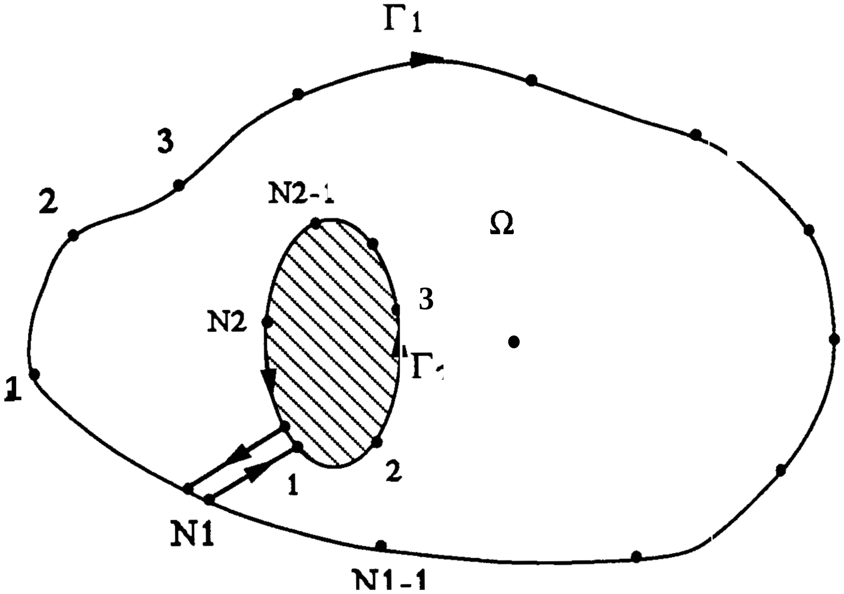


Fig. 5.2 Path integration scheme for a barrier in the reservoir

The integral equation for the pressure at **any** point within the reservoir (excluding the interiors of the subregions) in Laplace space is

$$\theta\, \bar{p}_D(x_D, y_D; s) \;=\; \int_{\Gamma_1 + \Gamma_2} (\bar{G}\frac{\partial \bar{p}_D}{\partial n} - \bar{p}_D \frac{\partial \bar{G}}{\partial n})\, d\Gamma \;+\; \sum_{i=1}^{n_w} \bar{G}_i\, Q_{D_i} \qquad (5.1)$$

$$\theta \;=\; 2\pi \qquad \text{if } (x_D, y_D)\ \varepsilon\ \Omega \qquad (5.2a)$$

$$\theta \;=\; \theta \qquad \text{if } (x_D, y_D)\ \varepsilon\ \Gamma_1 \qquad (5.2b)$$

$$e \;=\; 2\pi - \theta \qquad \text{if } (x_D, y_D)\ \varepsilon\ \Gamma_2 \qquad (5.2c)$$

where θ is the internal angle between two adjacent boundary elements. Eq. (5.2c) implies that the exterior angles between adjacent boundary elements should be considered for interior subregions.

$\bar{G}$ in Eq. (5.1) corresponds to the free space Green's function in Laplace space. $\bar{G}$ is given by Eq. (4.20).

To reiterate, the solution procedure is the same **as** outlined in Chapter 4 for the diffusion operator in homogeneous reservoirs, except that $N_1 + N_2$ equations are obtained in **as** many unknowns. The solution in Laplace space is sampled a number of times and inverted to real space by the use of the ***Stehfesr*** *(1970)* algorithm. With a few modifications the algorithm for solving the diffusivity equation in homogeneous reservoirs can be used for sectionally homogeneous reservoirs.

5.2.2 Two Region Composite Reservoir

Fig. 5.3 shows the schematic of a composite reservoir. The properties of the inner and outer regions are different. The interface between the two regions is assumed stationary. This assumption can be relaxed in an approximate sense and will be discussed later in this chapter. There are three parameters which govern the system belhavior. These are mobility ratio (M), storativity ratio (F_s) and the distance to the discontiriuity (R_D). The storativity ratio is

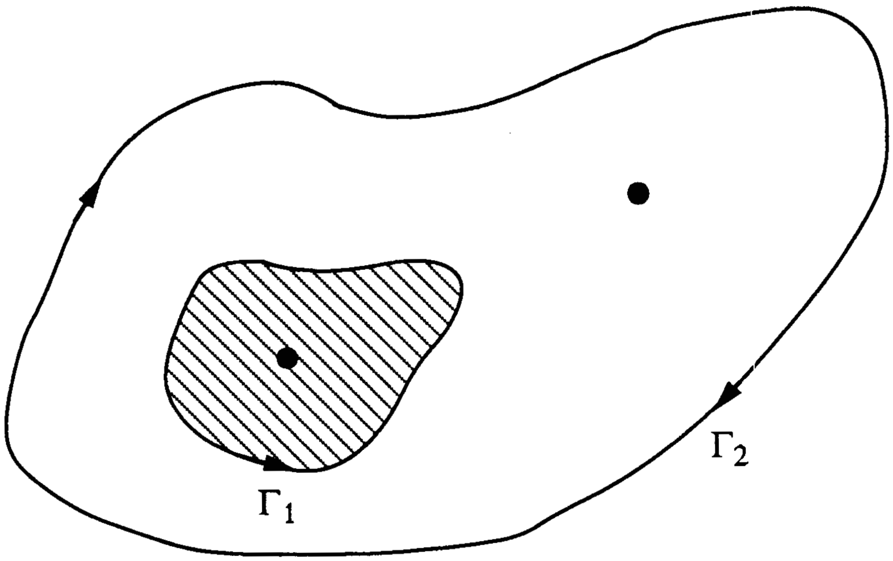


Fig. 5.3 Schematic of a composite reservoir

$$F_s = \frac{M}{\eta} \tag{5.3}$$

where η is the diffusivity ratio between the inner and outer **regions.**

The two regions can assume an arbitrary shape and can include multiple injection and production wells. The non-dimensional governing equations and boundary conditions **are**

Inner Region

$$\frac{\partial^2 p_{D1}}{\partial x_D^2} + \frac{\partial^2 p_{D1}}{\partial y_D^2} = \frac{\partial p_{D1}}{\partial t_{DA}} -$$

$$\sum_{i=l}^{\text{“wl}} QD\,1_i\ \delta(x_D - x_{D_i})\ \delta(y_D - y_{D_i}) \qquad [\ (x_D^2 + y_D^2 < R_D^2)\ 1 \tag{5.4}$$

Outer Region

$$\frac{\partial^2 p_{D2}}{\partial x_D^2} + \frac{\partial^2 p_{D2}}{\partial y_D^2} = \frac{M}{F_s}\ \frac{\partial p_{D2}}{\partial t_{DA}} +$$

$$M \sum_{j=1}^{n_{w2}} Q_{D2_j}\ \delta(x_D - x_{D_j})\ \delta(y_D - y_{D_j}) \qquad [\ (R_D^2 < x_D^2 + y_D^2 < R_{eD}^2)\ ] \tag{5.5}$$

Initial Condition

$$p_{D1}(t_{DA} = 0) = p_{D2}(t_{DA} = 0) = 0 \tag{5.6}$$

Interface Conditions

$$p_{D1}\ (\ x_{D1}^2 + y_{D1}^2 = R_D^2\ ) = p_{D2}\ (\ x_{D2}^2 + y_{D2}^2 = R_D^2\ ) \tag{5.7}$$

$$\frac{\partial p_{D1}\,(R_D)}{\partial n} = -M\frac{\partial p_{D2}\,(R_D)}{\partial n} \tag{5.8}$$

The negative sign in Eq. *(5.8)* accounts for the opposite sense of the outward normal in the two regions.

Exterior Boundary Condition

$$p_{D2}(R_{eD}, t_{DA}) = p_e \tag{5.9}$$

$$\frac{\partial p_{D2}}{\partial n}(R_{eD}, t_{DA}) = q_e \tag{5.10}$$

$$\alpha\, p_{D2} + \beta \frac{\partial p_{D2}}{\partial n} = \gamma_e \tag{5.11}$$

The dimensionless variables in the above equations are defined as

$$p_{D1} = \frac{p_1 - p_0}{\tilde{p}} \qquad p_{D2} = \frac{p_2 - p_0}{\tilde{p}} \tag{5.12}$$

$$x_D = \frac{x}{\sqrt{A_I}} \qquad y_D = \frac{y}{\sqrt{A_I}} \tag{5.13}$$

$$t_{DA} = \frac{\eta_1 t}{A_I} \qquad Q_{Di} = \frac{Q_i \mu_1}{k_1 \rho \tilde{p}} \tag{5.14}$$

A_I in Eq. (5.13) and (5.14) is the area of the inner region. Alternatively, the above equations can be normalized with respect to the exterior drainage area. $\tilde{p}$ is a reference normalization pressure and p_0 is the initial reservoir pressure.

Transforming Eqs. (5.4) through (5.11) to Laplace space and casting them into a boundary integral, we obtain the following integral equations :

$$\theta_1 \bar{p}_{D1}(x_D, y_D; s) = \int_{\Gamma_1} \left(\bar{G}_1 \frac{\partial \bar{p}_{D1}}{\partial n} - \bar{p}_D \frac{\partial \bar{G}_1}{\partial n} \right) dS + \frac{1}{s} \sum_{i=1}^{n_{w1}} \bar{G}_{1i}\, Q_{D_{1i}} \tag{5.15}$$

$$\theta_2 \bar{p}_{D2}(x_D, y_D; s) = \int_{\Gamma_1 + \Gamma_2} \left(\bar{G}_2 \frac{\partial \bar{p}_{D2}}{\partial n} - \bar{p}_D \frac{\partial \bar{G}_2}{\partial n} \right) dS + \frac{1}{s} \sum_{i=1}^{n_{w2}} \bar{G}_{2i}\, Q_{D_{2i}} \tag{5.16}$$

where θ_i has the same meaning as in Eq. (5.1). The $\bar{G}_\iota$ in Eqs. (5.15) and (5.16) denote the contributions of injector/producer wells at the Green's function source location. The details of the derivation for the Eqs. (5.15) and (5.16) are similar to the single region homogeneous reservoir and are given in Chapter 4.

The weighting functions or kernel functions (Green's functions) in the integral equations of Eqs. (5.15) and (5.16) are denoted by $\bar{G}_1$ and $\bar{G}_2$. The Green's function for the inner region is given by Eq. *(4.20).* The Green's function for the outer region is

$$\bar{G}_2(x_D, y_D, \xi, \zeta; s) = -K_0(|r_D| \sqrt{s \frac{M}{F_s}}) \tag{5.17}$$

where $K_0(z)$ is the modified Bessel function of second kind of order zero, and r_D is given in Eq. *(4.5)*

5.3 Implementation

The integral equations obtained in the previous section are solved discretely on the boundaries by the method of collocation with linear interpolation fimctions. Once the interpolation functions are defined, the integral equations are discretized in a local coordinate system in order to reduce the integrals in two-dimensions to a one-dimensional contour integral. A local coordinate system is sketched in Fig. *2.2.* The point i is the fictitious source point and ξ_j and ξ_{j+1} are the nodal locations for the element, on which the contribution of the source is evaluated. The fictitious source point is moved, in turn, to all the nodes and the respective contributions at all nodal locations are calculated. The procedure is the same as outlined in Chapters 3 and 4, the only difference being the discretization of all the surfaces instead of only the external surface. The discrete form of the integral equations (5.1), (5.15) or (5.16) is given symbolically in the local coordinate system as

$$0 = \sum_{j=1}^{N_1+N_2} \int_{\xi_j}^{\xi_{j+1}} d\xi \left[\bar{p}_{D_{j+1}} f_1(\xi, \zeta_i; s) + (1 - 2\theta_i \delta_{ij}) \bar{p}_{D_j} f_2(\xi, \zeta_i; s) + \right.$$

$$\left. \bar{p}_{Dn_j} g_1(\xi, \zeta_i; s) + \bar{p}_{Dn_j} g_2(\xi, \zeta_i; s) \right] \qquad i = 1, N_1 + N_2 \tag{5.20}$$

The integrals in Eq. (5.20) consisting of functions f_1, f_2, g_1, and g_2 are evaluated either analytically or numerically. Rewriting Eq. (5.20) in a matrix form we obtain

$$\sum_{j=1}^{N} a_{ij} \bar{p}_{D_j} + b_{ij} \bar{p}_{Dn_j} = 0 \qquad i = 1, N_1 + N_2 \tag{5.21}$$

where a_{ij} and b_{ij} are the elements of the augmented coefficient matrix. The coefficients a_{ij} and b_{ij} are exactly the same as those given in Eqs. (4.24a) and (4.24b). Equation (5.21) is then sorted for the unknowns. The resulting matrix equation for the unknown boundary conditions is

$$\sum_{j=1}^{N_1+N_2} H_{ij}u_j = G; \qquad i = 1, N_1 + N_2 \qquad (5.22)$$

where u is the solution vector and G_j the right hand side vector. Solution of Eq. (5.22) gives the nodal values of the unknown at the boundaries. The solutions in the interior are obtained by using the same discretized integral equation again for an internal. point.

The implementation of the integral equation for sectionally heterogeneous reservoirs is similar to that for homogeneous reservoirs. The only difference between the two is in the numbering of the nodes. For the multiply-connected surface, the sub-domains must be considered.

For two-region composite reservoirs, Eqs. (5.15) and (5.16) must be solved separately and coupled through the interface conditions. The boundary of the inner zone is divided into N_1 nodes and the exterior boundary is divided into N_2 nodes. Figure 5.4 shows the partition of the boundaries in the two regions. Region I consists of the fluid interface and the domain internal to it. Region II includes both the external boundary and the internal surface (interface). In other words region II is the annular region between the interface and the outer boundary.

Discretized equations as in *Eq.* (5.20) are written for both regions. The inner region is a simply-connected surface. The fictitious Green's function source is placed at each of the nodal points and the contributions at all other nodes are calculated in turn. This gives N_1 equations in $2N_1$ unknowns. There are more unknowns than equations because the boundary of the inner region is the fluid interface, or the discontinuity, and neither pressures nor the fluxes are known a *priori* on this surface.

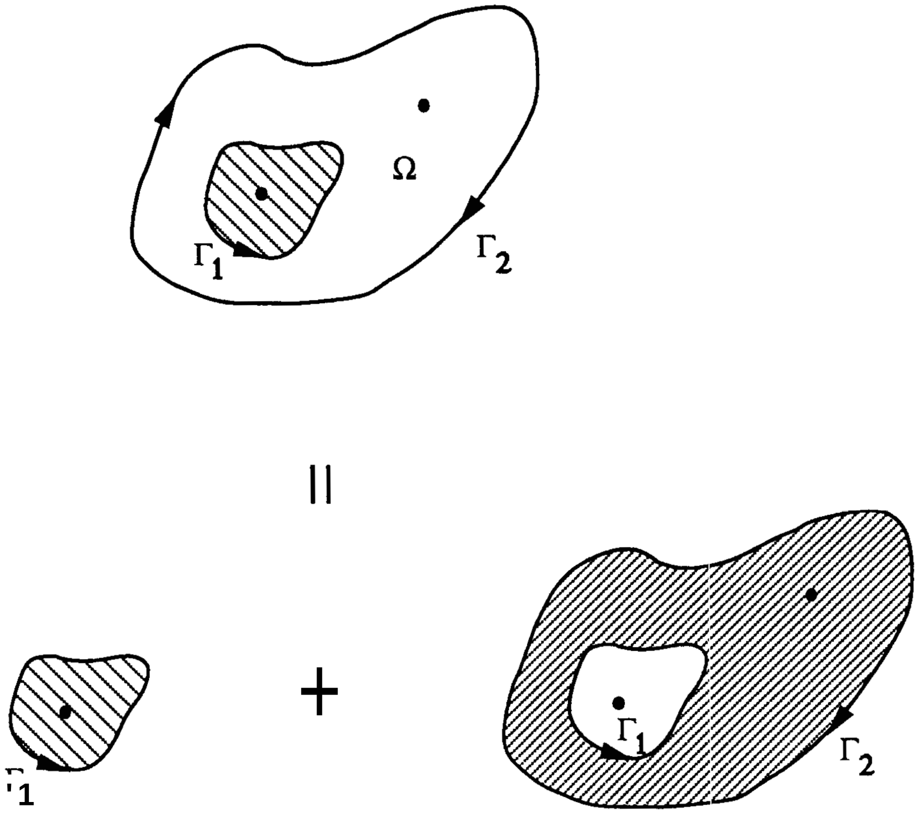


Fig. 5.4 Partition of reservoir **boundary** into two regions

Meanwhile, the outer region of the composite reservoir consists of a multiply-connected surface. The node numbering for the inner subregion in **this** case follows the sectionally heterogeneous reservoir case. The number of nodes in this region are $N_1 + N_2$. Thus, $N_1 + N_2$ equations in $N_2 + 2N_1$ unknowns are obtained. In Fig. **5.4 this** is shown as the region on the extreme right. Once the coefficient matrix for **this** region is compiled, the interface conditions are forced to be satisfied pointwise. Figure ***5.5*** shows the matrix structure resulting from constraining the solution to the interface conditions. The two rectangular sub-matrices are the coefficient matrices obtained from the boundary integral equations. These are completely filled. The other sub-matrices relate to the interface constraints and are diagonal. The blank spaces correspond to zero elements. These constraints may be eliminated in terms of common interface variables, reducing the matrix **size.** The structure of the reduced matrix is shown in Fig. ***5.6.*** The bottom left hand comer sub-matrix is a null matrix. On solving the matrix, the boundary unknowns both on the inner and outer regions are known. This implies that both the pressure and the velocity of the fluid interface are known, pointwise. **This** provides a way to move the front in time if required. In other words, the front may be tracked in time in an Eulerian sense, if it were moving as in an injection situation. Of course **this** would be an approximation because an injection problem truly is a moving boundary problem and belongs to a class of problems known as Stefan's problem.

Once the boundary solutions are known, the pressures can be calculated at any well **or** interference location, be it in the inner or outer region. **This** is done by re-solving the integral equation (5.15) or (5.16) depending on where the well is located. Since the boundary information in both pressure and fluxes is known, the problem reduces to quadratures.

5.4 Results and Discussion

Boundary element solutions for both internal boundaries and composite reservoirs were verified against known analytical solutions. Figure 5.7 is a schematic of **a** well producing external to **a** circular boundary such as a flow barrier or a constant pressure "hole." The

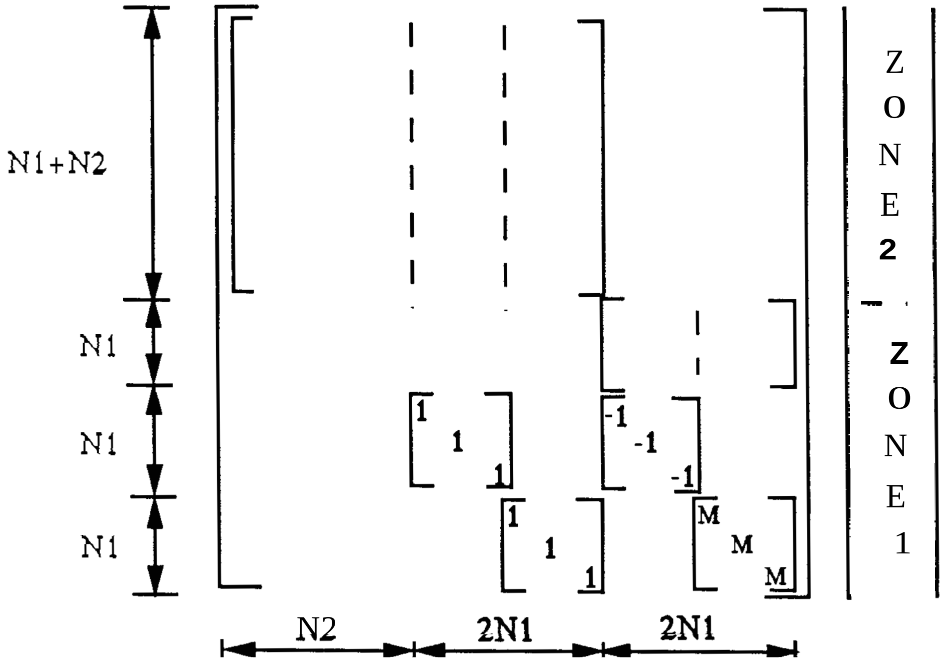
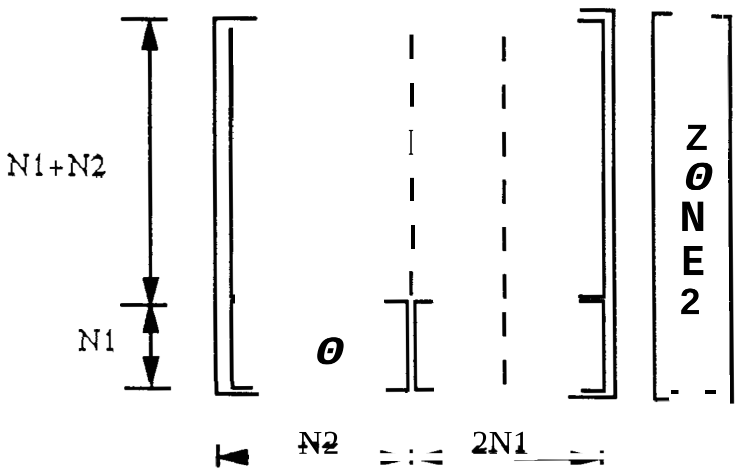


Fig. 5.5 **Matrix** structure for the composite reservoir



Fig, 5.6 Reduced matrix structure on removing the constraint equations

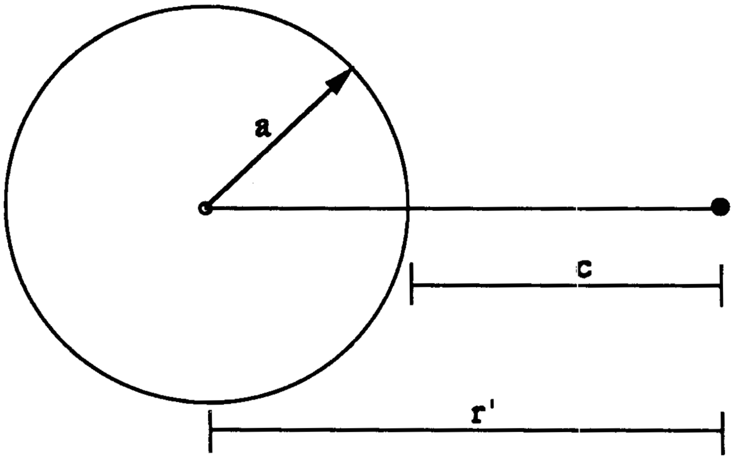


Fig. 5.7 Well producing external to an internal circular boundary

dimensionless parameters goveming the system behavior are F, the ratio of the radius of the internal circular boundary, a, to its distance from the well, r' $(F = \frac{a}{r'})$, and c which is the distance from the well to the closest part of the internal boundary. Figure *5.8* shows a semi-log plot of dimensionless pressure vs. time for c = 250 and values of F ranging from 0.1 to 0.9. In the limit when F = 0, the system behaves like a line source solution. On the other extreme F = 1, shows the behavior of well near a linear fault. Both closed and constant pressure internal boundaries are considered. The analytic solution was given by *Sageev* (1983). The analytic solution is composed of infinite series of modified Bessel functions $[I_n(x), K_n(x)]$ of all integer orders. Table **5.1** compares the boundary element solution with the analytic solution for F = 0.5 and c = 250. The results are within 0.5 % for $t_D \leq 8 x \ 10^5$. Log-linear instead of linear interpolation between the computed values to obtain pressures at the times shown in the table, would show even smaller errors.

Figure **5.9** is a log-log plot of dimensionless semilog pressure derivative vs. time corresponding to Fig. *5.8.* It is instructive to consider the derivative plot. At early times, the derivative group shows infinite-acting behavior given by a slope of 0.5. As the presence of the internal boundary is felt, the derivative group deviates from infinite-acting behavior for a short time. Presence of more fluid in the system results in another period of infinite-acting behavior. The pressure derivative at late time becomes a constant and the value depends on the size and distance of the boundary from the active well. If the internal boundary is at constant pressure, the derivative group dips below the infinite-acting behavior when the presence of pressure support becomes evident. However the derivative group does not go to zero as the system is infinitely large and the well can deplete the reservoir farther away.

Figure 5.10 is a schematic of a well producing external to inipermeable elliptic internal boundaries of various sizes. The major axis and the minor axis are labeled a_D and b_D, respectively. The minimum distance from the well to the internal boundary is shown as $r_{D,\min}$. Figure 5.11 shows a log-log plot of dimensionless pressure vs. time for a constant value of the ratio $\frac{b_D}{a_D} = 0.25$. $r_{D,\min}$ is chosen to be 100. The sizes of the ellipses are

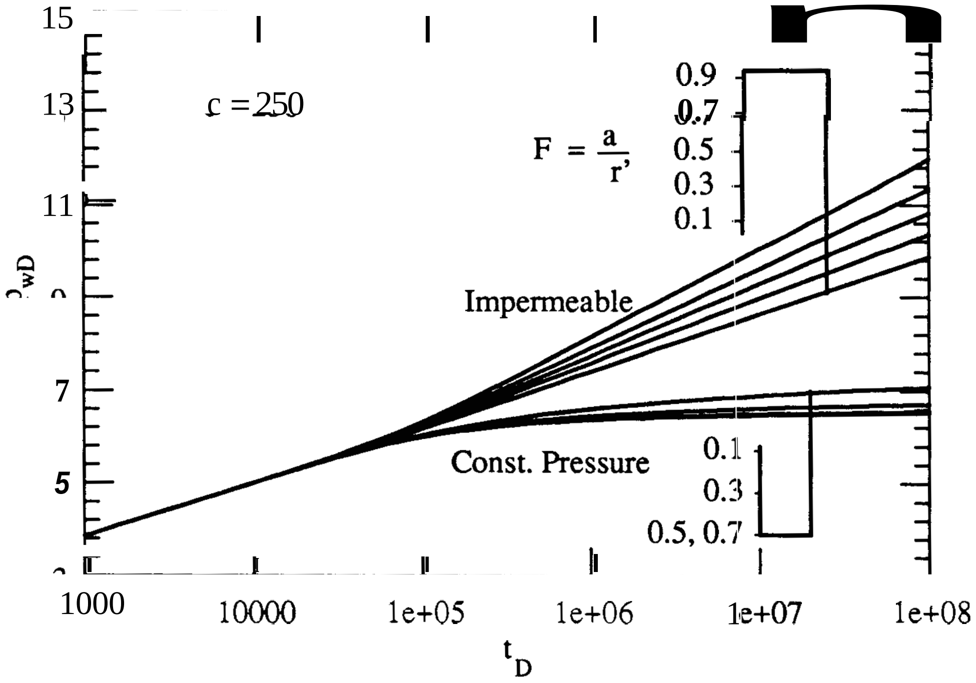


Fig. 5.8 Pressure response at a well producing external to a circular sub-region

Table 5.1 Comparison with Analytical Solution (Circular Impermeable Barrier)

Dimensionless Time (t_D)	p_{wD} Analytical (Sageev,1983)	p_{wD} Boundary Element (16 nodes)	Abs. Percent Error
4.0e+0	1.1283	1.1289	0.053
8.0e+0	1.4597	1.4590	0.048
4.0e+1	2.2520	2.2527	0.031
8.0e+1	2.5970	2.5963	0.027
4.0e+2	3.4005	3.4012	0.021
8.0e+2	3.7469	3.7462	0.019
4.0e+3	4.5515	4.5510	0.011
8.0e+3	4.8981	4.8974	0.014
4.0e+4	5.7288	5.7257	0.054
8.0e+4	6.1299	6.1246	0.086
4.0e+5	7.0921	7.0400	0.738
8.0e+5	7.4739	7.4578	0.215

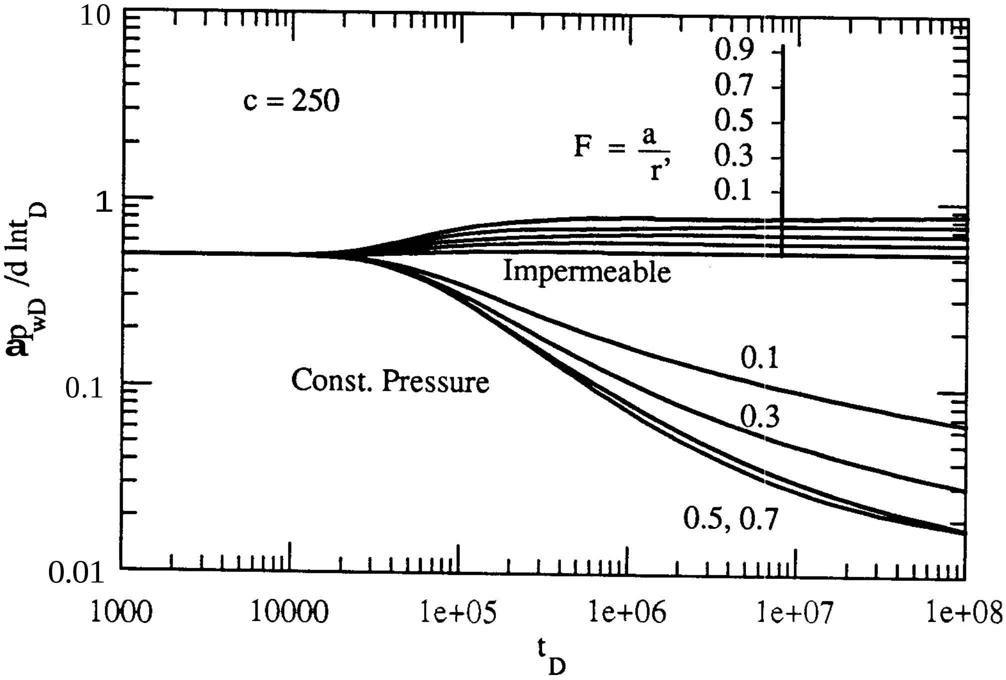


Fig. 5.9 Pressure derivative behavior at a well producing external to a circular sub-region

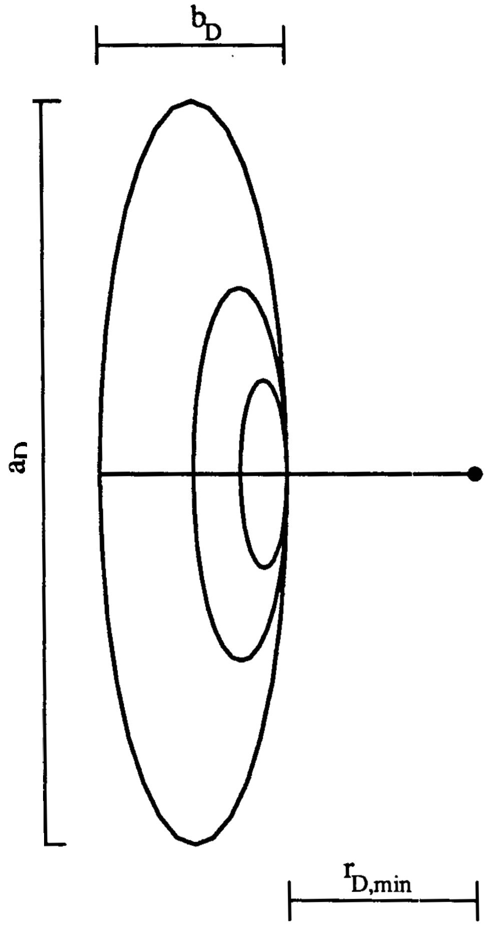


Fig. 11: Schematic of a well external to impermeable elliptical sub-regions of different sizes

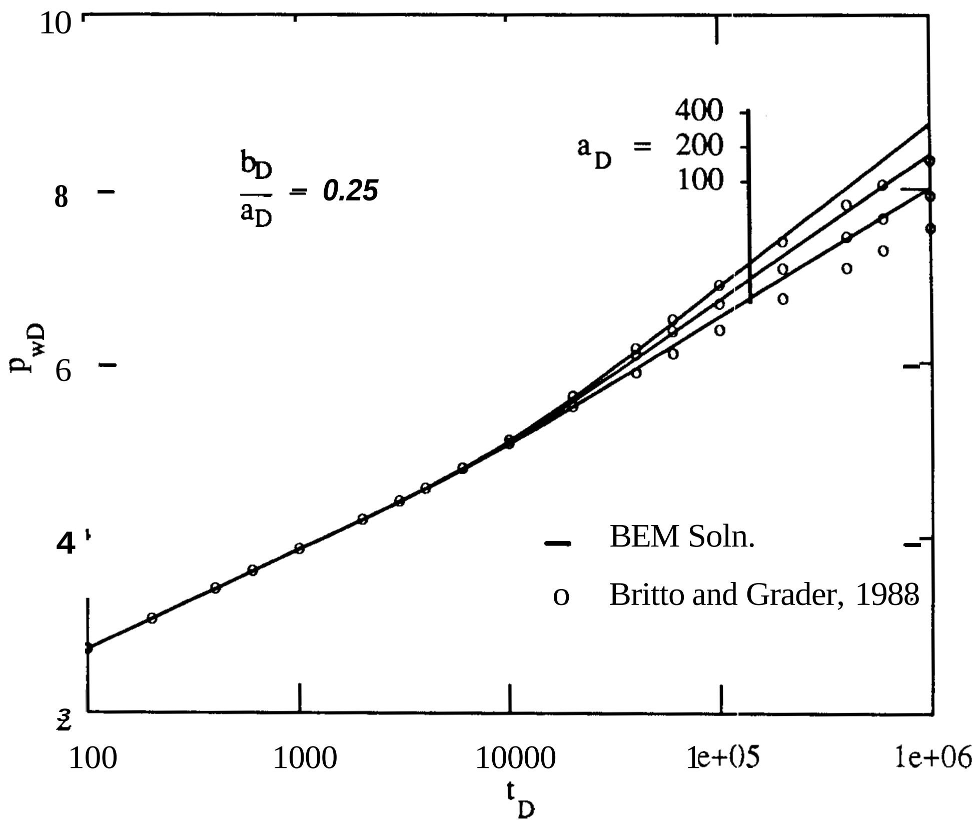


Fig. 5.11 Pressure response at a well producing external to an impermeable elliptic sub-region

shown in Fig. **5.10.** *Britto and Grader (1988)* used an iterative procedure as described earlier to generate pressure transient solution for elliptical internal boundary. Their solutions are represented as open circles in Fig. **5.11.** At late times, the two solutions do not match well. The reasons for this could not be determined due to lack of detailed information about possible errors associated with *Britto* and Grader's *(1988)* solution.

Composite Reservoirs

Figure **5.12** shows the effect of mobility (M) and storativity (F_s) ratios on the semilog pressure derivative behavior of an infinite circular composite reservoir. The storativity ratios chosen cover a spectrum of values. The time axis is scaled by the front radius

$$t_{De} = t_D / R_D^2 \tag{5.23}$$

Early time radial infinite acting behavior corresponding to the inner region is seen as a constant value 0.5 on the pressure derivative plot. Depending on the value of storativity ratio, the pressure derivative deviates from infinite acting behavior. The derivative passes through a maximum and goes to another radial infinite acting behavior, corresponding to the external region. The pressure derivative group is given by a value of M/2, where M is the mobility ratio of the inner to the outer region. The solid lines in Fig. **5.12** are the analytical solution ***(Ambustha,** 1988)* and the circles are the boundary element solution. Figure **5.13** shows the effects of a closed external boundary on the pressure transient response of a two-zone circular composite reservoir with a mobility ratio of **10** and storativity ratio of 1000. The composite reservoir effects are not seen if the ratio R_{eD}/R_D is less than 10. For R_{eD}/R_D of 1000, the infinite-acting behavior due to the outer region can be seen before pseudosteady state begins, which appears as a unit slope straight line on the derivative plot.

The results in Figs. **(5.12)** and **(5.13)** show the accuracy obtained with the boundary element method. For such good results the geometry of the inner region must be defined accurately. The reason for this will be discussed later.

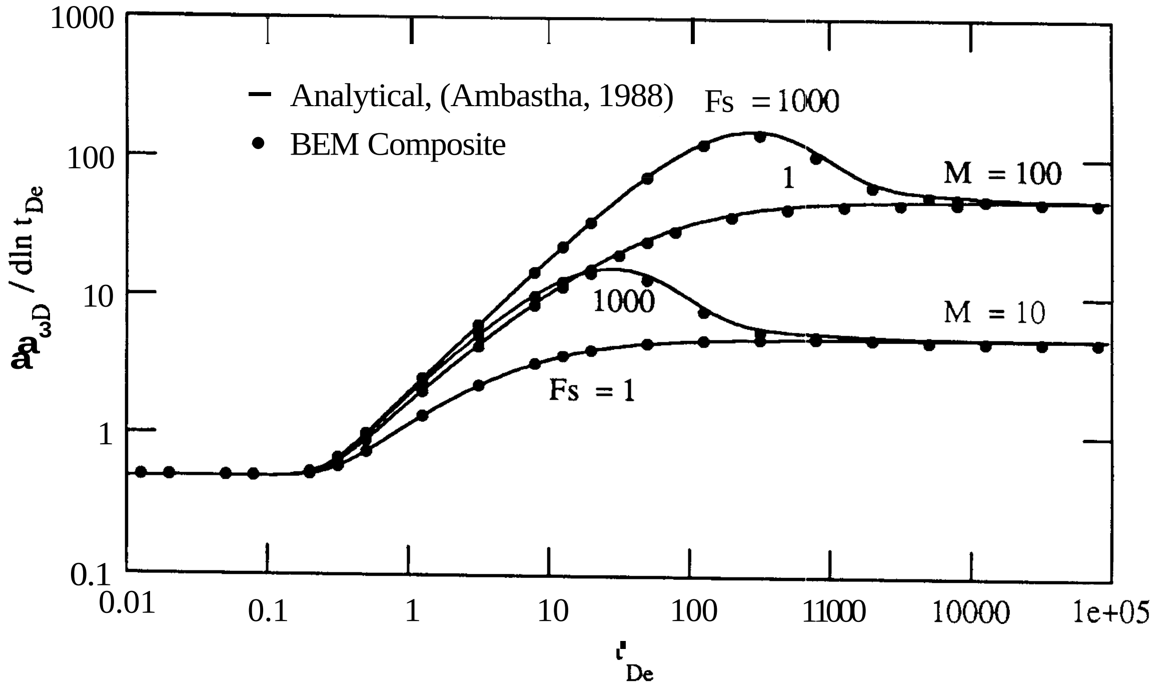


Fig. 5.12 Effect of mobility and storativity ratio on tlhe derivative response of a radial composite reservoir

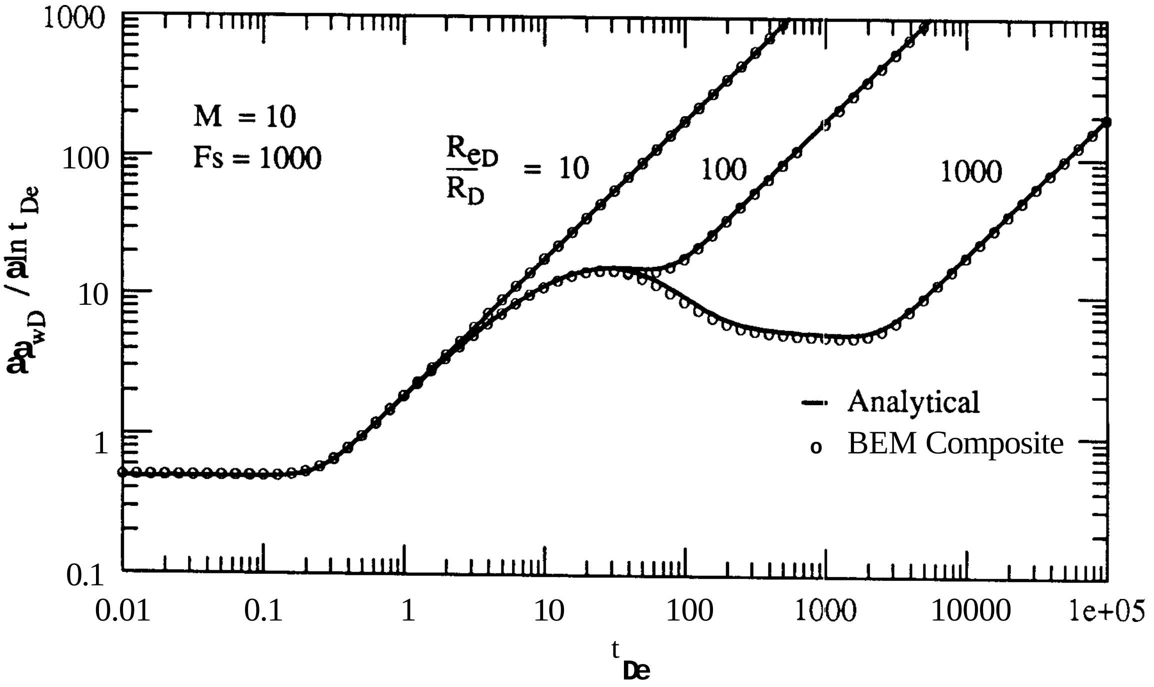


Fig. 5.13 Effect of external boundary on the derivative: behavior of a radial composite reservoir (Analytical Soln., after Ambastha, 1988)

Field data suggest that there are thermal injection wells which are intercepted by **a** vertical fracture. In such cases, the displacement **front** will move in a direction normal to the plane of the fracture at early times. The swept region at early times can be idealized as a low width to length ratio rectangle. Figure **5.14** shows the dimensionless pressure vs. time based on length **of** the inner zone for various width to length ratios of the inner swept region. The mobility and storativity ratios used were **200** and **16.67** respectively. ***Teng (1984)*** studied a similar problem for gas injection in a vertically fractured well with a finite difference simulator. The nonlinearities caused by the **gas** flow were included in. the study. The solution matches qualitatively with Teng's results.

Computational Effort

The computations for composite reservoirs require substantial computing effort. At every time step an $(N_2 + 2N_1)$ by $(N_2 + 2N_1)$ matrix is inverted six. to eight times, depending on the number of sampling points in Laplace space. In addition, the geometry of the inner region must **be** represented accurately. For example, for a radial composite reservoir the phase discontinuity is a circle. If the circle is represented by a hexagon (six nodes), the pressure behavior is affected significantly at late times, because the isopotential lines are configured so that they are hexagons instead of circles.

No effort has been made to optimize the code for speedup. However, considerable improvement could **be** obtained by eliminating repetitive calculations, in favor of increased storage. The optimum number of nodes required varies from problem to problem. For a homogeneous reservoir good results were obtained with 16 nodes on a circular boundary. However, it was necessary to represent a circular phase discontinuity by **32** nodes in order to obtain accurate results, although representing the outer boundary by only **16** nodes did not degrade the results.

A few runs were made for the composite rectangular reservoir to compare the run-times with different number of nodal points on the fluid interface. Since the computational effort in

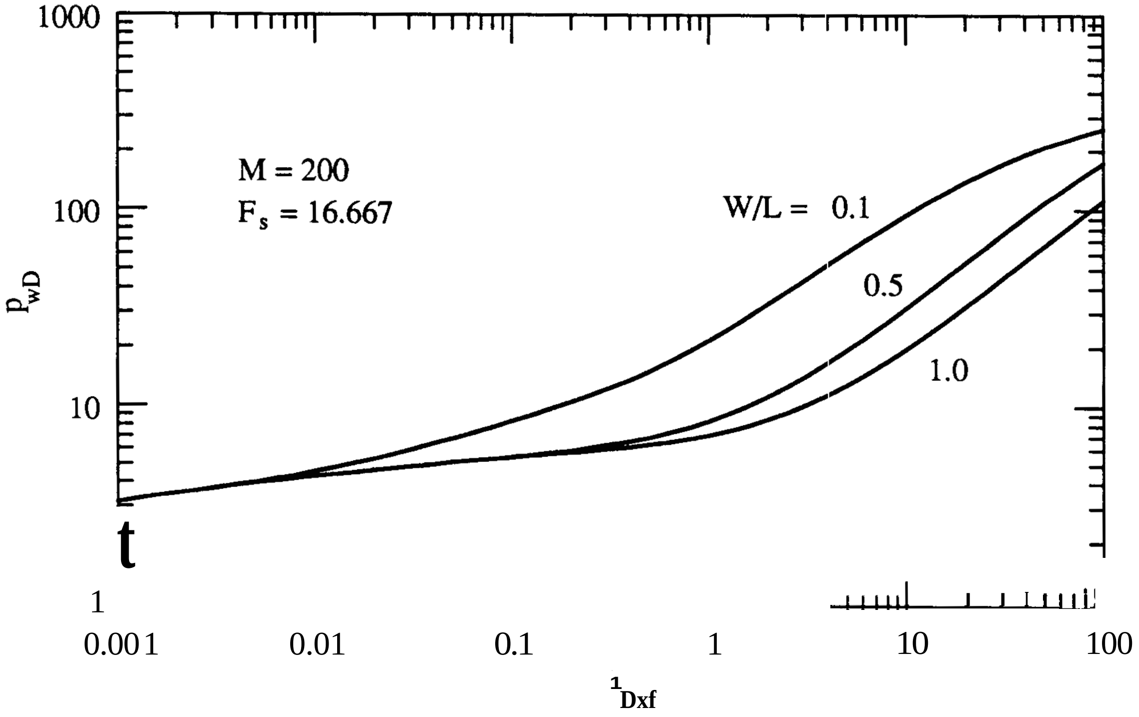


Fig. 5.14 Effect of W/L ratio on the pressure response of a rectangular composite reservoir

solving a full matrix is proportional to the cube of the number of equations, thus the effort for solving a composite problem increases tremendously as the number of nodal points increase. The size of the matrix increases more rapidly if the number of points on the interface are increased instead of increasing the points on the external boundary. For comparison purposes pressure transient solutions were calculated for 10 values of time. ‘The system considered was a rectangular composite reservoir with mobility and storativity ratios as 200 and 16.67 respectively. The width to length ratio of the fluid interface was one. The external boundary was discretized in eight nodal points for all the runs. Computation times were calculated on an Apollo 10000 for four, eight, sixteen and thirty-nodes on the interface. The coefficient matrix for the case with four nodes was 16 x 16. The CPU time needed was 19.5 seconds. For the case of eight interface nodes the matrix size was 24 x 24 and the run-time was 87.8 seconds. Computation times required for the cases with 16 and 32 interface nodes were respectively 1689 and 13146 seconds. These results indicate that the run-times increase very rapidly as the number of nodal points are increased.

Chapter 6

CONCLUSIONS

This study provides an analytical-numerical approach for computing pressure transient response in homogeneous and sectionally homogeneous reservoirs of arbitrary configuration. Multiwell production/injection schemes are easily incorporated. Three different categories of problems have been addressed. These are :

- Incompressible fluid flow (steady state)
- Transient flow in homogeneous media
- Unsteady flow in sectionally heterogeneous media

Arbitrary geometries are considered by a weak formulation. This gives rise to an integral equation for unknown boundary conditions. *An* auxiliary equation leads to the determination of solutions in the interior of the domain. The computations are carried out on the boundary of the domain, thus the name Boundary Element Method (BEM). Use of the Green's function in the solution scheme restricts the application of this method to linear problems. Based on the results of this study the following conclusions are made.

Boundary Element Method

- A reduction in dimensionality of the problem is achieved. This leads to smaller matrices for inversion compared to domain type methods.

- The global support of the basis (weighting) functions (Green's functions) leads to **full** matrices. Thus, for large problems the advantage of having small matrices **is** overwhelmed by the non-sparsity of the matrices. For such large problems, there may not **be** any computational savings for the boundary element method when compared to the domain methods.

- The analytical nature of the solution is preserved due to the use **of** the fundamental solution to the differential operator.

- The real space formulation to transient problems requires consideration of the convolution character of the resulting integrals.

- The Laplace space formulation of transient problems gives better results with a coarse nodal definition compared to **a** real space formulation and also allows for straightforward inclusion of rate decline, double porosity and wellbore storage effects.

- There may not **be** a gain in computing efficiency over other numerical methods, however accuracy is generally better.

- Comparison with analytical results show that the solutions are accurate for a variety of problems of increasing complexity.

- Consistency and convergence of the solution has been shown via a broad range of examples.

Pressure Transient Solutions

- Both pressure and pressure derivative solutions for arbitrarily shaped reservoirs with any linear boundary condition at a section of the reservoir were obtained by using the boundary element method.

- Multiple line sources (producers/injectors) at different rates are easily incorporated.

- Problems in reservoirs with internal subregions in the form of impermeable or recharge zone inclusions of any size shape and orientation can be solved.

o The pressure transient response in a two-zone composite reservoir has been solved with good results.

Computational Effort

o Most of the integrals over the boundary elemenls were performed analytically. Specifically, all the integrals for the steady state problems, 60% of the integrals for the real space transient problems and about 30% for Laplace space problem were done analytically.

o Singular integrals were evaluated analytically in terms of well known, easy to compute functions. Term by term integration of an infinite series or numerical evaluation of these integrals have been done previously in the literature. The analytical solution is exact and computationally efficient.

o Numerical integration was performed with highly accurate Gauss-Legendre quadrature. A six-point integration scheme and two panel subdivision gave good results, as it is necessary for accurate inversion of Laplace space solution by the Stehfest algorithm.

o The computational effort would increase tremendously for multizone problems or multiple internal boundaries. Thus, improvement in the area of solution of very large problems should be attempted.

6.1 Recommendations

The Boundary element method has shown promise in solving classical pressure transient problems and extended those to include a variety of new features. The following aspects should be addressed in future work.

- Extension of BEM to include weak nonlinearities by defining a perturbation Green's function.

- Investigation of increased flexibility in problem solving by including body force terms and discretizing the domain in finite element type meshes to compute integrals.

- Truncation of the stored convolution integrals. This will reduce storage requirements for the convolution BEM and improve computational efficiency.

- Extension of internal subregion and composite reservoir problems to more than two zones.

NOMENCLATURE

A	=	Area of the problem domain
A_i	=	Area of the region i
B_k	=	Set of undetermined parameters
$E_1(z)$	=	Exponential Integral
$\mathbf{F}\ (\lambda)$	=	Fourier transform with parameter λ
F	=	$\frac{a}{r}$, Size of the internal circular sub-region
F_{ij}	=	Elements of coefficient mamx corresponding to pressure
F_s	=	Storativity ratio, $\frac{\phi c_{t\,1}}{\phi c_{t\,2}}$
$G(x, y, t, \xi, \eta, \tau)$	=	Free Space Green's Function, (two-point function)
G_β	=	Right hand side vector in the matrix equation
$G(x, y, \xi, \zeta; s)$	=	Free Space Green's Function, Laplace Space
H	=	Coefficient matrix
$H(t_{DA} - \tau)$	=	Heaviside step function, 0 if $t_{DA} < \tau$, 1 if $t_{DA} \geq \tau$
K	=	Permeability tensor
$K_i(x)$	=	Modified Bessel function of the second kind of order i
$K_0(x)$	=	Modified Bessel function of the second kind of order zero
$K_1(x)$	=	Modified Bessel function of the second kind of order one
L_{ij}	=	Elements of coefficient mamx corresponding to **flux**
$L(p)$	=	Linear operator operating on p

$L(s)$	=	Laplace transform with parameter s
$L^*(G)$	=	Adjoint of operator L, operating on G
M	=	Mobility ratio, $\frac{(k/\mu)_1}{(k/\mu)_2}$
Q^*	=	Flow rate of a well, $[ML^{-3}T^{-1}]$
R	=	Distance to the discontinuity (phase front.), ft
b	=	Right hand side vector
c	=	Constant in the integral equation
c_t	=	Single phase fluid compressibility
$\exp(x)$	=	Exponential function of argument x
$erf\ (x)$	=	Error function or probability function, Eq. (B8)
i	=	Imaginary number = $\sqrt{-1}$
k_{xx}, k_{yy}	=	principal directions of permeability
n	=	Outward pointing normal to a line segment
n_w	=	Number of wells in the problem domain
p	=	Pressure, psia.
$\tilde{p}$	=	Reference pressure, for normalization
r_i	=	Radial distance from point i
s	=	Laplace transform parameter
t	=	Time
u	=	Solution vector
$\vec{v}$	=	Velocity vector
x, y	=	Variables in Cartesian coordinate system
x', y'	=	Stretched Cartesian coordinates, Eq. (3)

Greek Symbols

Ω	=	Domain of the solution region
Γ	=	Boundary of the solution region
α, β, γ	=	constants
$\delta(x - \xi)$	=	Dirac delta function, singular at $x = \xi$
δ_{ij}	=	Kronecker delta, 0 if $i \neq j$, and 1 if $i = j$
θ	=	Angle subtended between two adjacent elements
ξ, ζ	=	Local coordinate system based on the fictitious source point
η	=	Diffusivity $= \frac{k_{xx}}{\phi\mu c}$
ϕ	=	Porosity
ϕ_k	=	Set of linearly independent functions
ψ	=	Set of weighting functions
ε	=	Belongs to, or is an element of
ε	=	Error function in a residual scheme
μ	=	Viscosity, cp
ρ	=	Density, gm./cc.
ω	=	Fracture to total system storativity ratio
λ	=	Interporosity flow coefficient
$\nabla\bullet$	=	Divergence operator
∇	=	Gradient operator

Subscripts

D	=	Dimensionless
DA	=	Nondimensionalized with respect to area
eD	=	Dimensionless, based on external boundary area

wD	=	Dimensionless, based on well radius
1, 2, 3	=	Sections of the boundary
x, y	-	Partial derivative with respect to x or y
t	=	Partial derivative with respect to t
s	=	singular
ns	=	Nonsingular

REFERENCES

Abramowitz, M., Stegun, I.A., (ed.): *"Handbook of Mathematical Functions"*, U.S. Dept. of Commerce, National Bureau of Standards, Applied Mathematics Series - **55,** Nov. **1970.**

Ambastha, A.K.,: *"Pressure Transient Analysis for Composite Systems,"* Ph.D Thesis, Stanford University, Oct. **1988, 190.**

Ambastha, A.K. and Ramey, H.J. Jr.,: "Thermal Recovery Well Test Design and Interpretation," paper SPE **16746** presented at the **62**nd Annual Technical Conference and **Exhi**bition, Dallas, Sept. **27-30, 1987.**

Aral, M.M. and Tang, Y.,: "A New Boundary Element Formulation for Time-Dependent Confined and Unconfined Aquifer Problems," ***Wafer Resources Research, vol.*** *24,* ***no.*** 6, June **1988,** 831-42.

Aziz, K., and Settari, A.,: *"Petroleum Reservoir Simulation,"* Elsevier Applied Science, **1979.**

Bixel, H.C., and van Poollen, H.K.,: "Pressure Drawdown and Eluildup in the Presence of Radial Discontinuities," *Soc.* ***Pet. Eng. J.***, Sept. **1967,** 301-09.

Brebbia, C.,: "Weighted Residual Formulation of Approximate Methods", in ***Boundary Element Methods in Computer-Aided Engineering,*** NATO Advanced Study Institute on Boundary Elements in CAD, Series E : Applied Sciences - no. **84.,** Ed. Brebbia, C. A., Martinus Nijhoff Publishers, Dodrecht, **1984.**

Brebbia, C.A.,: "***The Boundary Element Method for Engineers,"***, Pentech Press, London, 1984b, 189.

Brebbia, C.A., and Walker, R.H.,: "***Boundary Element Techniques in Engineering,***" Newnes - Butterworths, London, 1980.

Britto, P.R., and Grader, A.S.,: "The Effects of Size, Shape, and Orientation of an Impermeable Region on Transient Pressure Testing," ***SPE Form. Eval.***, Sept. 1988, 595-606.

Carslaw, **H.S.,** and Jaegar, J.C.,: "***Conduction of Heat in Solids",*** Clarendon Press, Oxford, second **ed.,** 1959.

Cheng, A.H.-D.,: "Heterogeneities in Flows Through Porous Media by the Boundary Element Method," in "***Topics in Boundary Element Research, vol. 4*** **,"** Ed. Brebbia, C.A., Springer Verlag Publishers, 1984.

Connor, J.J., and Brebbia, C.A.,: "Boundary Element Methods," in "***Boundary Element Methods in Computer Aided Engineering,***" NATO Advanced Study Institute on Boundary Elements in CAD, Series E : Applied Sciences - no. 84., Ed. Brebbia, C.A., Martinus Nijhoff Publishers, Dodrecht, 1984.

Davis, E.G. and Hawkins, M.F.,: "Linear Fluid-Barrier Detection by Well Pressure Measurements," ***J. Pet. Tech.,*** Oct. 1963, 1077-79,

Earlougher, R.C.,Jr.,: "***Advances in Well Test Analysis,***" Monograph ***vol. 5,*** Society of Petroleum Engineers of AIME, Dallas, 1977, 186-91.

Earlougher, R.C.,Jr., Ramey, H.J.,Jr., Miller, F.G., and Mueller, T.D.,: "Pressure Distributions in Rectangular Reservoirs," ***J. Pet. Tech.,*** Feb. 1968, 199-208, *also* Trans. AIME 243.

Eggenschwiler, M., Ramey, H.J.Jr., Satman, A., and Cinco-Ley, H.,: "Interpretation of Injection Well Pressure Transient Data in Thermal Oil Recovery," paper presented at VI Joumadas Tecnicas de Petroleo Meeting, Maracaibo, Venezuela, Oct. 30 - Nov. 3, 1979.

Ellsworth, D.,: "A Model to Evaluate the Transient Hydraulic Response of Three-Dimensional Sparsely Fractured Rock Masses," ***Water Resources Research, vol. 22, no.*** 13, Dec. 1986, 1809-19.

Erdelyi, A., Magnus, W., Oberhettinger, F., and Tricomi, F.G.,: "***Tables of Integral Transforms,***" Bateman Manuscript Project, McGraw Hill Book Company, Inc., USA, 1954.

Gradshteyn, I.S., and Ryzhik, I.M.,: "***Tables of Integrals, Series and Products,***" Ed. Jeffrey, A., Academic Press, Inc., London, 1980.

Greenberg, M.D.,: "***Application of Green's Functions in Science and Engineering***", Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1971.

Hantush, M.S., and Jacob, C.E.,: "Flow to an Eccentric Well in a Leaky Circular Aquifer," ***J. Geoph. Res., vol. 65, no. 10,*** Oct. 1966, 3425-31.

Hughes, T.J.R.,: "***The Finite Element Method: Linear Static and Dynamic Finite Element Analysis***", Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1987.

Jialin, Z.,: "On the Mathematical Foundations of the Boundary Element Methods," ***in Boundary Elements,*** Proceedings of the International Conference; Beijing, China, Ed. Qinghua, D., Pergamon Press, Oct. 14-17, 1986,

Johnson, C.,: "***Numerical Solution of Partial Differential Equations by the Finite Element Method,***" Cambridge University Press, 1987.

Kikani, J., and Home, R.N.,: "A Convolution Boundary Element :Method for Unsteady State Groundwater Flow in Homogeneous Aquifers," Proc. XIII Geothermal Res. Eng. Workshop, Stanford U., Stanford, Jan. 19-21, 1988.

Kikani, J., and Home, R.N.,: "Pressure Transient Analysis of Arbitrary Shaped Reservoirs with the Boundary Element Method," SPE 18159 presented at the 63^rd^ Annual Technical Conference and Exhibition, Houston, Texas, Oct. 1-4, 1988b.

Kikani, J., and Home, R.N.,: "Application of Boundary Element Method to Reservoir Engineering Problems," [To be published] in ***The Journal of Petroleum Science and Engineering,*** Elsevier Science, 1989.

Kikani, J., and Home, R.N.,: "Modeling Pressure Transient Behavior of Sectionally Heterogeneous Reservoirs by the Boundary Element Method," SPE 19778 to **be** presented at the **64**th Annual Technical Conference and Exhibition, San Antonio, Texas, Oct. 8-11, 1989b.

Lafe, O.E., and Cheng, A. H-D.,: "A Perturbation Boundary Element Code for Steady State Groundwater Flow in Heterogeneous Aquifers," ***Water Resources Research,*** *vol.* ***23, no. 6,*** June 1987, 1079-1084.

Lafe, O.E., Liggett, J.A., and Liu, P.L-F.,: "BIEM Solutions to Combinations of Leaky, Layered, Confined, Unconfined, Nonisotropic Aquifers," ***Water Resources Research,*** *vol.* ***17, no.*** *5*, Oct. 1981, 1431-44.

Lafe, O.E., Montes, J.S., Cheng, A.H-D., Liggett, J.A., and Liu, P.L-F.,: "Singularities in Darcy Flow Through Porous Media," **J. *Hyd.*** *Div.,* ***Proc.*** *of* ***the ASCE,*** *vol.* ***106,*** *no.* ***HY6,*** June 1980, 977-97.

Larsen, L.: "A Simple Approach to Pressure Distribution in Geometric Shapes," *Soc.* ***Pet.*** Eng. **J.,** Feb. 1985, 113-120.

Lennon, G.P., Liu, P.L-F., and Liggett, J.A.,: "Boundary Integral Equation Solution to Axisymmetric Potential Flows: 1. Basic Formulation," ***WaferResources Research,*** *vol.* ***15,*** *no.* ***5,*** Oct. 1979, 1102-106.

Lennon, G.P., Liu, P.L-F., and Liggett, J.A.,: "Boundary Integral Equation Solution to Axisymmetric Potential Flows: 2. Recharge and Well Problems in Porous Media," ***Water Resources Research,*** *vol.* ***15,*** *no. 5*, Oct. 1979, 1107-115.

Lennon, G.P., Liu, P.L-F., and Liggett, J.A.,: "Boundary Integral Solutions to Three-Dimensional Unconfined Darcy's Flow," ***Water Resources Research,*** *vol.* ***16,*** *no.* ***4,*** Aug. 1980, 651-58.

Liggett, J.A.,: "Location of Free Surface in Porous Media," **J.** *of* ***Hyd.*** *Div.,* ***Proc. of the ASCE,*** *vol. 103, no.* HY4, April, 1977.

Liggett, J.A., and Liu, P.L-F.,: "Unsteady Flow in Confined Aquifen: **A** Comparison of Two

Boundary Integral Methods," ***Water Resources Research, vol. 15, no. 4*** , Aug. 1979, 861-66.

Liggett, J.A., and Liu, P.L-F.,: *"**The Boundary Integral Equation Method for Porous Media Flow,**"* George Allen and Unwin, London, 1983.

Liu, P.L-F., and Liggett, J.A.,: "Boundary Integral Solutions to Groundwater Problems," Proc. First Int. Conf. on App. Num. Modeling, U of Southampton, England, July 11-15, 1977, 559-69.

Liu, P.L-F., **and** Liggett, J.A.,: ***"An*** Efficient Numerical Method of Two-Dimensional Steady Groundwater Problems," ***Water Resources Research,*** *vol.* ***14, no. 3,*** June 1978, 385-390.

Liu, P.L-F, and Liggett, J.A.,: "Boundary Solutions to Two Probleims in Porous Media," ***J. of Hyd. Div., Proc. of the ASCE,*** *vol.* ***105, no.*** HY3, March 1979, 171-183.

Liu, P.L-F., and Liggett, J.A.,: "Numerical Stability and Accuracy of Implicit Integration of Free Surface Groundwater Equations," ***Water Resources Research, vol. 16, no. 5,*** Oct. 1980, 897-900.

Liu, P. L-F, Cheng, A.H-D, Liggett, J.A., and Lee, J.H.,: "Boundary Integral Equation Solutions to Moving Interface Between Two Fluids in Porous Media," ***Water Resources Research, vol. 17, no. 5,*** Oct. 1981, 1445-52.

Masukawa, J.,: *"**The Application of The Boundary Integral Method to Immiscible Displacement Problems**"* M.S. Report, Stanford University, Stanford, June 1985.

Masukawa, J., and Home, R.N.,: "Application of the Boundary Integral Method to Immiscible Displacement Problems," ***SPE Res. Eng.,*** Aug. 1988, 1069-77.

Morel-Seytoux H.J.,: "Analytical-Numerical Method in Waterflooding Predictions," ***Soc. Pet. Eng. J.,*** Sept. 1965.

Morse, P.M., and Feshbach, H.,: *"**Methods of Theoretical Physics**",* McGraw Hill, New York, 1953.

Nabor, G.W., and Barham, R.H.,: "Linear Aquifer Behavior," ***J. Pet. Tech.,*** May 1964, 561-63.

Numbere, D.T.,: "*A General Streamline Modeling Technique for Homogeneous and Heterogeneous Porous Media with Applications to Steamflood Prediction,"* Ph.D Dissertation, University of Oklahoma, Norman, 1982.

Numbere, D.T., and Tiab, D.: "An Improved Streamline-Generating Technique That Uses the Boundary (Integral) Element Method," ***SPE*** Res. ***Eng.,*** Aug. 1988, 1061-68.

Odeh, A.S.,: "Flow Test Analysis for a Well with Radial Discontinuity," ***J. Pet. Tech.,*** Feb. 1969, 207-10.

Pina, H.L.G,: "Time Dependent Potential Problems," in ***Boundary Element Methods in Computer Aided Engineering,*** NATO Advanced Study Institute on Boundary Elements in CAD, Series E : no. **84,** Ed. Brebbia, C.A., Martinus Nijhoff Publishers, Dodrecht, 1984.

Pina, H.L.G.,: "Numerical Integration and Other Computational Techniques," ***Boundary Element Techniques in Computer-Aided Engineering,*** Series E : Applied Sciences - no. 84, Ed. Brebbia C.A., Martinus Nijhoff Publishers, Dodrecht, 1984b.

Prasad, R.K.,: "Pressure Transient Analysis in the Presence of Two Intersecting Boundaries," ***J. Pet. Tech.*** (Jan. 1975) 89-96.

Ramey, H.J.Jr.,: "Approximate Solutions for Unsteady State Liquid Flow in Composite Reservoir," J. Cdn. Pet. Tech., Jan.-Mar. 1970, 32-37.

Ramey, H.J.Jr., Kumar, A., and Gulati, M.S.,: **"Gas** ***Well Test Analysis Under Water Drive Conditions,"*** AGA Press, Arlington, 1973.

Rizzo, F.J., and Shippy, D.J.,: "A Method of Solution for Certain Problems of Transient Heat Conduction," ***AIAA Journal, vol. 8, no. 11,*** Nov. 1970, 2004-2009.

Roache, P.J.,: "***Computational Fluid Dynamics,"*** Hermosa Publishers, Albuquerque, NM, 1972.

Russell, T.F., and Wheeler, M.F.,: "Finite Element and Finite Difference Methods for Continuous Flows in Porous Media," ***The Mathematics of Reservoir Simulation,*** Frontiers in Applied Mathematics, Ed. Ewing, R.E., SIAM, Philadelphia, 1983.

Sageev, A.,: *"Pressure Transient Analysis of Reservoirs with Linear or Internal Circular Boundaries,"* Ph.D Thesis, Stanford University, June 1983, 228.

Shapiro, A., and Andersson J.,: "Steady State Fluid Response in Fractured Rock : A Boundary Element Solution for a Coupled, Discrete Fracture Continuum Model," *Water Resources Research,* **vol.** *19,* ***no. 4,*** Aug. 1983, 959-969.

Shaw, R.P.,: ***"An*** Integral Equation Approach to Diffusion," *Znt. J. Heat Mass Transfer,* ***vol.*** *17,* 1974, 693-699.

Spiegel, M.R.,: *"Theory and Problems of Complex Variables,"* Schaum's outline *series* in mathematics, McGraw Hill Book Co., 1964.

Stehfest, H.,: *"Algorithm* 368, Numerical Inversion of Laplace Transforms [D5]," *Communications of ACM,* **vol.** *13, No. 1,* Jan. 1970, 47-49.

Streltsova, T.D., and McKinley, R.M.,: "Effect of Flow Time Duration on Buildup Pattern for Reservoirs with Heterogeneous Properties," *Soc. Pet. Eng. J.,* June 1984, 294-306.

Stroud, A.H., and Secrest, D.,: *"Gaussian Quadrature Formulas,"* Prentice *Hall,* Englewood Cliffs, New Jersey, 1966.

Taigbenu, A.E.,: *"A* New Boundary *Element Formulation Applied to Unsteady Aquifer Problem,"* Ph.D Thesis, Comell University, Ithaca, 1985.

Taigbenu, A.E., and Liggett, J.A.,: "Boundary Element Calculations;of Diffusion Equation," *J. Eng. Mech.,* **vol.** *111, no.* 3, March 1985, 311-28.

Taigbenu, A.E., and Liggett, J.A.,: ***"An*** Integral Formulation Applied to the Diffusion and Boussinesq Equations," *Int.* **J.** *for Num. Methods in Eng.,* ***vol.*** *23,* 1986, 1057-1079.

Teng, E.,: *"Transient Pressure Analysis in Composite Reservoirs with Rectangular Discontinuities,"* M.S. Report, Stanford University, Sept. 1984, 153.

Tiab, D. and Cnchlow, H.B.,: "Pressure Analysis of Multiple-Sealing Fault Systems and Bounded Reservoirs by Type Curve Matching," *Soc. Pet. Eng. J.,* Dec. 1979, 379-92.

Tiab, D. and Kumar, A.,: "Detection and Location of Two Parallel Sealing Faults Around a Well," **J.** *Pet. Tech.,* Oct. 1980, 1701-08.

Van Everdingen, A.F., and Hurst, W.,: "The Application of the Laplace Transformation to Flow Problems in Reservoirs," ***Trans. AZME*** 186, 1949.

Van Kruysdijk, C.P.J.W. and Dullaert, G.M.,: "A Boundary Element Solution to the Transient Pressure Response of Multiply Fractured Horizontal Wells," Preprint, The Joint IMA/SPE European Conference on the Mathematics of Oil Recovery, Robinson College, Cambridge University, July 25-27, 1989.

Van Poollen, H.K.,: "Transient Tests Find Fire Front in an In-Situ Combustion Project," ***Oil and Gas J.,*** Feb. 1, 1965, 78-80.

Wong, W.D., Mofhersele, C.D., Hamngton, A.G., and Cinco-Ley, H.,: "Pressure Transient Analysis of Finite Linear Reservoirs Using Derivative and Conventional Techniques : Field Examples," Paper SPE 15421 presented at the 61st Annual Mtg. in New Orleans, LA, Oct. ***5-8,*** 1986.

Wrobel, L.C., and Brebbia, C.,: "Time Dependent Potential Problems", ***Progress in Boundary Element Methods, vol.*** 1, Ed. Brebbia, C.A., 1981.

Wrobel, L.C., Brebbia, C.A., and Nardini, D.,: "The Dual Reciprocity Boundary Element Formulation for Transient Heat Conduction," Proc. of the Sixth Int. Conf. on Finite Elements in Water Resources, Lisbon, Portugal, 1986 801-811.

Zauderer, E.,: ***Partial Differential Equations in Applied Matheinatics,*** Wiley-Interscience, 1983.

APPENDIX A

Free Space Green's Function

A.l Diffusion Operator

The adjoint problem for the diffusion operator is given by

$$\nabla^2 G + \frac{\partial G}{\partial t_{DA}} = -\delta(x_D - \xi)\,\delta(y_D - \zeta)\,\delta(t_{DA} - \tau) \qquad \text{(A.1)}$$

Only one space dimension is considered for simplicity. Once the Green's function is obtained, extension to the second dimension is straightforward. Thus in one dimension

$$\frac{\partial^2 G}{\partial x_D^2} + \frac{\partial G}{\partial t_{DA}} = -\delta(x_D - \xi)\,\delta(t_{DA} - \tau) \qquad \text{(A.2)}$$

The infinite Fourier transform is defined **as**

$$\mathbf{F}\,(\lambda, \xi, t_{DA}, \tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\lambda x_D} G\,(x_D, \xi, t_{DA}, \tau)\,dx_D \qquad \text{(A.3)}$$

where $i = \sqrt{-1}$. Multiply Eq. (A.2) by $\frac{1}{\sqrt{2\pi}}\, e^{i\lambda x_D}$ and integrating over the **infinite** domain we obtain on using **Eq. (A.3)**

$$\frac{\partial \mathbf{F}}{\partial t_{DA}} - \lambda^2 \mathbf{F} = -\frac{1}{\sqrt{2\pi}}\, e^{i\lambda\xi}\,\delta(t_{DA} - \tau) \qquad \text{(A.4)}$$

where

$$\mathbf{F}\,\{\delta(x_D - \xi)\} = \frac{1}{\sqrt{2\pi}}\, e^{i\lambda\xi} \qquad \text{(A.5)}$$

Rewriting Eq. (A.4) in a shorthand notation with subscripts denoting partial derivatives, we get

$$\mathbf{F}_t - \lambda^2 \mathbf{F} = -\frac{1}{\sqrt{2\pi}} e^{i\lambda\xi}\, \delta(t - \tau) \tag{A.6}$$

where the subscripts from the dimensionless variables have been omitted for convenience. Solving the homogeneous equation first

$$\mathbf{F}_t - \lambda^2 \mathbf{F} = 0 \tag{A.7}$$

$$\Longrightarrow \qquad \mathbf{F} = A\ e^{\lambda^2 t} \qquad \tau \leq t \leq T \tag{A.8}$$

where T is the total time to which the solution is required. Using the homogeneous initial condition, we obtain the solution to Eq. (A.6)

$$\mathbf{F}(\lambda, \xi, t, \tau) = \frac{1}{\sqrt{2\pi}} e^{i\lambda\xi}\, e^{\lambda^2(t-\tau)}\, H(t - \tau) \tag{A.9}$$

where $H(t - \tau)$ is the Heaviside step function. The Fourier inversion formula is given by

$$G(x_D, \xi, t_{DA}, \tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\lambda x_D}\, \mathbf{F}(\lambda, \xi, t_{DA}, \tau)\, d\lambda \tag{A. 10}$$

Inverting Eq. (A.9) according to Eq. (A.lO)

$$G(x, \xi, t_{DA}, \tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp\left[-i\lambda(x - \xi) - \lambda^2(\tau - t)\right] H(t - \tau)\, d\lambda \tag{A. 11}$$

Adding and subtracting $\left\{\frac{i(x - \xi)}{2\sqrt{t - \tau}}\right\}^2$ in the exponent

$$G(x, \xi, t, \tau) = \frac{1}{2\pi} H(t - \tau) \int_{-\infty}^{\infty} \exp\left[-i\lambda(x - \xi) - \lambda^2(t - \tau) + \left\{\frac{i(x - \xi)}{2\sqrt{t - \tau}}\right\}^2 - \left\{\frac{i(x - \xi)}{2\sqrt{t - \tau}}\right\}^2\right] d\lambda \tag{A. 12}$$

and completing the square results in

$$G(x, \xi, t, \tau) = \frac{1}{2n:} H(t - \tau) \exp\left[\frac{-(x - \xi)^2}{4(t - 2)}\right] \left[\int_{-\infty}^{\infty} \exp\left\{-\left[\lambda\sqrt{t - \tau} + \frac{i(x - \xi)}{2\sqrt{t - \tau}}\right]^2\right\} d\lambda\right] \tag{A.13}$$

Making a substitution

$$\frac{x - \xi}{2\sqrt{t - \tau}} = c \tag{A.14}$$

Eq. **(A.13)** becomes

$$G(x, \xi, t, \tau) = \frac{1}{2\pi} H(t - \tau) e^{-c^2} \left[\int_{-\infty}^{\infty} e^{-\left[\lambda\sqrt{t - \tau} + ic\right]^2} d\lambda\right] \tag{A.15}$$

Making another substitution

$$ic + \lambda\sqrt{t - \tau} = z \tag{A.16}$$

Eq. **(A.15)** becomes

$$G(x, \xi, t, \tau) = \frac{1}{2n:} H(t - \tau) \frac{e^{-c^2}}{\sqrt{t - \tau}} \left[\int_{-\infty + ic}^{\infty + ic} e^{-z^2} dz\right] \tag{A.17}$$

The integrand of **Eq. (A.17)** is an analytic function as it satisfies the Cauchy-Riemann equations. Thus, the Cauchy residue theorem which defines the integral of an analytic function around **a** closed contour as

$$\int_c f(z)\, dz = 2\pi i \sum \left(\textit{Residues due to singularities}\right) \tag{A.18}$$

can be applied to Eq. **(A.17).** The closed contour is defined **as** a rectangular region on the complex surface shown in Fig. (A.1). There are no singularities **of** the function e^{-z^2} within the contour. Thus, by the Cauchy residue theorem

$$\int_\Gamma e^{-z^2} dz = 0 \tag{A.19}$$

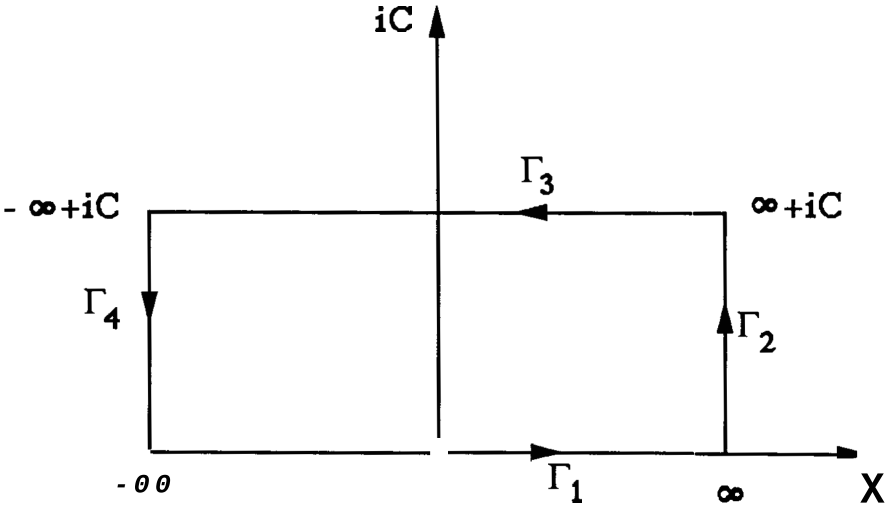


Fig. **A.l** Schematic of integration contour in the complex plane

Breaking the left hand integral into sections

$$\left\{\int_{\Gamma_1}+\int_{\Gamma_2}+\int_{\Gamma_3}+\int_{\Gamma_4}\right\}(e^{-z^2}\,dz) = 0 \tag{A.20}$$

where z is x on the real axis and y on the imaginary axis. Rewriting Eq. **(A.20)** after transposing the terms

$$\left\{-\int_{\Gamma_3} = \int_{-\infty+ic}^{\infty+ic} = \int_{\Gamma_1}+\int_{\Gamma_2}+\int_{\Gamma_4}\right\}(e^{-z^2}\,dz) \tag{A.21}$$

The sense of the integrals over the lines Γ_2 and Γ_4 are opposite and equal, and therefore cancel. Eq. **(A.20)** becomes

$$\int_{-\infty+ic}^{\infty+ic} e^{-z^2}\,dz = \int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi} \tag{A.22}$$

Using the results of Eq. **(A.22)** in Eq. **(A.17)** and restoring the value of c from Eq. **(A.14)**

$$G(x,\xi,t,\tau) = \frac{1}{4\pi(t-\tau)^{1/2}}\,H(t-\tau)\exp\left[-\frac{(x-\xi)^2}{4(t-\tau)}\right] \tag{A.23}$$

Extending this to two-dimensional space

$$G(x,y,t,\xi,\zeta,\tau) = \frac{1}{4\pi(t-\tau)}\,H(t-\tau)\exp\left[-\frac{(x-\xi)^2+(y-\zeta)^2}{4(t-\tau)}\right] \tag{A.24}$$

Eq. **(A.24)** is the free space Green's function. It can be derived in a variety of ways. ***Zauderer (1983)*** and ***Morse and Feschbach (1953)*** give an alternate derivation.

A.2 Modified Helmholtz Operator

The adjoint problem in Laplace space is given by

$$\nabla^2\bar{G} - s\bar{G} = -\delta(x-\xi)\,\delta(y-\zeta) \tag{A.25}$$

The operator in Eq. **(A.25)** is known as the modified Helmholtz operator. In two dimensions, Eq. **(A.25)** can be written as

$$\frac{\partial^2 \overline{G}}{\partial x_D^2} + \frac{\partial^2 \overline{G}}{\partial y_D^2} - s\overline{G} = -\delta(x_D - \xi)\,\delta(y_D - \zeta) \qquad \textbf{(A.26)}$$

The Fourier transform in the x-direction is defined slightly differently compared to Eq. **(A.3).** It is given by

$$\overline{\mathbf{F}}\,(\lambda, \xi, y, \zeta; s) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\lambda x} G(x, \xi, y, \zeta; s)\, dx \qquad \textbf{(A.27)}$$

Taking the Fourier transform first in the x and then in the y direction with the y direction transform being denoted by $\hat{\overline{\mathbf{F}}}$, we obtain

$$\frac{\partial^2 \overline{\mathbf{F}}}{\partial y^2} - (\lambda^2 + s)\overline{\mathbf{F}} = -\frac{1}{\sqrt{2\pi}} e^{i\lambda\xi}\,\delta(y - \zeta) \qquad \textbf{(A.28)}$$

Transforming Eq. (A.28) again to Fourier space in the y dimension, and denoting the transform by $\hat{\overline{\mathbf{F}}}$ and the second Fourier space parameter as ω we obtain

$$(\lambda^2 + \omega^2 + s)\,\hat{\overline{\mathbf{F}}} = \frac{1}{2\pi} e^{-i(\lambda\xi + \omega\zeta)} \qquad \textbf{(A.29)}$$

Solving Eq. **(A.29)** we obtain

$$\hat{\overline{\mathbf{F}}} = \frac{1}{211:} \frac{e^{-i\lambda\xi}\, e^{-i\omega\zeta}}{\lambda^2 + \omega^2 + s} \qquad \textbf{(A.30)}$$

Using the shifting property of Fourier transforms [*Erdelyi* et. ***al* (1954)** *vol.* I pg. **117** #7] gives

$$\overline{\mathbf{F}} = \hat{\overline{\mathbf{F}}}^{-1}\left\{\frac{1}{2\pi} \frac{e^{-i\lambda\xi}}{\lambda^2 + \omega^2 + s}\right\}\Bigg|_{y = \zeta - y} \qquad \textbf{(A.31)}$$

Using the inversion in ***Erdelyi*** et. *al.* ***(1954)*** ***vol.*** I. pg. 118 #5 we obtain

$$\overline{\mathbf{F}} = \frac{1}{2\pi} \frac{\pi}{\sqrt{\lambda^2 + s}} e^{-|y - \zeta|\sqrt{\lambda^2 + s}} \qquad \textbf{(A.32)}$$

Using the shifting property of Fourier transforms again on x

$$\bar{G} \;=\; \bar{\mathbf{F}}^{-1}\left\{ \frac{e^{-|y-\zeta|\sqrt{\lambda^2+s}}}{2\sqrt{2\pi}\,\sqrt{\lambda^2+s}} \right\}\Bigg|_{x\,=\,\xi\,-\,x} \qquad \textbf{(A.33)}$$

The function to be inverted in **Eq. (A.33) is** an even function, thus the exponential transform can **be** reduced to **a** cosine Fourier transform

$$\int_{-\infty}^{\infty} e^{i\lambda x}\, f_{even}(x)\, dx \;=\; 2\int_{0}^{\infty} \cos(\lambda x)\, f_{even}(x)\, dx \qquad \textbf{(A.34)}$$

Using ***Erdelyi*** *et. al.* ***(1954)*** *vol.* I pg. 17 #27, the free space Green's function **is** given **by**

$$\bar{G}(x,\xi,y,\zeta;s) \;=\; \frac{1}{2\pi}K_0\left(\sqrt{(x-\xi)^2+(y-\zeta)^2}\,\sqrt{s}\right. \qquad \textbf{(A.35)}$$

APPENDIX B

Boundary Integral Evaluations

B.1 Steady State Flow

The integrals in Eq. **(3.6)** can be performed analytically. **Each** of the integrals are given here. Referring to Fig. **(3.1),** r_{D_i} **is** given by **Eq.** (3.7). The normal derivative is given **by** Eq. **(3.9).** This information is used in simplifying the integrals. These integrals are shown in an indefinite form, subsequently the limits of integration must be used for the definite integrals.

$$\int \xi \ln r_{D_i}\, d\xi = \frac{1}{2}\int \xi \ln(\xi^2 + \zeta_i^2)\, d\xi = \frac{1}{4}(\xi^2 + \zeta_i^2)\left[\ln(\xi^2 + \zeta_i^2) - 1\right] \qquad \text{(B.1)}$$

$$\int \ln r_{D_i}\, d\xi = \frac{1}{2}\int \ln(\xi^2 + \zeta_i^2)\, d\xi \qquad \text{(B.2)}$$

$$= \frac{1}{2}\left[\xi \ln(\xi^2 + \zeta_i^2) - 2\xi + 2\zeta_i \tan^{-1}\frac{\xi}{\zeta_i}\right] \qquad \text{(B.3)}$$

$$\int \xi \frac{1}{r_{D_i}}\frac{\partial r_{D_i}}{\partial n}\, d\xi = \int \xi \frac{\zeta_i}{(\xi^2 + \zeta_i^2)d\xi} = \frac{\zeta_i}{2}\ln(\xi^2 + \zeta_i^2) \qquad \text{(B.4)}$$

where **Eq. (B.4)** was obtained **by** using **Eqs. (3.7)** and **(3.8)**

$$\int \frac{1}{r_{D_i}}\frac{\partial r_{D_i}}{\partial n}\, d\xi = \int \frac{\zeta_i}{(\xi^2 + \zeta_i^2)d\xi} = \tan^{-1}\frac{\xi}{\zeta_i} \qquad \text{(B.5)}$$

B.2 Velocity Calculations for Steady Flow

Equation (3.12) is discretized using linear interpolation functions for both pressure and fluxes. Figure (B.1) shows a sketch of integration over an element from an interior point. The discretized equation for the velocity field in the interior in terms of boundary values is

$$v_x = -\frac{k}{\mu}\frac{1}{2\pi}\sum_j\left[\int_{\xi_j}^{\xi_{j+1}} \frac{1}{r_D}\frac{\partial r_D}{\partial x}\,(\frac{\partial p_D}{\partial n}{}_j) - \frac{p_{D_j}}{r_D^2}\frac{\partial \zeta}{\partial x} + 2\,\zeta\,\frac{p_{D_j}}{r_D^3}\frac{\partial r_D}{\partial x}\right] d\xi \tag{B.6}$$

where from Figure (2.2)

$$\frac{\partial r_D}{\partial n} = \frac{\zeta}{r_D} \quad , \quad \frac{\partial \zeta}{\partial x} = \sin\psi \tag{B.7}$$

and

$$\frac{\partial r_D}{\partial x} = \frac{\xi\,\cos\psi - \zeta\,\sin\psi}{r_D} \tag{B.8}$$

Using Eqs. (B.7) and (B.8) in Eq. (B.6) gives

$$v_x = \frac{k}{\mu}\frac{1}{2\pi}\sum_j\int_{\xi_j}^{\xi_{j+1}}\left[\frac{\xi\,\cos\psi - \zeta\,\sin\psi}{r_D^2}(\frac{\partial p_D}{\partial n})_j + \frac{\sin\psi}{r_D^2}p_{D_j}\right.$$

$$\left. + \frac{2\,\zeta\,(\,\xi\,\cos\psi - \zeta\,\sin\psi\,)}{r_D^4}p_{D_j}\right] d\xi \tag{B.9}$$

Similarly, v_y is

$$v_y = -\frac{k}{\mu}\frac{1}{2\pi}\sum_j\int_{\xi_j}^{\xi_{j+1}}\left[-\,\frac{\cos\psi + \xi\,\sin\psi}{r_D^2}(\frac{\partial p_D}{\partial n})_j + \frac{\cos\psi}{r_D}\right.$$

$$\left. - \frac{2\,\zeta\,(\,\zeta\,\cos\psi - \xi\,\sin\psi)}{r_D^4}p_{D_j}\right] \tag{B.10}$$

If ζ and $\boldsymbol{n}$ are in the same direction as shown in Fig. (B.l) then ζ is positive. If ζ and $\boldsymbol{n}$ are in opposite directions, as may **be** the case when the vector r from point P to Q passes outside the region, the sign on ζ is negative.

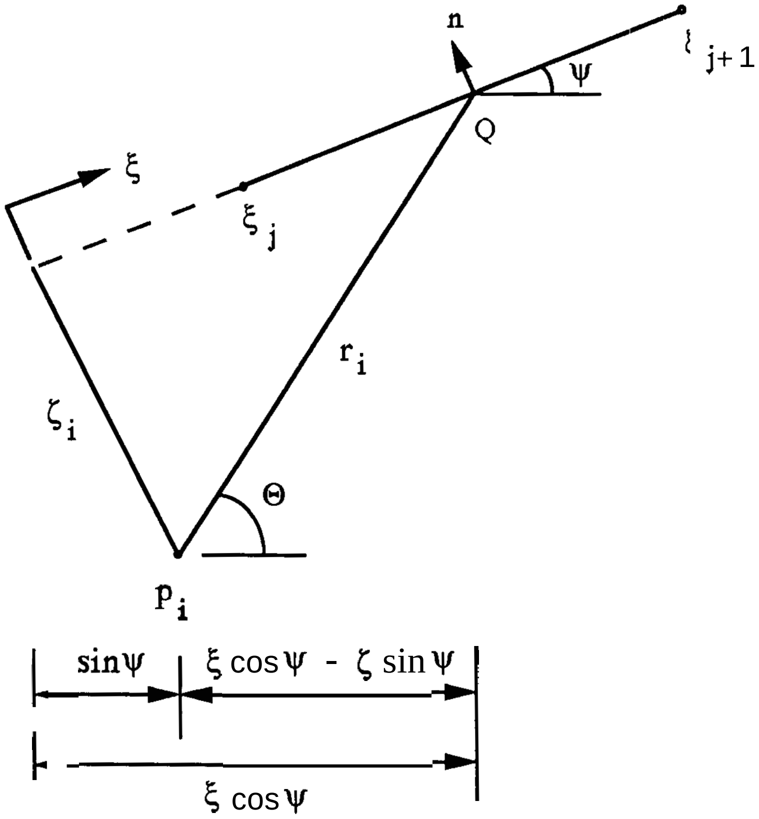


Fig. B. 1 Geometric scheme for integration from an interior point

The integrals needed to evaluate **Eqs. (B.9)and (B.lO)** may be done analytically and are

$$\int \frac{d\xi}{r_D^4} = \int \frac{d\xi}{(\xi^2+\zeta^2)^2} = \frac{\xi}{2\,\zeta^2\,(\xi^2+\zeta^2)} + \frac{1}{2\,\zeta^3}\tan^{-1}\frac{\xi}{\zeta} \tag{B.11}$$

$$\int \frac{\xi}{r_D^4}\,d\xi = \int d\xi \frac{\xi}{(\xi^2+\zeta^2)^2} = -\frac{1}{2(\xi^2+\zeta^2)} \tag{B.12}$$

$$\int d\xi\,\frac{\xi^2}{\xi^2+\zeta^2} = \xi - \zeta\tan^{-1}\frac{\xi}{\zeta} \tag{B.13}$$

$$\int d\xi\,\frac{\xi^2}{(\xi^2+\zeta^2)^2} = -\frac{\xi}{2(\xi^2+\zeta^2)} + \frac{1}{2\,\zeta}\tan^{-1}\frac{\xi}{\zeta} \tag{B.14}$$

B.3 Convolution BEM

The following boundary integrals which arise in the integral formulation with linear elements may be performed analytically

$$I_1 = \int_{t_0}^{t_1} \frac{d\tau}{2\,(t_1-\tau)^2}\exp\left[-\frac{r^2}{4(t_1-\tau)}\right. \tag{B.15}$$

Making a substitution

$$y = \frac{r^2}{4(t_1-\tau)} \quad \Longrightarrow d\tau = \frac{r^2}{4y^2}\,dy \tag{B.16}$$

Eq. (B.15) becomes

$$\Longrightarrow \quad I_1 = \int_{y_0}^{\infty} \frac{r^2}{4y^2}\left[\frac{4y^2}{r^2}e^{-y}\right]dy = \int_{y_0}^{\infty}\frac{4}{r^2}\,e^{-y}\,dy \tag{B.17}$$

On integrating **Eq. (B.17)** we obtain

$$I, = \frac{4}{r^2}\,e^{-y_0} \tag{B.18}$$

where $\quad y_0 = \dfrac{r^2}{4(t_1-t_0)}.$

Another boundary integral obtained is integrated exactly by making a couple of substitutions. This is shown in the following derivation.

$$I_2 = \int_{\xi_j}^{\xi_{j+1}} \frac{\xi}{(\xi^2 + \eta_i^2)} \exp\left[- \frac{\xi^2}{4(t - \tau)} \right] d\xi \tag{B.19}$$

$$\textit{Let,} \quad \frac{\xi^2}{4(\mathrm{t} - \tau)} = y \Longrightarrow \xi \, d\xi = 2(t - \tau)\, dy \tag{B.20}$$

Substituting Eq. (B.20) into Eq. (B.19) and simplifying

$$I_2 = \frac{1}{2} \int_{\frac{\xi_j^2}{4(t - \tau)}}^{\frac{\xi_{j+1}^2}{4(t - \tau)}} \frac{e^{-y}}{\left[y + \eta_i^2 / 4(t - \tau) \right]} dy \tag{B.21}$$

To simplify Eq. (B.21) further, assume

$$a = \frac{\eta_i^2}{4(t - 2)} \tag{B.22}$$

Eq. (B.21) reduces to

$$I_2 = \frac{1}{2} \int_{\frac{\xi_i^2}{4(t - 7)}}^{\frac{\xi_{j+1}^2}{4(t - \tau)}} \frac{e^{-y}}{(y + a)} dy \tag{B.23}$$

To reduce Eq. (B.23) into a familiar form, let

$$y + a = z \tag{B.24}$$

Then, Eq. (B.23) becomes

$$I_2 = \frac{1}{2} e^{a} \int_{\frac{\xi_j^2}{4(t - 7)}}^{\frac{\xi_{j+1}^2}{4(t - \tau)}} \frac{\mathrm{e}^{-z}}{z} dz \tag{B.25}$$

Eq. (B.25) is the familiar Exponential Integral form. Integrating Eq. (B.25) and resubstituting Eqs. (B.20), (B.22) and (B.24)

$$I_2 = \frac{1}{2} \exp\left[\frac{\eta_i^2}{4(t-\tau)}\right] \left[E_1\left[\frac{\xi_j^2+\eta_i^2}{4(t-\tau)}\right] - E_1\left[\frac{\xi_{j+1}^2+\eta_i^2}{4(t-\tau)}\right]\right] \qquad \text{(B.26)}$$

Another integral is of the form

$$I_3 = \int_{\xi_j}^{\xi_{j+1}} \xi\, E_1\left[\frac{\xi^2+\eta_i^2}{4(t-\tau)}\right] d\,\xi \qquad \text{(B.27)}$$

Substituting

$$\frac{\xi^2+\eta_i^2}{4(t-\tau)} = z \qquad \text{(B.28)}$$

and integrating by parts

$$I_3 = 2\,(t-\tau)\left[\; z\; E_1(z) + (1-e^{-z})\right]\Bigg|_{\frac{\xi_j^2+\eta_i^2}{4(t-\tau)}}^{\frac{\xi_{j+1}^2+\eta_i^2}{4(t-\tau)}} \qquad \text{(B.29)}$$

Incorporating the limits of integration we obtain

$$I_3 = \frac{1}{2}\left\{(\xi_{j+1}^2+\eta_i^2)\,E_1\left[\frac{\xi_{j+1}^2+\eta_i^2}{4(t-\tau)}\right] - (\xi_j^2+\eta_i^2)\,E_1\left[\frac{\xi_j^2+\eta_i^2}{4(t-\tau)}\right]\right\}$$

$$+\,2(t-\tau)\exp\left[-\frac{\eta_i^2}{4(t-\tau)}\right]\left\{\exp\left[-\frac{\xi_j^2}{4(t-\tau)}\right] - \exp\left[-\frac{\xi_{j+1}^2}{4(t-\tau)}\right]\right\} \qquad \text{(B.30)}$$

B.4 Laplace Space BEM

Only one of the boundary integrals could be evaluated analytically. The integrals which **are** singular at one of their limits are shown in Appendix C.

$$I_4 = \int_{\xi_j}^{\xi_{i+1}} \frac{\xi}{\sqrt{\xi^2+\eta_i^2}}\, K_1\left[\sqrt{s\;(\xi^2+\eta_i^2)}\right] d\xi \qquad \text{(B.31)}$$

Eq. (B.31) can **be** rewritten **as**

$$I_4 = \int_{\xi_j}^{\xi_{j+1}} -\frac{1}{\sqrt{s}}\left[\frac{d}{d\xi} K_0\left(\sqrt{s\,(\xi^2+\eta_i^2)}\right)\right] d\xi \tag{B.32}$$

Because

$$\frac{d}{dx} K_0\,(ax) = -\,aK_1\,(ax) \tag{B.33}$$

Thus the integral I_4 becomes

$$I_4 = -\frac{1}{\sqrt{s}}\left[K_0\left(\sqrt{s\,(\xi_{j+1}^2+\eta_i^2)}\right) - K_0\left(\sqrt{s\,(\xi_j^2+\eta_i^2)}\right)\right] \tag{B.34}$$

APPENDIX C

Singular Integral Evaluation

C.l Convolution BEM

An integral that arises when the field and collocation points coincide is

$$I_1 = \int_0^c \xi \, E_1 \left[\frac{\xi^2 + \zeta_i^2}{4(t - \tau)} \right] d\xi \tag{C.1}$$

At the lower limit, the integral seems to be singular. We will show that owing to the logarithmic nature of the singularity the integral is bounded. Let us consider a non-zero lower limit in Eq. (C.l) as b. Equation (C.1) can be integrated analytically and the results were presented in Eq. (B.30). Setting ζ to zero in Eq. (B.30) we obtain, at the lower limit

$$I_1\Big|_b = 2\,(t - \tau) \left\{ 1 - \exp\left[- \frac{b^2}{4(t - \tau)} \right] \right\} + \frac{1}{2} \left\{ b^2 E_1 \left[\frac{b^2}{4(t - \tau)} \right] \right\} \tag{C.2}$$

If $b = 0$ then the first term on the right hand side becomes zero. The second term is of the form

$$\lim_{x \to 0} x \, E_1 \left(\frac{x}{k}\right) \tag{C.3}$$

Expanding $E_1(\frac{x}{k})$ in ascending series

$$\lim_{x \to 0} x \, E_1 \left(\frac{x}{k}\right) = \lim_{x \to 0} x \left[-\gamma - \ln \frac{x}{k} - \sum_{n = l}^{\infty} \frac{(-1)^n \, (x/k)^n}{n \; n!} \right] \tag{C.4}$$

where γ is Euler's constant. Rewriting Eq. **(C.4)**

$$\lim_{x\to0} x\, E_1\left(\frac{x}{k}\right) = -\lim_{x\to0}(\gamma x) - \lim_{x\to0} x \ln x + \lim_{x\to0} x \ln k - \lim_{x\to0}\sum_{n=1}^{\infty}\frac{(-1)^n\,(x/k)^n}{n\;n!} \qquad (C.5)$$

The first, third and fourth term on the right go to zero. Also $\lim_{x\to0} x \ln x \longrightarrow 0$ because $\ln x$ grows much slower than any polynomial. Thus x goes to zero quicker than $\ln x$ grows. Thus, in the limit

$$I_1\Big|_0 = 0 \qquad (C.6)$$

Another singular integral encountered is of the form :

$$I_2 = \int_0^c E_1\left(\frac{x^2}{4t}\right) dx \qquad (C.7)$$

The integral has a logarithmic singularity at the lower limit. Two methods have been suggested previously to alleviate this problem. One requires expanding the exponential integral in ascending series and evaluating the integrals term by term. This produces an infinite sum that must be evaluated whenever the integral is evaluated, and is slowly convergent for large values of the argument. Another method suggested is to add and subtract a logarithmic term to the exponential integral. This removes the logarithmic singularity of $E_1(z)$ after which it can be integrated numerically. The remaining logarithmic term can be integrated analytically. We present here a simple closed form value of Eq. *(C.7)* in terms of smoothly behaved functions.

Rewriting Eq.*(C.7)*

$$I_2 = \int_0^c E_1(\alpha x^2)\, dx \qquad (C.8)$$

where

$$\alpha = \frac{1}{4t} \qquad (C.9)$$

Integrating once by parts

$$I_2 = x\,E_1(\alpha x^2)\Big|_0^c - \int_0^c \frac{d}{dx}\left[E_1(\alpha x^2)\right] dx \qquad \text{(C.10)}$$

$$I_2 = cE_1(\alpha c^2) - xE_1(\alpha x^2)\Big|_0 + 2\int_0^c \exp\,(-\alpha x^2)\,dx \qquad \text{(C.11)}$$

Since $E_1(\alpha x^2)$ grows logarithmically as x goes to zero, and hence grows slower than any polynomial, then x tends to zero faster than E_1 grows, as shown in Eq. (C.6). This implies that

$$x\,E_1(\alpha x^2)\Big|_0 \to 0 \;\; as \;\; x \to 0 \qquad \text{(C.12)}$$

thus

$$\int_0^c E_1(\alpha x^2)\,dx = cE_1(\alpha c^2) + \sqrt{\frac{\pi}{\alpha}}\; erf\;(\sqrt{\alpha c} \qquad \text{(C.13)}$$

where

$$erf\,(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp\,(-x^2)\,dx \qquad \text{(C.14)}$$

For large arguments, $E_1(\alpha c^2)$ goes to zero asymptotically and $erf\;(\sqrt{\alpha c})$ goes to one asymptotically. Thus a stable and easy to evaluate form is obtained. Resubstituting *a* from Eq. (C2) we obtain

$$\int_0^c E_1(x^2/4t)\,dx = cE_1(\frac{c^2}{4t}) + \sqrt{4\pi t}\;\, erf\;(\frac{c}{\sqrt{4t}} \qquad \text{(C.15)}$$

C.2 Laplace Space

The integrals singular at one of the limits are performed as follows. Some integrals of the form

$$I_3 = \int_0^c \xi\, K_0\,(\xi\sqrt{s}\,)\; d\xi \tag{C.16}$$

can **be** rewritten **as**

$$I_3 = \int_0^c \xi K_0\,(\xi\sqrt{s}\,)\; d\xi \;=\; c^2 \int_0^1 zK_0\,(c\sqrt{s}\,z)\; dz \tag{C.17}$$

Gradshteyn & Rhyzhik (1980) give the following integral

$$\int_0^1 x^{\nu+1}\, K_\nu\,(ax)\; dx \;=\; 2^\nu\, a^{-\nu-2}\Gamma(\nu+1) - a^{-1}K_{\nu+1}\,(a) \tag{C.18}$$

Customizing ***Eq. (C.18)*** for ***Eq. (C.17)***

$$c^2 \int_0^1 zK_0\,(c\sqrt{s}\,z)\; dz \;=\; c^2 \left[\frac{1}{(c\sqrt{s}\,)^2} - \frac{1}{(c\sqrt{s}\,)} K_1\,(c\sqrt{s}\,) \right] \tag{C.19}$$

$$\int_0^c \xi K_0\,(\xi\sqrt{s}\,)\; d\xi \;=\; \frac{c}{\sqrt{s}} \left[\frac{1}{c\sqrt{s}} \;-\; K_1\,(c\sqrt{s}\,) \right] \tag{C.20}$$

Another integral encountered is

$$I_4 = \int_0^c K_0\,(\xi\sqrt{s}\,)\; d\xi \tag{C.21}$$

I_4 can **be** rewritten as

$$I_4 = c \int_0^1 K_0\,(c\sqrt{s}\,z)\; dz \tag{C.22}$$

The ascending series valid for small arguments of the function is given **by** *Abramowitz* and ***Stegun (1970)*** **as**

$$\int_0^x K_0(t)\; dt \;=\; -\left[\gamma + \ln\frac{x}{2}\right] x \sum_{k=0} \frac{(x/2)^{2k}}{(k!)^2\,(2k+1)} \;+$$

$$x \sum_{k=0}^{\infty} \frac{(x/2)^{2k}}{(k!)^2\,(2k+1)^2} + x \sum_{k=1}^{\infty} \frac{(x/2)^{2k}}{(k!)^2\,(2k+1)} \left[1 + \frac{1}{2} + \ldots\ldots + \frac{1}{k} \right] \tag{C.23}$$

where, γ is Euler's constant. The large argument asymptotic series is defined as

$$x^{\frac{1}{2}} e^{x} \int_{x}^{\infty} K_0(t)\, dt = \left[\frac{\pi}{2}\right]^{\frac{1}{2}} \sum_{k=0}^{\infty} (-1)^k a_k\, x^{-k} \tag{C.24}$$

where, a_k are defined as

$$a_k = \frac{\Gamma(k+\frac{1}{2})}{\Gamma(\frac{1}{2})} \sum_{s=0}^{k} \frac{\Gamma(s+\frac{1}{2})}{2^s\, s!\Gamma(\frac{1}{2})} \tag{C.25}$$

In order to use Eq. (C.24) as the large argument asymptotic series for Eq. (C.22), the following identity is used.

$$I_4 = \int_0^x K_0(t)\, dt = \left[\int_0^{\infty} - \int_x^{\infty}\right] K_0(t)\, dt \tag{C.26}$$

but

$$\int_0^{\infty} K_0(t)\, dt = \frac{\pi}{2} \tag{C.27}$$

Thus

$$I_4 = \frac{\pi}{2} - x^{-\frac{1}{2}} e^{-x} \left[\frac{\pi}{2}\right]^{\frac{1}{2}} \sum_{k} = \overset{\infty}{0}\, (-1)^k a_k\, x^{-k} \tag{C.28}$$

where, the large argument asymptotic series given by Eq. (C.24) is used. The coefficients a_k in Eq. (C.28) are defined in Eq. (C.25). The two equations Eq. (C.23) and Eq. (C.28) are used to calculate the integral. The switch from the ascending series to the asymptotic series depends upon the computer used. For our purposes the switch was made at an argument of 11.2.

APPENDIX D

Non-Formal Proof for the Convolution Matrix

D.l Origin of Convolution Structure and Incremental Solution

The time step sizes for the convolution **BEM** are assumed **to** be constant. In order to show the development of the convolution structure at the matrix level let us assume that a problem with constant pressure outer boundary conditions is being solved. The **BEM** first evaluates the fluxes at all locations on the boundary. Accounting for the boundary condition, the integral equation looks as follows

$$\int_t d\tau \int_\Gamma G \frac{\partial p_D}{\partial n}\, dS = 0 \tag{D.1}$$

Suppose both the space and time interpolation functions are constant. This means that the element value of the variables are replaced by the average of the two adjacent nodal values. For illustration purposes, assume that the total solution time is divided into three steps of size At. The respective time levels can be written

$$t_1 = \Delta t \qquad t_2 = 2\,\Delta t \qquad t_3 = 3\,\Delta t \tag{D.2}$$

Substituting the Green's function in Eq. (D.l) the solution at an element for the first time step is

$$I = p_{D_{n1}} \int_0^{t_1} \frac{d\tau}{t_1 - \tau} \exp\left[- \frac{r^2}{4(t_1 - \tau)} \right] = p_{D_{n1}} \left[E_1(\frac{1}{\Delta t}) - E_1(\frac{1}{0}) \right] \tag{D.3}$$

where $p_{D_{n1}}$ denotes the flux at time step 1, and the distance from the collocation to the field point is one. The last term in Eq. (D.3) is zero. At the second time step, since the flux is piecewise constant in time, the element integral is

$$I = p_{D_{n1}} \int_0^{t_1} \exp\left[-\frac{r^2}{4(t_2-\tau)}\right]\frac{d\tau}{(t_2-\tau)} + p_{D_{n2}} \int_{t_1}^{t_2} \exp\left[-\frac{r^2}{4(t_2-\tau)}\right]\frac{d\tau}{(t_2-\tau)} \qquad \text{(D.4)}$$

Integrating Eq. (D.4),

$$I = p_{D_{n1}}\left[E_1(\frac{1}{t_2-0}) - E_1(\frac{1}{t_2-t_1})\right] + p_{D_{n2}}\left[E_1(\frac{1}{t_2-0}) - E_1(\frac{1}{t_2-t_2})\right] \qquad \text{(D.5)}$$

Since the time step size is constant, Eq. (D.5) can be rearranged as

$$I = [p_{D_{n2}} - p_{D_{n1}}]\, E_1(\frac{1}{\Delta t}) + p_{D_{n1}}\, E_1(\frac{1}{2\Delta t}) \qquad \text{(D.6)}$$

Denoting the difference between the flux values at times t_1 and t_2 as $\Delta p_{D_{n2}}$ Eq. (D.6) can be rewritten as

$$I = \Delta p_{D_{n2}}\, E_1(\frac{1}{\Delta t}) + p_{D_{n1}}\, E_1(\frac{1}{2\Delta t}) \qquad \text{(D.7)}$$

Similarly, at the third time step the integral can be broken into three parts as

$$I = \left[p_{D_{n1}} \int_0^{t_1} + p_{D_{n2}} \int_{t_1}^{t_2} + p_{D_{n3}} \int_{t_2}^{t_3}\right| \left\{ \frac{d\tau}{4(t_3-\tau)} \exp\left| -\frac{1}{4(t_3-\tau)}\right]\right\} \qquad \text{(D.8)}$$

Integrating and rearranging the terms

$$I = \Delta p_{D_{n3}}\, E_1(\frac{1}{\Delta t}) + \Delta p_{D_{n2}}\, E_1(\frac{1}{2\Delta t}) + \Delta p_{D_{n1}}\, E_1(\frac{1}{3\Delta t}) \qquad \text{(D.9)}$$

where

$$\Delta p_{D_{n3}} = p_{D_{n3}} - p_{D_{n2}} \qquad \text{(D.10)}$$

One thing obvious from the above procedure is that the solution obtained at each time step is incremental and a running sum must be kept in the algorithm to get the cumulative result at the end of a particular time step. Also, the convolution character of the coefficients is evident in Eq. (D.9). The flux at the third time slot ($p_{D_{n3}}$) is multiplied by the coefficient obtained from the first time slot (At) and the flux at the first time slot ($p_{D_{n1}}$) is multiplied by the coefficient from the third time slot $(3At)$, and so on.

The coefficient $E_1(\frac{1}{At})$ always multiplies the unknown quantity $[Ap_{D_{n2}}$ in Eq. (D.3) and $A\,p_{D_{n3}}$ in Eq. (D.9)]. This implies that the coefficient matrix for the unknown vector remains the same throughout. Only the right hand side vector is modified at each time level. The left hand coefficient matrix can thus be inverted and stored once and for all. This was also observed by Pina (1984). Matrix level rather than element level treatment will make this clearer, and is given in the next section.

D.2 Matrix Computation for Convolution BEM

Appendix D.1 dealt with a specific boundary condition and the interpolation function in space. This may be generalized to the matrix level. At the first time level, the augmented matrix equation is

$$\left[H_{11}\right]\left[u_1\right] = 0 \qquad \text{(D.10)}$$

Separating the right hand side vector, we obtain

$$\left[H_{11}\right]\left[u_1\right] = -\left[H'_{11}\right]\left[u_1\right]_\Gamma \qquad \text{(D.11)}$$

where, Γ denotes the vector of boundary conditions. Solution of Eq. (D.11) gives the unknown vector u_1.

$$u_1 = \left[H_{11}\right]^{-1}\left\{\left[H'_{11}\right]\left[u_1\right]_\Gamma\right\} \qquad \text{(D.11)}$$

At the second time level a similar procedure yields

$$\left[H_{11}\right]\left[\Delta u_2\right] = -\left[H_{21}\right]\left[u_1\right] - \left[H'_{11}\right]\left[\Delta u_2\right]_\Gamma - \left[H'_{21}\right]\left[u_1\right]_\Gamma \qquad \text{(D.12)}$$

Since u_1 is known from the previous time step, it is moved to the right hand side. The coefficient matrix on the left hand side is the same matrix $[H_{11}]$, as before. It may be seen from **Eq. (D.12)** that on the right hand side, the new matrix generated at the second time step namely, $[H_{21}]$ is multiplied **by** the solution at the first time level. **A** u_2 in the second term on the right hand side is the change in the boundary condition over the time step. For time dependent boundary conditions this term is non-zero.

To show the convolution of the right hand side matrices clearly, consider the third time step. The resulting matrix equation is

$$\left[H_{11}\right]\left[\Delta u_3\right] = -\left\{\left[H_{21}\right]\left[\Delta u_2\right] + \left[H_{31}\right]\left[u_1\right]\right\} -$$

$$\left\{\left[H'_{11}\right]\left[\Delta u_3\right]_\Gamma + \left[H'_{21}\right]\left[\Delta u_2\right]_\Gamma + \left[H'_{31}\right]\left[u_1\right]_\Gamma\right\} \tag{D.13}$$

Eq. (D.13) and Eq. (4.12b) are equivalent, if the diagonal structure of the matrix array in Fig. (4.1) is considered.

APPENDIX E

I/O for Laplace Space BEM Simulator

E.1 Setup of Datafile for BEM Simulator

SN.	Variable	Comments
1.	Title	[80 Characters (maximum)]
2.	Logistics	Well name, field name, etc. [80 characters]
3.	ϕ	Porosity
4.	k_x, k_y	X-direction and Y-direction Permeabilities (Darcy)
5.	μ, c_t	Viscosity (cp) and Compressibility (atm^{-1})
6.	INUM, JNUM	Output of Bdry. Press. from Point INUM to JNUM
7.	ICB, **NBEGIN**	No. of Internal Boundaries [currently one], Beginning Node No. for Internal Boundary
8.	KEY	0 --> No Diagnostics 1 --> Diagnostics
9.	N	Number of Boundary Nodes
10.	X-Coord., Y-Coord., p_D, $p_{D_{nl}}$, $p_{D_{na}}$, CONST, **ID**	Nodal Location (cm.) and Boundary Values (atm, atm/cm), [Repeated N times]
11.	PINIT	Initial Boundary Pressure (atm.)

12.	**NW**	Number of Line Source Wells
13.	xw, YW, NFLOW	Well Location (cm.) and No. of Flow Periods
14.	QW, **TP**	Prod. Rate (cm^3/sec) **and** Starting Time [Repeated NFLOW Times]
15.	**IOPT**	Time Stepping Option
	If **IOPT=0**	Cartesian Time Stepping
16.	TMAX	Maximum Solution Time (seconds)
17.	TO	Starting Time (seconds)
18.	At	Step Size (seconds)
	If IOPT=1	Logarithmic Time Stepping
16.	**ILOG**	Number of **Log** Cycles
17.	ICYC	Number of Points/Log Cycle
18.	TDSM	Starting Time (seconds)
19.	**IGANPT**	N-point Gauss Integration Rule
20.	**ISFINV**	N-point Laplace Space Inversion
21.	**INT**	Number of Internal Points
22.	xx, YY	Coordinates of Internal Points (cm) [Repeated INT Times
23.	C_D, **S**	Wellbore Storage, Skin [Dimensionless]

The datafile shown above is set up for input of variables in Darcy **units.** The input variables are cast into dimensionless variables internally in the program. The output is presented in dimensionless time (t_{DA}) and pressure (p_D) format.

E.2 Sample Datafile

```
PROBLEM #14A
WELL IN THE CENTER OF A CLOSED EQUILATERAL TRIANGLE
1.0d+00
1.0d+00   1.0d+00
1.0d+00   1.0d+00
6   7
0
12
-0.7598356   -0.4386913   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
-0.5698767   -0.1096728   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
-0.3799178   +0.2193456   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
-0.1899589   +0.5483641   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.0000000   +0.8773826   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
4.1899589    +0.5483641   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.3799178   +0.2193456   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.5698767   -0.1096728   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.7598356   -0.4386913   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.3799178   -0.4386913   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
+0.0000000   -0.4386913   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
-0.3799178   -0.4386913   0.0d+00   0.0d+00   0.0d+00   0.0d+00   4
0.0d+00
1
0.0d+00   0.0d+00   1
6.28318   0.0d+00
```

```
1
4
10
0.001
6
8
1
0.0   0.0005
0.0   0.0
```

E.3 Sample Output File

```
41.0000000000000     1
   1.00000021754510E-03   4.55150082073090   0.500002138825894
   1.25892568566722E-03   4.66662355046387   0.500005697999735
   1.58489353724685E-03   4.78174759969168   0.500008340973189
   1.99526274902841E-03   4.89687268229402   0.500010283509125
   2.51188697795814E-03   5.01199851234593   0.500011658162106
   3.16227834810636E-03   5.12712466604652   0.500012492190590
   3.98107257159755E-03   5.24225039041891   0.500012834094193
   5.01187342658091E-03   5.35737483617244   0.500013574725628
   6.30957481741862E-03   5.47249896966325   0.500018240563590
   7.94328407526483E-03   5.58762977854023   0.500032519554669
   1.00000021754508E-02   5.70278235404304   0.500052402862917
   1.25892568566720E-02   5.81796879366257   0.500040131472140
   1.58489353724683E-02   5.93317042180628   0.499935008270483
   1.99526274902838E-02   6.04832658428773   0.499815225822297
```

2.51188697795811E-02	6.16341825553511	0.500315970372134
3.16227834810631E-02	6.27871601916173	0.503227818186001
3.98107257159751E-02	6.39516525905925	0.511952172514025
5.01187342658086E-02	6.51476772778330	0.531474330190670
6.30957481741852E-02	6.64082080169802	0.567880315724269
7.94328407526472E-02	6.77802265535270	0.627881253904955
0.100000021754507	6.93259273683730	0.718806415889670
0.125892568566718	7.11252924597900	0.849034263757746
0.158489353724680	7.32797079262657	1.02842527745842
0.199526274902835	7.59155032959813	1.26849646469117
0.251188697795807	7.91872674262002	1.58260024272634
0.316227834810627	8.32823484824593	1.98659709769266
0.398107257159746	8.84284020037596	2.50021739272089
0.501187342658079	9.49050283530347	3.14888554744700
0.630957481741845	10.3059567994461	3.96567235638688
0.794328407526462	11.3326913573696	4.99326740416317
1.00000021754505	12.6253712723555	6.28614539199894
1.25892568566716	14.2528058419577	7.91323256133403
1.58489353724678	16.3016353994230	9.96136759372297
1.99526274902832	18.8809487894316	12.5398009322737
2.51188697795804	22.1280899252805	15.7859646756708
3.16227834810622	26.2159684014410	19.8727891091795
3.98107257159740	31.3622651562781	25.0179277318239
5.01187342658071	37.8410233918702	31.4953659101767
6.30957481741836	45.9972415772979	39.6500249763780
7.94328407526451	56.2652453030240	49.9161385583680
10.0000021754504	69.1918161374446	62.8403809091606

Appendix F

Computer Program

```
c***********************************************************************
c     PROGRAMMAIN
c
c***********************************************************************
c     Features :
c                 o   Boundary Element Method
c                 o   Two-dimensional
c                 o   Single phase
c                 o   Homogeneous, Anisotropic
c                 o   Slightly compressible
c     Fluid Flow through Porous Media
c     Diffusivity Equation Solved in Laplace Space
c
c
c                            JITENDRA KIKANI
c                          STANFORD UNIVERSITY
c
c_______________________________________________________________________
c     Characterisitics :
c                   o      Pressure and Semi-log pressure derivative
c                   o      Logarithmic or Cartesian time stepping
c                   o      nw line source wells
c                   o      Rate decline
c                   o      Dirichlet, Neumann or Radiation Boundary Conds
c                   o      Laplace inversion by Stehfest algorithm
c                   o      Matrix Inversion by Gauss Jordan elimination
c                          with pivoting
c                   o      Numerical integration with Gauss-Legendre
c                   quadrature, 2 panels, 6 point schemes in each
c                   o      Calculation of reservoir area by triangulation
c                   o      Local Coordinate system based on fictitious
c                          source point
c***********************************************************************
c
c     Nodal Identification numbers :
c          ID #             (known)           (unknowns)
c           1    -------->   pnl, p                  pna
c           2    -------->   pna, p             pnl
c           3    -------->    p                pnl, pna
c           4    -------->   pnl,pna             p
c           5    -------->   pna,               p,pnl
c                            a p + b pnl=c
c           6    -------->   pnl,               p,pna
c                            a p + b pna=c
c           7    -------->   p + a pnl=c        p,pnl.pna
c                            p + a pna=c
c           8    -------->  p + a pnl=c        p,pnl,pna
c                            p + b pna=d
```

```
c-----------------------------------------------------------------------
c
c     Subroutines Referred :
c          Subroutines          Function Subprograms
c            Ainteg.f                  Angle.f
c            Aintern.f                 Area.f
c            Assemble.                  fxk0.f
c            Gauss.f                   xik0.f
c            Grule.f                   xk1.f
c            Imsl function K0,K1        xik1.f
c            Matinvert.f                aIntK.f
c            Matmult.f
c            Scgen.f
c            Steh.f
c            Source.f
c            Well.f
c
C-----------------------------------------------------------------------
c
c     CONTROL STATEMENTS
c
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER TITLE*80, LOGISTIC*80
      DIMENSION X(100),Y(100),P(100),XW(20),YW(20),TW(20),TWD(20),T(500)
     &,TP(20,10)

      DIMENSION XX(10),YY(10)
c
      COMMON/INV/N,NW,XD(100),YD(100),ID(100),NFLOW(10),XWD(20),YWD(20),
     &QW(20,10),TPD(20,10),MOB,KEY,IGANPT,INT,GINV(100),PD(100),PNL(100)
     &,PNA(100),ALPHA(100),BETA(100),CONST(100),CONS(100),CHID(100),
     &CHI(100),XXD(10),YYD(10),PP(10),PPD(10),CD,SKIN
      COMMON/COEFGEN/V(50)
      COMMON/SINTASS/ COEFP(100,100),COEFPL(100,100),COEFPA(100,100)
      COMMON/FUNC/ETA,S
c
c     Input Data
c
      READ(5,9100) TITLE
      READ(5,9100) LOGISTIC
 9100 FORMAT(A80)
c
c     Petrophysical and Fluid Properties
c
      READ(5,*) POR
      READ(5,*) PERMX,PERMY
      READ(5,*) VISC,COMP
c
```

```
C     40 for boundary pressure
C
      READ(5,*) INUM,JNUM
C
C     No. of internal boundaries (currently only one), Starting node no.
C     for the internal boundary
C
      READ(5,*) ICB,NBEGIN
C
C     Geometrical Data and Boundary Conditions
C
      READ(5,*) KEY
      READ(5,*) N
      READ(5,*) (X(I),Y(I),P(I),PNL(I),PNA(I),CONST(I),ID(I), I=1,N)
      READ(5,*) PINIT
C
C     Well(source/sink) Data
C       NW = Number of Wells
C       NFLOW = Number of Flow periods for each well
C
      READ(5,*) NW
           IF (NW .EQ. 0) GO TO 0009
         DO 1212 I=1,NW
            READ(5,*) XW(I),YW(I),NFLOW(I)
            DO 1213 J=1,NFLOW(I)
               READ(5,*) QW(I,J),TP(I,J)
 1213         CONTINUE
 1212      CONTINUE
C
C     Transient Solution Data
C
 0009       READ(5,*) IOPT
           IF (IOPT .EQ. 0) THEN
C
C          Cartesian time stepping
C
              READ(5,*) TMAX
              READ(5,*) TO
              READ(5,*) DELT
           ELSE
C
C          Logarithmic time stepping
C
              READ(5,*) ILOG
              READ(5,*) ICYC
              READ(5,*) TDSM
           ENDLF
                IF (IOPT .EQ. 1) THEN
```

```
                 IRDNO = ICYC*ILOG
                 RTD = 10.0D+00**(1.0D+00/ICYC)
                 T(1) = TDSM
                   DO 1225 I=2,IRDNO+1
 1225                T(I) = TDSM*(RTD)**(I-1)
                ENDIF
C
c      Gaussian Quadrature Rule(N-point)
c
       READ(5,*) IGANFT
C
c      Number of terms in Laplace Inversion Routine
C
       READ(5,*) ISFINV
c
c      Internal Solution Data
C
       READ(5,*) INT
            IF (INT .EQ. 0) GO TO 73
       READ(5,*) (XX(I),YY(I), I=1,INT)
 73    CONTINUE
C
c       Wellbore Storage and Skin
C
        READ(5,*) CD,SKIN
c
C
c      Echo Input Data
C
       WRITE(6,9200)
 9200 FORMAT (/5X,'Boundary Integral Equation Method',/10X,'Solution of
     &Diffusivity Equation')
       WRITE(6,9300) TITLE
 9300 FORMAT(/1X,A80)
       WRITE(6,9400)
 9400 FORMAT(/4X, 'I',4X,'X-Cood',4X,'Y-Cood',5X,'Pot',5X,'Pn(lv.)',
     &4X,'Pn(app.)',4X,'Const',4X,'Id')
          DO 9444 I=1,N
 9444    WRITE(6,9500) I,X(I),Y(I),P(I),PNL(I),PNA(I),CONST(I),ID(I)
 9500    FORMAT(3X,I2,3X,F6.2,3X,F6.2,3X,F7.3,4X,F7.3,4X,F7.3,4X,F7.3,
     &   3X,I3)
       WRITE(6,9600) POR,PERMX,PERMY,VISC,COMP
 9600 FORMAT(/5X,'Porosity = ',F5.3,/5X,'X-Permeability = ',F5.3,'Dar
     &cies',1OX,'Y-Permeability=',F5.3,' Darcies',/5X,'Viscosity = '
     &,F5.2,' Centipoise',10X,'Compressibility =',F7.5)
       WRITE(6,9700) TMAX,DELT
 9700 FORMAT(10X,'Max. Simulation Time =',F5.1,8X,'Time Step Size =',
     &F7.4)
```

```
C
C---------------------------------------------------------------------
C
C     Non-Dimensionalize variables and parameters
C
      IBUG = KEY
      AR = AREA(N-NBEGIN+1,X,Y,1,N-NBEGIN+1,IBUG)
           DO  10 I=1,N
             XD(I) = X(I)/DSQRT(AR)
             YD(I) = Y(I)*DSQRT(PERMX/(PERMY*AR))
 10         CONTINUE
               DO 20 I=1,INT
                 XXD(I) = XX(I)/DSQRT(AR)
                 YYD(I) = YY(I)*DSQRT(PERMX/(PERMY*AR))
 20             CONTINUE
      DLFF = PERMX/(POR*VISC*COMP)
      MOB = PERMX/VISC
              IF (IOPT .EQ. 0) THEN
                 TDMAX = TMAX*DIFF/AR
                 TOD = T0*DIFF/AR
                 DTD = DELT*DIFF/AR
              ENDIF
                   IF (NW .EQ. 0) GO TO 40
                        DO 30 1=1,NW
                            XWD(I) = XW(I)/DSQRT(AR)
                            YWD(I) = YW(I)*DSQRT(PERMX/(PERMY*AR))
                            DO 37 J=1,NFLOW(I)
                              TPD(I,J) = TP(I,J)*DIFF/AR
 37                          CONTINUE
 30                      CONTINUE
 40     CONTINUE
            DO 50 I=1,N
               IF (ID(I) .EQ. 1 .OR. ID(I) .EQ. 2 .OR. ID(I) .EQ. 3) THEN
               PD(I) = P(I) - PINIT
               ENDIF
 50          CONTINUE
 C
 C      If Boundary Conditions are of TYPE 3. The Value of Constants are
 C      Given in the Input P,PNL or PNA. Excoriate them.
 C
                  DO 60 J=1,N
                      IF (ID(J) .EQ. 5) THEN
                      ALPHA(J) = P(J)
                      BETA(J) = PNL(J)
                      P(J) = 0.00D+00
                      PNL(J) = 0.00D+00
                      ENDIF
                          IF (ID(J) .EQ. 6) THEN
```

```
                    ALPHA(J) = P(J)
                    BETA(J) = PNA(J)
                    P(J) = 0.00D+00
                    PNA(J) = 0.00D+00
                    ENDIF
                      IF (ID(J) .EQ. 7 .OR. ID(J) .EQ. 8) THEN
                      ALPHA(J) = PNL(J)
                      BETA(J) = PNA(J)
                      CHI(J) = P(J)
                      P(J) = 0.00D+00
                      PNL(J) = 0.00D+00
                      PNA(J) = 0.00D+00
                      ENDIF
 60              CONTINUE
c
c
c      Create Plot Files
c
       OPEN (UNIT=1, FILE='pres.plt')
       rewind 1
       IF (IOPT .EQ. 0) THEN
         AAAA = TMAX/DELT
       ELSE
         AAAA = IRDNO+1
       ENDIF
       NANUM = JNUM - INUM + 1
        WRITE(1,*) AAAA,NANUM

       OPEN (UNIT=2, FILE='int.plt')
       REWIND 2
        WRITe(2,*) AAAA,INT
c
c
c
                 CALL SCGEN(ISFINV)
c
c     Begin the time loop.
c
      TSTEP = 1.
      IF (IOPT .EQ. 0) TTOT = TOD
         DO 8888 J=1,1500
            IF (IOPT .EQ. 1) then
                IF (J .GE. IRDNO+2) GO TO 9999
            TD = T(J)*DIFF/AR
            ELSE
            T D=  “TOT
            ENDIF
```

```
c     Reinitialize the unknowns
            DO 1732 K=1,N
            GO TO (21,22,23,24,25,26,27,27), ID(K)
C
 21           PNA(K) = 0.0d+00
            GO TO 1732
 22           PNL(K) = 0.0d+00
            GO TO 1732
 23           PNL(K) = 0.0d+00
            PNA(K) = 0.0d+00
            GO TO 1732
 24           PD(K) = 0.0d+00
            GO TO 1732
 25           PD(K) = 0.0d+00
            PNL(K) = 0.0d+00
            GO TO 1732
 26           PD(K) = 0.0d+00
            PNA(K) = 0.0d+00
            GO TO 1732
 27           PD(K) = 0.0d+00
            PNL(K) = 0.0d+00
            PNA(K) = 0.0d+00
 1732         CONTINUE


              CALL STEH(YD,AR,ISFINV,ICB,NBEGIN)
c
c        Allocate boundary solutions to proper unknowns
c
            DO 2000 K=1,N
            GO TO (1,2,3,4,5,6,7,7), ID(K)
C
 1           PNA(K) = GINV(K)
            GO TO 2000
 2           PNL(K) = GINV(K)
            GO TO 2000
 3           PNL(K) = GINV(K)
            PNA(K) = PNL(K)
            GO TO 2000
 4           PD(K) = GINV(K)
            P(K) = PD(K) + PINIT
            GO TO 2000
 5           PD(K) = GINV(K)
            P(K) = PD(K) + PINIT
            PNL(K) = CONS(K)/BETA(K) - ALPHA(K)/BETA(K)*PD(K)
            GO TO 2000
 6           PD(K) = GINV(K)
            P(K) = PD(K) + PINIT
```

```
                PNA(K) = CONS(K)/BETA(K) - ALPHA(K)/BETA(K)*PD(K)
                GO TO 2000
 7              PD(K) = GINV(K)
                P(K) = PD(K) + PINIT
                PNL(K) = CHID(K)/ALPHA(K) - PD(K)/ALPHA(K)
                PNA(K) = CONS(K)/BETA(K) - PD(K)/BETA(K)
 2000             CONTINUE

          DIMT = TD*AR/DIFF
C
C.......................  PRINT THE BOUNDARY SOLUTIONS--------.----------
C
        IF (KEY .EQ. 1) THEN
          WRITE(6,8 100)
 8100       FORMAT(//X,' .........BOUNDARY SOLUTIONS........')
          WRITE(6,8200) TSTEP,DELT,DIMT
 8200       FORMAT(/5X,'TIME STEP NO.   ',F5.1,15X,'STEP SIZE   ',F6.3,
     &      /14X,'TOTAL SIMULATION TIME  = ',F7.3)
          WRITE(6,8300)
 8300       FORMAT(/3X,'NODE',3X,'X-COOD',3X,'Y-COOD',9X,'P',7X,'FLUX(LVG
     &.)',4X,'FLUX(APP.)',6X,'ID')
                DO 8222 I=1,N
 8222             WRITE(6,8400) I,X(I),Y(I),P(I),PNL(I),PNA(I),ID(I)
 8400             FORMAT(/3X,I3,2X,F7.3,3X,F7.3,2X,F12.6,4X,F10.6,4X,
     &            F10.6,5X,12)
        ENDIF
c
c           Write the Graphing Information in Plot File
c
      WRITE(1,*) DIMT, P(INUM),(P(KKK), KKK=INUM+1,JNUM-1),P(JNUM)
c
 6000       CONTINUE
c
c           Write the Interior Solutions in Plot File
c
      WRITE(2,*) TD,(PP(KKK),TD*PPD(KKK), KKK=1,INT)

          TSTEP = TSTEP + 1.
             IF (IOF'T .EQ. 0) THEN
                IF (TD .GE. TDMAX) GO TO 9999
             TTOT = TTOT + DTD
             ENDIF
 8888       CONTINUE
 9999       STOP
          END
      SUBROUTINE STEH(T,AR,ISFINV,ICB,NBEGIN)
c-----------------------------------------------------------------------
c     This routine samples the solution at a number of values
c     of the laplace parameter S and calculates the inverse
c     after setting up the matrix and solving it at every
c     parameter value
c
c  T   - TIME AT WHICH INVERSE IS WANTED.
c  ISFINV   - PARAMETER THAT GOVERNS THE ACCURACY.
c        ISFINV MUST BE EVEN.
c        ISFINV IS ROUGHLY EQUIVALENT TO THE NUMBER OF DIGITS
c        WITH WHICH THE COMPUTER IS WORKING.
c        E.G.  HONEYWELL CP-5
c             PRECISION  DIGITS   N
c              SINGLE      8      8
c              DOUBLE      16     16
c-------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PDNS(100),PNLS(100),PNAS(100),G(100),H(100,100),TPP(10)
      COMMON/INV/N,NW,XD(100),YD(100),ID(100),NFLOW(10),XWD(20),YWD(20),
     &QW(20,10),TPD(20,10),MOB,KEY,IGANPT,int,GINV(100),PD(100),PNL(100)
     &,PNA(100),ALPHA(100),BETA(100),CONST(100),CONS(100),CHID(100),
     &CHI(100),XXD(10),YYD(10),PP(10),PPD(10),CD,SKIN
      COMMON/COEFGEN/ V(50)
      COMMON/FUNC/ETA,S
      COMMON/SSOURCE/ AINHOM(100)
      COMMON/SINTASS/ COEFP(100,100),COEFPL(100,100),COEFPA(100,100)
c
      DATA DLN2/.6931471805599453D0/
c
c     Very Important to Initialize ginv, and pp because they may
c     Contain Numbers from Previous Time Steps
c
        DO 73 J = 1,N
          GINV(J) = 0.0d+00
 73     CONTINUE

        DO 67 J=1,INT
          PP(J) = 0.0d+00
          PPD(J) = 0.0d+00
 67     CONTINUE
c
c     Setting the Limit on the Matrix Inversion Tolerance
c
        EPS = 1.0d-15

       AD  = DLN2/T
        DO 11 J= 1,ISFINV
```

```
         S   = AD*J
C
          DO 70 K=1,N
            PDNS(K) = PD(K)/S
            PNLS(K) = PNL(K)/S
            PNAS(K) = PNA(K)/S
            IF (ID(K) - 6) 100,200,300
 100          CONS(K) = CONST(K)/S
            GO TO 70
 200          CONS(K) = CONST(K)/S
            GO TO 70
 300          CHID(K) = CHI(K)/S
            CONS(K) = CONST(K)/S
 70         CONTINUE
C
C
C         Evaluate the Coefficients of Known and Unknown Variables
C
         CALL AINTEG(N,XD,YD,IGANPT,ICB,NBEGIN,KEY)
C
            CALL ASSEMBLE(N,ID,PDNS,PNLS,PNAS,ALPHA,BETA,CONS,CHID,H,
    &        G,KEY)

C
C          Add the Inhomogeneity to the RHS Vector
C
          CALL SOURCE(N,NW,S,T,AR,QW,TPD,NFLOW,XD,YD,XWD,YWD)
C
          DO 1397 LL=1,n
           G(LL) = G(LL) + AINHOM(LL)
 1397      CONTINUE
C
C          LHS Matrix of Unknown Vector, Inverted by Gauss-Jordan
C          Elimination with Pivoting and Stored.
C
C
             CALL MATINV(N,H,EPS,-1,DTNRM)

                 IF (DTNRM .LT. 1.0D-15) THEN
                 WRITE(6,9800)
 9800              FORMAT(/'MATRIX IS ILL CONDITIONED')
                 ENDIF
C
C         MULTIPLY THE STORED INVERSE WITH RHS VECTOR TO GET SOLN.
C
         CALL MATMULT(N,H,G)

C               Calculate the Interior Solution, at Each Sampling Point
```

```
C             Perform the Laplace Inversion in the Same Fashion as
C             Boundary Point.
C
        CALL AINTERN(N,XD,YD,ID,INT,XXD,YYD,NW,XWD,YWD,QW,TPD,NFLOW,
    &                T,G,IGANPT,TPP,PDNS,PNLS,PNAS,AR,ICB,NBEGIN)

C
C         Inversion of the solution. This solution sampled isfinv times
C
            DO 3750 LL = 1,N
              GINV(LL) = GINV(LL) + V(J)*G(LL)
 3750         CONTINUE
C
C     Inclusion of Wellbore Storage and Skin
C
             DO 3755 LL =1,INT
               TMP = S/(SKIN + S*TPP(LL))
               TPP(LL) = 1./(CD*S*S + TMP)
               PP(LL) = PP(LL) + V(J)*TPP(LL)
               PPD(LL) = PPD(LL) + V(J)*S*TPP(LL)
 3755          CONTINUE

 6000         CONTINUE
  11     CONTINUE
           DO 79 LL = 1,N
             GINV(LL) = AD*GINV(LL)
 79          CONTINUE
              DO 85 LL=1,INT
                PP(LL) = AD*PP(LL)
                PPD(LL) = AD*PPD(LL)
 85             CONTINUE
      RETURN
      END


      SUBROUTINE AINTEG(N,XD,YD,IGANPT,ICB,NBEGIN,KEY)
C
c-----------------------------------------------------------------------
c     This routine carries out the integration and compiles the
c     coefficient matrices.
c
c--------------------------SUBROUTINES REFERRED-----------------------
c     GAUSS.NPT.F
c     MMDEI.F
c     DERF.F
c     ANGLE.F
c     FUNC_XPIXPO.F
c     FUNC_SNGLAR.F
```

```
c-----------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XD(100),YD(100),ALPHA(100)
      COMMON/SINTASS/ COEFP(100,100),COEFPL(100,100),COEFPA(100,100)
      COMMON/func/ETA,S
      EXTERNAL BESK0,BESK1,aIntK,XIK1,XK1,XIK0,XK0
C
c     Clearing the Subscripted Variables
C
      L = N + 1
         DO 20 1=1,N
            DO 10 J = l,N
            COEFP(I,J) = 0.0D0
            COEFPL(I,J) = 0.0D0
            COEFPA(I,J) = 0.0D0
 10          CONTINUE
         COEFP(I,L) = COEFP(I,1)
         COEFPA(I,L) = COEFPA(I,1)
 20       CONTINUE

         SS = DSQRT(S)
C
c     Compiling the Coefficients of Pressures and Derivatives for each
c     Collocation Point
C
         IF (ICB .EQ. 1) THEN
           XD(L) = XD(NBEGIN)
           YD(L) = YD(NBEGIN)
         ELSE
           XD(L) = XD(1)
           YD(L) = YD(1)
         ENDIF
       R1 = DSQRT((XD(L) - XD(N))**2 + (YD(L) - YD(N))**2)
       CO = (XD(L) - XD(N))/R1
       SI = (YD(L) - YD(N))/R1
       THETA1 = DATAN2(SI,CO)
          DO 500 1=1,N
             DO 400 J=1,N
C
C             Switching to a Local Co-ordinate System
C
              IF (ICB .EQ. 1) THEN
                IF (J .EQ. NBEGIN-1) THEN
                  R = DSQRT((XD(1) - XD(J))**2 + (YD(1) - YD(J))**2)
                  CO = (XD(1) - XD(J))/R
                  SI = (YD(1) - YD(J))/R
                  XIA = (XD(J) - XD(I))*CO + (YD(J) - YD(I))*SI
              XIB = (XD(1) - XD(I))*CO + (YD(1) - YD(I))*SI
              SIGNRN = {XD(1)-XD(J))*(YD(J)-YD(I)) -
     &         (XD(J)-XD(I))*(YD(1)-YD(J))
              GO TO 67
            ENDIF
          ENDIF
        R = DSQRT((XD(J+1) - XD(J))**2 + (YD(J+1) - YD(J))**2)
        CO = (XD(J+1) - XD(J))/R
        SI = (YD(J+1) - YD(J))/R
        XIA = (XD(J) - XD(I))*CO + (YD(J) - YD(I))*SI
        XIB = (XD(J+1) - XD(I))*CO + (YD(J+1) - YD(I))*SI
        SIGNRN = (XD(J+1)-XD(J))*(YD(J)-YD(I)) -
     &   (XD(J)-XD(I))*(YD(J+1)-YD(J))
 67       ETA = (YD(J) - YD(I))*CO - (XD(J) - XD(I))*SI
        DIFF = XIB - XIA
        ASQ = XIA*XIA + ETA*ETA
        SSQ = XIB*XIB + ETA*ETA

          IF (ICB .EQ. 1) THEN
            IF (J .EQ. I) GO TO 100
              IF (I .EQ. NBEGIN .AND. J .EQ. NBEGIN-1) GO TO 150
                IF (I .EQ. 1 .AND. J .EQ. NBEGIN-1) GO TO 200
                  IF (I .EQ. NBEGIN .AND. J .EQ. N) GO TO 200
                    IF (I .EQ. J+1) GO TO 200
          ELSE
            IF (J .EQ. I) GO TO 100
              IF (IEQ. J+1 .OR. I+N .EQ. J+1) GO TO 200
          ENDIF
C
C        Integrals and Function Evaluations
C
 150      CONTINUE
         XIKlSUM = (BESK0(SS*DSQRT(ASQ)) - BESK0(SS*DSQRT(BSQ)))/SS
         CALL GAUSS(IGANPT,XK1,1,XIA,XIB,2,XK1SUM,IND)
         CALL GAUSS(IGANPT,XIK0, 1,XIA,XIB,2,XIK0SUM,IND)
         CALL GAUSS(IGANPT,XK0, 1,XIA,XIB,2,XK0SUM,IND)
C
         PBEl 1 = ETA*SS*XIK1SUM/DIFF
         PBE12 = ETA*SS*XK1SUM/DIFF
         PNBE21 = XIK0SUM/DIFF
         PNBE22 = XK0SUM/DIFF
         PBEl = XIB*PBE12 - PBEll
         PBE2 = PBEll - XIA*PBE12
         PBEl = DSIGN(PBE1,SIGNRN)
         PBE2 = DSIGN(PBE2,SIGNRN)
         PNBEl = XIB*PNBE22 - PNBE21
         PNBE2 = PNBE21 - XIA*PNBE22
         GO TO 300
```

```
C
C          Determination of Coefficients for Elements where Field
C          Point Belongs to the Same Element as the Collocation Point
C          Singular Integrals
C
 100        ABC = DABS(XIB)
           PNBE21 = (1.0D+00/ABC/SS - BESK1(ABC*SS))*ABC/SS/DIFF
           PNBE22 = aIntK(ABC*SS)/SS/DIFF
           PBEl = 0.0D0
           PBE2 = 0.0D0
           PNBEl = XIB*PNBE22 - PNBE21
           PNBE2 = PNBE21 - XIA*PNBE22
           GO TO 300
 200        ABC = DABS(XIA)
           PNBE21 = (1.0D+00/ABC/SS - BESK1(ABC*SS))*ABC/SS/DIFF
           PNBE22 = aIntK(ABC*SS)/SS/DIFF
           PBEl = 0.0D0
           PBE2 = 0.0D0
           PNBEl = XIB*PNBE22 - PNBE21
           PNBE2 = PNBE21 - XIA*PNBE22
C
 300        CONTINUE
           COEFP(I,J) = COEFP(I,J) + PBEl
           COEFPL(I,J) = PNBEl
           IF (ICB .EQ. 1) THEN
             IF (J .EQ. NBEGIN-1) THEN
               COEFP(I,1) = COEFP(I,1) + PBE2
               COEFPA(I,1) = PNBE2
               GO TO 400
             ENDIF
           ENDIF
               COEFP(I,J+1) = PBE2
               COEFPA(I,J+1) = PNBE2
 400        CONTINUE
              IF (ICB .EQ. 1) THEN
                COEFP(I,NBEGIN) = COEFP(I,NBEGIN) + PBE2
                COEFPA(I,NBEGIN) = COEFPA(I,L)
              ELSE
                COEFP(I,1) = COEFP(I,1) + PBE2
                COEFPA(I,1) = COEFPA(I,L)
              ENDIF
C
C       Contribution to the coefficient of Pressure due to Fraction
C       of Circular Angle Subtended by the System Geometry at Every
C       Node Point.
C
      IF (ICB .EQ. 1) THEN
        IF (I .EQ. 1) THEN
```

```
         ALPHA(I) = ANGLE(XD(NBEGIN-1),YD(NBEGIN-1),XD(I),YD(I),
     &    XD(I+1),YD(I+1))
        ELSE
         IF (I .EQ. NBEGIN-1) THEN
          ALPHA(I) = ANGLE(XD(I-1),YD(I-1),XD(I),YD(I),XD(1),YD(1))
         ELSE
          IF (I .EQ. NBEGIN) THEN
           ALPHA(I) = ANGLE(XD(N),YD(N),XD(I),YD(I),XD(I+1),YD(I+1))
          ELSE
           IF (I .EQ. N) THEN
            ALPHA(I) = ANGLE(XD(I-1),YD(I-1),XD(I),YD(I),XD(L),YD(L))
           ELSE
            ALPHA(I) = ANGLE(XD(I-1),YD(I-1),XD(I),YD(I),XD(I+1),
     &       YD(I+1))
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       GO TO 69
       ENDIF

         IF (I .EQ. 1) THEN
          ALPHA(I) = ANGLE(XD(N),YD(N),XD(I),YD(I),XD(I+1),YD(I+1))
         ELSE
          IF (I .EQ. N) THEN
           ALPHA(I) = ANGLE(XD(I-1),YD(I-1),XD(I),YD(I),XD(L),YD(L))
          ELSE
           ALPHA(I) = ANGLE(XD(I-1),YD(I-1),XD(I),YD(I),XD(I+1),
     &      YD(I+1))
          ENDIF
         ENDIF
 69       COEFP(I,I) = COEFP(I,I) - ALPHA(I)
 500    CONTINUE
C
C        Print Out Diagnostics if KEY=1.
C
          IF (KEY .NE.1) GO TO 1000
       OPEN(UNIT=12,FILE='coefp.diag')
       REWIND 12
       WRITE(12,9000) 'COEFP'
         DO 600 I=1,N
         WRITE(12,9100)(COEFP(I,J), J=1,N)
 600    CONTINUE
       WRITE(12,9000) 'COEFPL'
          DO 700 I=1,N
          WRITE(12,9100) (COEFPL(I,J), J=1,N)
 700      CONTINUE
       WRITE(12,9000) 'COEFPA'
```

```
            DO 800 I=1,N
            WRITE(12,9100) (COEFPA(I,J), J= 1,N)
 800         CONTINUE
 9000 FORMAT(1H,A6)
 9100 FORMAT(1X,10F15.10)
 1000 RETURN
      END


      SUBROUTINE ASSEMBLE(N,ID,PDNS,PNLS,PNAS,ALPHA,BETA,CONS,CHID,H,
     &G,KEY)
C
C-----------------------------------------------------------------------
C
C     Assembles the Coefficient Matrix and the Right Hand Vector.
C     Takes the Coefficients Generated by the Routing AINTEG and
C     Depending on the Identification of the Node Allicates it to
C     the Matrix of Coefficients or the RHS Vector
C
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ID(N),PDNS(N),PNLS(N),PNAS(N),ALPHA(N),BETA(N),CONS(N),
     &CHID(N),H(100,100),G(100)
      COMMON/SINTASS/ COEFP(100,100),COEFPL(100,100),COEFPA(100,100)
C
C       CLEARING THE SUBSCRIPTED VARIABLES
C
        DO 80 I=1,N
          DO 70 J=1,N
 70         H(I,J) = 0.0D0
 80       G(I) = 0.0d+00
C
C       ALLOCATE COEFFICIENTS ACCORDING TO THE NODE IDENTIFICATION
C
            DO 200 I = 1,N
              DO 100 J = 1,N
              GO TO (1,2,3,4,5,6,7,7), ID(J)
 1            H(I,J) = COEFPA(I,J)
              G(I) = G(I) - COEFPL(I,J)*PNLS(J) - COEFP(I,J)*PDNS(J)
              GO TO 100
 2            H(I,J) = COEFPL(I,J)
              G(I) = G(I) - COEFPA(I,J)*PNAS(J) - COEFP(I,J)*PDNS(J)
              GO TO 100
 3            H(I,J) = COEFPL(I,J) + COEFPA(I,J)
              G(I) = G(I) - COEFP(I,J)*PDNS(J)
              GO TO 100
 4            H(I,J) = COEFP(I,J)
```

```
              G(I) = G(I) - COEFPA(I,J)*PNAS(J) - COEFPL(I,J)*PNLS(J)
              GO TO 100
5             H(I,J) = COEFP(I,J) - COEFPL(I,J)*ALPHA(J)/BETA(J)
              G(I) = G(I) - COEFPA(I,J)*PNAS(J) - COEFPL(I,J)*
   &           CONS(J)/BETA(J)
              GO TO 100
6             H(I,J) = COEFP(I,J) - COEFPA(I,J)*ALPHA(J)/BETA(J)
              G(I) = G(I) - COEFPL(I,J)*PNLS(J) - COEFPA(I,J)*
   &           CONS(J)/BETA(J)
              GO TO 100
7             H(I,J) = COEFP(I,J) - COEFPL(I,J)/ALPHA(J) -
   &           COEFPA(I,J)/BETA(J)
              G(I) = G(I) - COEFPL(I,J)*CHID(J)/ALPHA(J) -
   &           COEFPA(I,J)*CONS(J)/BETA(J)
C
C
 100           CONTINUE
 200         CONTINUE
C
C    Print Diagnostic Data if Needed
C
          IF (KEY .EQ. 1)'THEN
          OPEN(UNIT=13 ,FILE='mat.diag')
          REWIND 13
          WRITE(13,9500)
 9500      FORMAT(/1X,'LHS MATRIX,H(I,J)')
             DO 300 1=1,N
 300          WRITE(13,9600) (H(I,J), J=1,N)
 9600         FORMAT(8D13.5)
          WRITE(13,9700)
 9700      FORMAT(/1X,'RHS VECTOR,G(I)')
          WRITE(13,9800) (G(I), I=1,N)
 9800      FORMAT(5D13.5)
          ENDIF
       RETURN
       END


       SUBROUTINE SOURCE(N,NW,S,TD,AREA,QW,TPD,NFLOW,XD,YD,XWD,YWD)
c
c---------------------------------------------------------------------
c     Adjustment of Boundary Conditions for the Presence of Line
c     Sources and Sinks within the Problem Domain.
c
c---------------------------------------------------------------------
c
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON/SSOURCE/ AINHOM(100)
```

```
      DIMENSION XD(N),YD(N),QW(20,10),TPD(20,10),NFLOW(NW),
     &XWD(NW),YWD(NW)
      EXTERNAL BESKO

        DO 10 I = 1,N
        AINHOM(I) = 0.0d+00
10      CONTINUE

       DO 100 I=1,N
        DO 50 J = 1,NW
         dist = DSQRT((XD(I)-XWD(J))**2 + (YD(I)-YWD(J))**2)
         DO 30 K=1,NFLOW(J)
          IF (TD .GE. TPD(J,K)) THEN
           IF (K .EQ. 1) THEN
            COEF=QW(J,K)*DEXP(-S*TPD(J,K))/s
           ELSE
            COEF=(QW(J,K)-QW(J,K-1))*DEXP(-S*TPD(J,K))/S
           ENDIF
          ELSE
           COEF=0.0
          ENDIF
          AINHOM(I) = AINHOM(I) - COEF*BESK0(DIST*DSQRT(S))
30        CONTINUE
50       CONTINUE
100     CONTINUE
         RETURN
         END


       SUBROUTINE WELL(NW,XWD,YWD,QW,TPD,NFLOW,TD,XXD,YYD,AR,S,FORCE)
c
c----------------------------------------------------------------------
c      Contribution of Sources and Sinks to the Interior Point
c
c----------------------------------------------------------------------
c
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION QW(20,10),TPD(20,10),NFLOW(NW),XWD(NW),YWD(NW)
       EXTERNAL BESKO

       FORCE = 0.0d+00

         DO 50 J = 1,NW
          DIST = DSQRT((XXD-XWD(J))**2 + (YYD-YWD(J))**2)
          DO 30 K=1,NFLOW(J)
           IF (TD .GE. TPD(J,K)) THEN
            IF (K .EQ. 1) THEN
             COEF=QW(J,K)*DEXP(-S*TPD(J,K))/S
         ELSE
           COEF=(QW(J,K)-QW(J,K-1))*DEXP(-S*TPD(J,K))/S
         ENDIF
        ELSE
         COEF=0.0
        ENDIF
        FORCE = FORCE + COEF*BESK)(DIST*DSQRT(S))
 30    CONTINUE
 50   CONTINUE
      RETURN
      END


      SUBROUTINE AINTERN(N,XD,YD,ID,INT,XXD,YYD,NV  XWD  'WD,QW,TPD,NFL  ,W
     &T,G,IGANPT,PP,PDNS,PNLS,PNAS,AR,ICB,NBEGIN)
C
c---------------------------------------------------------------------
c    This Routine Calculates the Interior Solutions. It is called
c     by Stehfest.f (Steh) at every Laplace Space Sampling Point.
c     Interior Solution is Evaluated and Referred Back to
c     Steh for Complete Solution.
c
c--------------------------SUBROUTINES REFERRED-----------------------
c     Gauss.npt.f
c     Besk0.f
c     Besk1.f
c     Int_source.f
c---------------------------------------------------------------------
c
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XD(100),YD(100),ID(100),XXD(10),YYD(10),XWD(NW),YWD(NW),
     &QW(20,10),TPD(20,10),NFLOW(NW),G(100),PP(10),PDNS(100),
     &PNLS(100),PNAS(100)
      COMMON/FUNC/ETA,S
      EXTERNAL BESK0,BESK1,AINTK,XIK1,XK1,XIK0,XK0
C
c     Clearing the Subscripted Variables
C
      PI = 2.0d+00*DACOS(0.0d+00)
C
c     Initialize the Internal Solution Variable, which may Consist Values
c     from the last Laplace Space Sampling Level
C
      DO 73 1=1,INT
        PP(I) = 0.0d+00
  73  CONTINUE
      L = N + 1
      IF (ICB .EQ. 1) THEN
        XD(L) = XD(NBEGIN)
        Y D Q = YD(NBEGIN)
      ELSE
        XD(L) = XD(1)
        YD(L) = YD(1)
      ENDIF
C
C     Take the Current Solution Vector of Steh and Allocate to Proper
C     Unknowns and Use it for Evaluation at Interior Points
C
        DO 19 K=1,N
        GO TO (1,2,3,4), ID(K)
 1          PNAS(K) = G(K)
            GO TO 19
 2              PNLS(K) = G(K)
                GO TO 19
 3                  PNLS(K) = G(K)
                    PNLS(K) = PNLS(K)
                    GO TO 19
 4                      PDNS(K) = G(K)
 19       CONTINUE
C
C     If Internal Boundary Present, Proper Switching for Boundary
C     Conditions Necessary
C
      IF (ICB .EQ. 1) THEN
        PDNS(L) = PDNS(NBEGIN)
        P N A S Q = PNAS(NBEGIN)
      ELSE
       PDNS(L) = PDNS(1)
       P D A S Q = PDAS(1)
      ENDIF
       SS = DSQRT(S)
C
C     Source/Sink Contribution Accounted for
C
     do 9000 1=1,INT
        CALL WELL(NW,XWD,YWD,QW,TPD,NFLOW,T,XXD(I),YYD(I),AR,S,FORCE)

        DO 7000 J=1,N
C
C     Compiling the Coefficients of Pressures and Derivatives for Each
C     Collocation Point
C
C             Switching to a Local Co-ordinate System
C
         IF (ICB .EQ. 1) THEN
       IF (J .EQ. NBEGIN-1) THEN
         R = DSQRT((XD(1) - XD(J))**2 + (YD(1) - YD(J))**2)
         CO = (XD(1) - XD(J))/R
         SI = (YD(1) - YD(J))/R
         XIA = (XD(J) - XXD(I))*CO + (YD(J) - YYD(I))*SI
         XIB = (XD(1) - XXD(I))*CO + (YD(1) - YYD(I))*SI
         SIGNRN = -(XD(J) - XXD(I))*(YD(1) - YD(J)) +
     &          (XD(1) - XD(J))*(YD(J) - YYD(I))
         GO TO 67
       ENDIF
      ENDIF

       R = DSQRT((XD(J+1) - XD(J))**2 + (YD(J+1) - YD(J))**2)
       CO = (XD(J+1) - XD(J))/R
       SI = (YD(J+1) - YD(J))/R
       XIA = (XD(J) - XXD(I))*CO + (YD(J) - YYD(I))*SI
       XIB = (XD(J+1) - XXD(I))*CO + (YD(J+1) - YYD(I))*SI
       SIGNRN = -(XD(J) - XXD(I))*(YD(J+1) - YD(J)) +
     &   (XD(J+1) - XD(J))*(YD(J) - YYD(I))

 67      ETA = (YD(J) - YYD(I))*CO - (XD(J) - XXD(I))*SI
       DIFF = XIB - XIA
            IF (SIGNRN .LT. 0.) ETA = -ETA
       ASQ = XIA*XIA + ETA*ETA
       BSQ = XIB*XIB + ETA*ETA
C
C        Integrals and Function Evaluations
C
       XIK1SUM = (BESK0(SS*DSQRT(ASQ)) - BESK0(SS*DSQRT(BSQ)))/SS
       CALL GAUSS(IGANPT,XK1,1,XIA,XIB,2,XK1SUM,IND)
       CALL GAUSS(IGANPT,XIK0,1,XIA,XIB,2,XIK0SUM,IND)
       CALL GAUSS(IGANPT,XK0,1,XIA,XIB,2,XK0SUM,IND)
C
       PBE11 = ETA*SS*XIK1SUM/DIFF
       PBE12 = ETA*SS*XK1SUM/DIFF
       PNBE21 = XIK0SUM/DIFF
       PNBE22 = XK0SUM/DIFF
       PBEl = XIB*PBE12 - PBEll
       PBE2 = PBEll - XIA*PBE12
       PNBE1 = XIB*PNBE22 - PNBE21
       PNBE2 = PNBE21 - XIA*PNBE22

       IF (ICB .EQ. 1) THEN
         IF (J .EQ. NBEGIN-1) THEN
           PP(I) = PP(I) + PBE1*PDNS(J) + PBE2*PDNS(1) +
     &      PNBEl*PNLS(J) + PNBE2*PNAS(1)
           GO TO 7000
         ENDIF
```

```
            ENDIF
            PP(I) = PP(I) + PBE1*PDNS(J) + PBE2*PDNS(J+1) +
     &       PNBE1*PNLS(J) + PNBE2*PNAS(J+1)
C
C
 7000        CONTINUE
      PP(I) = PP(I) + FORCE
 9000  CONTINUE
                    DO 57 I=1,INT
 57                  PP(I) = PP(I)/(2.*PI)
      RETURN
      END


      FUNCTION ANGLE(X0,Y0,X1,Y1,X2,Y2)
C--------------------------------------------------------------
C     Computes the Angle subtended by two adjacent elements by
C     Considering the Triangle formed by the Three Nodal Points
C--------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      PI = 3.141592653589793D0
      ANGLE = 0.0D0
      A = (X0-X2)**2 + (Y0-Y2)**2
      B = (X2-X1)**2 + (Y2-Y1)**2
      C = (X1-X0)**2 + (Y1-Y0)**2
          IF (B*C .EQ. 0.0) RETURN
      TMP  = (B+C-A)/(2.0D0*DSQRT(B*C))
          IF (TMP .GT. 1.0D0) TMP = 1.0D0
            IF (TMP .LT. -1.000) TMP = -1.0D0
      ANGLE = DACOS(TMP)
      AREA = (X2-X1)*(Y1-Y0) - (X0-X1)*(Y1-Y2)
          IF (AREA .LT. 0.0D0) ANGLE = 2.0D0*PI - ANGLE
      RETURN
      END


      FUNCTION AREA(N,X,Y,S1,S2,IBUG)

C--------------------------------------------------------------
C     'AREA' Calculates the ARea Enclosed by the Polygon Enclosed
C     by the POints S1 through S2 by Triangulation
C--------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER S1,S2
      DIMENSION X(*),Y(*)
      AREA = 0.0D0
        IF (S1 .EQ. S2) RETURN
      RS2 = S2
      IF (S2.LT. Sl) RS2 = S2 + N
        IF (N .LT. 3) THEN
        WRITE(6,*) '** ERROR IN AREA.F **'
        ENDIF
C
      DO 10 I=S1+1,RS2-1
      X1 = X(I)
      Y1 = Y(I)
        IF (I .GT. N) THEN
        X1 = X(I-N)
        Y1 = Y(I-N)
        ENDIF
      X2 = X(I+1)
      Y2 = Y(I+1)
        IF (I+1 .GT. N) THEN
        X2 = X(I+1-N)
        Y2 = Y(I+1-N)
        ENDIF
      AREA = AREA - 0.5*(X(S1)*(Y1-Y2) + X1*(Y2-Y(S1)) + X2*(Y(S1)-Y1))
 10   CONTINUE
        IF (IBUG.EQ. 1) THEN
        OPEN (UNIT = 11,FILE = 'area.diag')
        REWIND 11
        WRITE(11,*) 'CALCULATION OF CUMULATIVE AREAS OF TRIANGLES
     &  FORMED IN THE POLYGON'
        WRITE(11,*) '          CLOCKWISE NUMBERING OF NODES'
        WRITE(11'20)
        WRITE(11,30) X0,Y0,X1,Y1,X2,Y2,AREA
 20     FORMAT(/4X,'X0',4X,'Y0',6X,'X1',4X,'Y1',6X,'X2',4X,'Y2',7X,
     &  'CUM AREA')
 30     FORMAT(2X,F5.3,2X,F5.3,2X,F5.3,2X,F5.3,2X,F5.3,2X,F5.3,5X,F6.3
     &  )
        ENDIF
      RETURN
      END


      SUBROUTINE GAUSS(N,FUN,KEY,A,B,M,SUM,IND)
C
C.....................................................................
c     All purpose integration program to be used when the number of
c     integration abscissas,their values and the corresponding
c     weights are known in advance.
c     GAUSS can be used for :
c     Compound GAUSS                  I
c     NEWTON-COTES                    I        FINITE INTERVAL
c     LOBATTO                         I
c     RADAU                               I
```

```
C     LAGUERRE               I
C     GENERALIZED LAGUERRE I       SEMI-INFINITE, INFINITE
C     HERMITE                I
C
C     Parameters needed :
C     X : Array containing abscissas of an N-point integration rule
C     W : Array containing weights of an N-point integration rule
C     A,B : Endpoints for integration over a finite interval
C     M : Number of subintervals [A,B] is divided
C     Result appears in 'sum' if :
C     IND = 1
C     IND = 0 (SET) IF N<1, KEY<1, KEY>6, OR M<1  WHEN KEY<=3
C     KEY = 1   Composite symmetric rule (GAUSS), interval [-1,1],
C               non-negative abscissas and weights stored in X & W
C               to be given in ascending order of the abscissas
C               The abscissas and weights for N-point Gauss-Legendre
C               integration rule can be generated in ascending order
C               by the subroutine "GRULE".
C     KEY = 2   Composite non-symmetric rule (RADAU), interval [-1,1],
C               to be given in any order
C     KEY = 3   Composite symmetric rule (Closed NEWTON-COTES,LOBATTO)
C               includes endpoints of integration among abscissas,
C               Abscissas and weights to be given in order as KEY=1
C     KEY = 4   Symmetric rule (HERMITE), or any rule where exact
C               abscissas and weights given, interval [-INF,INF],
C               to be given for non-negative abscissas in ascending order
C     KEY = 5   LAGUERRE integration, for  A -> INF,EXP(-X)F(X)dX,
C               abscissas and weights in any order
C     KEY = 6   Any approximate integration such as generalized LAGUERRE
C               integration, where integral is approximated by
C               SUM (I=1 -> N) Wi F(Xi) and N values of Wi and Xi given
C
C     A PROGRAM FUNCTION FUN(X) MUST BE SUPPLIED BY THE USER AND
C     FUN MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM
C------------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(25),W(25)
      EXTERNAL FUN

      IND = 0
      IF (N .LT. 1 .OR. KEY .LT. 1 .OR. KEY .GT. 6) RETURN
      IF (KEY .GE. 4) IND = 1
      L = (N+1)/2
      K = 2*L-N+1
      SUM = 0.
        IF (KEY- 5) 7,5,6
 7          IF (KEY .EQ. 4) GO TO 4
```

```
              IF (M .LT. 1) RETURN
      IND = 1
      AM = M
      H1 = (B - A)/AM
      H = H1*.5
      A0 = A + H
      GO TO (1,2,3),KEY
C      KEY = 1, M*SYMMETRIC RULE, e.g.GAUSS
C
C      GENERATE THE ABSCISSAS AND WEIGHTS FOR N POINT GAUSS-LEGENDRE
C      QUADRATURE.  SUBROUTINE GRULE GENERATES NON-NEGATIVE VALUES IN
C      ASCENDING ORDER.
C
 1       CALL GRULE(N,X,W)
          DO 10 M1 = l,M
             IF (K .EQ. 2) SUM = SUM + W(1)*FUN(A0)
                DO 15 I = K,L
                H2 = H*X(I)
 15              SUM = SUM + W(I)*(FUN(A0+H2) + FUN(A0-H2))
 10        A0 = A0 + H1
 16        SUM = H*SUM
          RETURN
C      KEY =2, M*NON-SYMMETRIC RULE,.e.g.RADAU
 2          DO 20 M1 = l,M
              DO 21 I = l,N
 21            SUM = SUM + W(I)*FUN(A0+H*X(I))
 20          A0 = AO+ H1
       GO TO 16
C      KEY = 3, M*SYMMETRIC RULE WITH ENDPOINT 1,.e.g.LOBATTO
 3      SUM = W(L)*(FUN(B) - FUN(A))
       W1 = 2*W(L)
       L1 = L - 1
           DO 30 M1 = l,M
              IF (K .EQ. 2) SUM = SUM + W(1)*FUN(A0)
                  IF (K .GT. L1) GO TO 33
 32                   DO 35 I = K,L1
                     H2 = H*X(I)
 35                   SUM = SUM + W(I)*(FUN(A0+H2) + FUN(A0-H2))
 33           SUM = SUM + W1*FUN(A0-H)
 30           A0 = A0 + H1
       GO TO 16
C      KEY = 4, INFINITE SYMMETRIC RULE, .e.g.HERMITE
 4          IF (K .EQ. 2) SUM = SUM + W(1)*FUN(X(1))
              DO 40 I = K,L
 40           SUM = SUM + W(I)*(FUN(X(I)) + FUN(-X(I)))
       RETURN
C      KEY = 5, LAGUERRE INTEGRATION FROM A TO INFINITY
 5         DO 50 I = l,N
 50     SUM = SUM + W(I)*FUN(X(I) + A)
      SUM = EXP(-A)*SUM
      RETURN
C      KEY = 6, PURE INNER PRODUCT, .e.g. GENERALIZED LAGUERRE
 6      DO 60 I = l,N
 60     SUM = SUM + W(I)*FUN(X(I))
      RETURN
      END


      SUBROUTINE GRULE(N,X,W)

C------------------------------------------------------------------------
C      Computes the [(N+1)/2] Non-negative Abscissas X(I) and
C      Corresponding Weights W(I) of the N-point Gauss-Legendre
C      Integration Rule, Normalized to the Interval [-1,1].  The
C      Abscissas Appear in Ascending Order.  Ref.: Stroud and Secrest (1966)
C------------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION PKM1,PK,T1,PKP1,DEN,D1,DPN,D2PN,D3PN,D4PN,U,
     &V,H,P,DP,FX
      DIMENSION TX(25),TW(25)
      DOUBLE PRECISION X(N),W(N)
      M = (N+1)/2
      El = N*(N+1)
        DO 1 I = l,M
        T = (4*I-1)*3.1415926536/(4*N+2)
        XO = (1. -(1. -1./N)/(8.*N*N))*COS(T)
        PKMl = 1.
        PK = XO
          DO 3 K = 2,N
          T1 = X0*PK
          PKP1 = Tl - PKM1 - (T1-PKM1)/K + T1
          PKMl = PK
 3        PK = PKP1
        DEN = 1. - X0*X0
        D1 = N*(PKM1 - X0*PK)
        DPN = D1/DEN
        D2PN = (2.*X0*DPN - E1*PK)/DEN
        D3PN = (4.*X0*D2PN + (2.-E1)*DPN)/DEN
        D4PN = (6.*X0*D3PN + (6.-E1)*D2PN)/DEN
        U = PK/DPN
        V = D2PN/DPN
        H = -U*(1. + .5*U*(V + U*(V*V- D3PN/(3.*DPN))))
        P = PK + H*(DPN + .5*H*(D2PN + H/3.*(D3PN+.25*H*D4PN)))
        DP = DPN + H*(D2PN + .5*H*(D3PN+H*D4PN/3.))
        H = H - P/DP
```

```
          TX(I) =  X0 + H
          FX = D1 - H*E1*(PK + .5*H*(DPN + H/3.*(D2PN+.25*H*(D3PN+
     &     .2*H*D4PN))))
 1        TW(I) = 2.*(1. -TX(I)*TX(I))/(FX*FX)
            IF (M+M .GT. N) TX(M) = 0.
           DO 5 I=1,M
           X(M-I+1) = TX(I)
           W(M-I+1) = TW(I)
 5          CONTINUE
        RETURN
        END


        SUBROUTINE MATINV(N,A,EPS,INDIC,DTNRM)
C
C-------------------------------------------------------------------
C      The Routing Employs GAUSS-JORDAN Elimination with Maximum
C      Pivoting Strategy.
C      INDIC < 0    ------->  Inverse of N x N Matrix 'A'  is Computed
C      INDIC = 0    ------->  Solution X(1 .. N) Computed with A as
C                             Augmented(N x N+1) Array, Inverse also
C                             Computed
C      INDIC > 0    ------->  Solution Computed, Inverted Matrix Not
C                             Computed
C      EPS     If the Potential Pivot of Largest Magnitude be Smaller
C              than EPS, Matrix Considered Singular, DTNRM = 0
C      DTNRM   Condition Number of Matrix = Determinant 'A'/ Euclidean
C              NORM.
C      Euclidean Norm is the Square Root of the Sum of the Squares of
C      Each of the Elements of the Matrix.
C      Row and Column Subscripts for Successive Pivot Elements are Saved
C      in Order in the IROW and JCOL Arrays Respectively.
C      K   is the Pivot Counter, Pivot    The Algebraic Value of the
C      Pivot Element, Max    The number of Columns in a .
C      The Solutions are Computed in the N+1th Column of A and then
C      Unscrambled and Put in Proper Order in X(1)..X(N) Using the
C      Pivot Subscript Information Available in the IROW and JCOL
C      Arrays. The Sign of the Determinant is Adjusted if Necessary by
C      Determining if an Odd or Even Number of Pairwise Interchanges is
C      Required to put the Elements of the JORD Array in Ascending
C      Sequence where JORD(IROW(I)) = JCOL(I).
C      If the INverse is Required, It is Unscrambled in Place Using
C      Y(1)..Y(N) as Temporary Storage.
C
c      note:   the routine has been modified just to calculate the inverse
c      of a matrix(n x n). If needed to solve a system of eqations add
c      JORD(100),X(N) in dimension statement and X in the argument of the
c      subroutine. also remove the comment signs from statements.
```

```
C
C------------------------------- --------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IROW(100),JCOL(100),JORD(100),Y(100),A(1013,100)
C
      MAX = N
         IF (INDIC .GE. 0) MAX = N+l
C
C      IS N LARGER THAN 100?
C
            IF (N .LE. 100) GO TO 5
      WRITE(6,500)
      DTNRM = 0.
      RETURN
C
C      BEGIN ELIMINATION PROCEDURE
C
 5     DETER = 1.
      PD = 0.
          DO 100 L = l,N
             DO 100 K = 1,N
 100          PD = PD + A(L,K)*A(L,K)
          PD = DSQRT(PD)
              DO 18 K = l,N
              KM1 = K - 1
C
C      SEARCH FOR THE PIVOT ELEMENT
C
              PIVOT = 0.
                 DO 11 I = l,N
                 DO 11 J = l,N
C
C      SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS
C
                   IF ( K .EQ. 1) GO TO 9
                       DO 8 ISCAN = 1, KM1
                       DO 8 JSCAN = 1, KM1
                          IF (I .EQ. IROW(ISCAN)) GO TO 11
                             IF (J .EQ. JCOL(JSCAN)) GO TO 11
 8                      CONTINUE
 9                   IF (DABS(A(I,J)) .LE.DABS(PIVOT)) GO TO 11
                 PIVOT = A(I,J)
                 IROW(K) = I
                 JCOL(K) = J
 11               CONTINUE
C
C      INSURE THAT SELECTED PIVOT IS LARGER 'THAN EPS
```

```
C
         IF (DABS(PIVOT) .GT. EPS) GO TO 13
         DTNRM = 0.
         RETURN
C
C     UPDATE THE DETERMINANT VALUE
 13      IROWK = IROW(K)
         JCOLK = JCOL(K)
         DETER = DETER*PIVOT
C
C     NORMALIZE PIVOT ROW ELEMENTS
            DO 14 J = 1,MAX
 14         A(IROWK,J) = A(IROWK,J)/PIVOT
C
C     CARRY OUT ELIMINATION AND DEVELOP INVERSE
      A(IROWK,JCOLK) = 1./PIVOT
         DO 18 I = 1,N
         AIJCK = A(I,JCOLK)
            IF (I .EQ. IROWK) GO TO 18
         A(I,JCOLK) = -AIJCK/PIVOT
              DO 17 J = 1,MAX
 17           IF (J .NE.JCOLK) A(I,J) = A(I,J) - AIJCK*A(IROWK,J)
 18       CONTINUE
C
C     ORDER SOLUTION VALUES( IF ANY) AND CREATE JORD ARRAY
C
      DO 20 I = 1,N
      IROWI = IROW(I)
      JCOLI = JCOL(I)
 20    JORD(IROWI) = JCOLI
c 20     IF (INDIC .GE. 0)  X(JCOLI) = A(IROWI,MAX)

C     ADJUST SIGN OF DETERMINANT
      INTCH = 0
      NM1 = N - 1
        DO 22 I = 1,NM1
        IP1 = I + 1
        DO 22 J = IP1,N
           IF ( JORD(J) .GE. JORD(I))    GO TO 22
        JTEMP = JORD(J)
        JORD(J) = JORD(I)
        JORD(I) = JTEMP
        INTCH = INTCH + 1
 22     CONTINUE
          IF (INTCH/2*2 .NE.INTCH) DETER = -DETER
C
C     IF INDIC IS POSITIVE RETURN WITH RESULTS
```

```
C
         IF (INDIC .LE. 0) GO TO 26
         DTNRM = ABS(DETER)/PD
         RETURN
C
C     IF INDIC IS NEGATIVE OR ZERO, UNSCRAMBLE THE INVERSE
C
C     FIRSTBYROWS
C
 26      DO 28 J = l,N
           DO 27 I = l,N
           IROWI = IROW(I)
           JCOLI = JCOL(I)
 27         Y(JCOLI) = A(IROWI,J)
         DO 28 I = 1,N
 28       A(I,J) = Y(I)
C
C     THEN BY COLUMNS
C
            DO 30 I = l,N
               DO 29 J = l,N
               IROWJ = IROW(J)
               JCOLJ = JCOL(J)
 29             Y(IROWJ) = A(I,JCOLJ)
            DO 30 J = l,N
 30          A(I,J) = Y(J)
C
C      RETURN FOR INDIC NEGATIVE OR ZERO
      DTNRM = ABS(DETER)/PD
      RETURN
 500  FORMAT(10H0N TOO BIG )
C
      END


       SUBROUTINE MATMULT(N,A,B)
C
C-------------------------------------------------------------------
C      EVALUATES THE PRODUCT OF TWO NON-ZERO VECTORS A AND B
C      B IS OVERWRITTEN WITH THE VECTOR OF THE PRODUCTS.
C
C-------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(100,*),B(N),C(100)
C
          DO 30 I=1,N
          C(I) = 0.0D0
```

```
           DO 20 J=1,N
 20         C(I) = C(I) + A(I,J)*B(J)
 30      CONTINUE
           DO 50 J=1,N
 50         B(J) = C(J)
      RETURN
      END


      SUBROUTINE SCGEN(N)
c------------------------------------------------------------
c
c     This Routine Generates the Coefficients for the Stehfest
c     Algorithm for Inversion of Laplace Transforms.
c
c------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION G(0:50),H(25)
      COMMON/coefgen/ V(50)

    1 G(0) = 1.W
      NH   = N/2
      SN   = 2*MOD(NH,2)-1
      DO 2 I= 1,N
         G(I) = G(I-1)*I
    2 CONTINUE
      H(1) = 2.D0/G(NH-1)
      DO 3 I= 2,NH
         FI   = I
         H(I) = FI**NH*G(2*I)/G(NH-I)/G(I)/G(I-1)
    3 CONTINUE
      DO 5 I= 1,N
         V(I) = 0.D0
         KBG  = (I+1)/2
         KND  = MIN0(I,NH)
         DO 4 K= KBG,KND
            V(I) = V(I)+H(K)/G(I-K)/G(2*K-I)
    4    CONTINUE
         V(I) = SN*V(I)
    5    SN   = -SN
      RETURN
      END


      DOUBLE PRECISION FUNCTION XK0(X)
C
c     Evaluates K0(r*dsqrt(s)) for the integration of this
c     function by the gaussian quadrature over a finite range
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FUNC/ETA,S
      EXTERNAL BESKO

      SS = DSQRT(S)
      R = DSQRT(X*X + ETA*ETA)
      TERM = BESK0(R*SS)
      XKO = TERM
      RETURN
      END

      DOUBLE PRECISION FUNCI'ION XIK0(X)
c
c     Evaluates x*K0(r*dsqrt(s)) for the integration of this
c     function by the gaussian quadrature over a finite range
c
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FUNC/ETA,S
      EXTERNAL, BESKO

      SS = DSQRT(S)
      R = DSQRT(X*X + ETA*ETA)
      TERM = X*BESK0(R*SS)
      XIKO = TERM
      RETURN
      END

      DOUBLE PRECISION FUNCTION XK1(X)
c
c     Evaluates K1(r*dsqrt(s))/r for the integration of this
c     function by the gaussian quadrature over a finite range
c
      IMPLICIT REAL*8(A-H,0-Z)
      COMMON/FUNC/ETA,S
      EXTERNAL BESK1

      SS = DSQRT(S)
      R = DSQRT(X*X + ETA*ETA)
      TERM = BESK1(R*SS)/R
      XK1 =TERM
      RETURN
      END

      DOUBLE PRECISION FUNCTION XIKl (X)
C
C     Evaluates x*K1(r*dsqrt(s))/r for the integration of this
C     function by the gaussian quadrature over a finite range
```

```
c
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/FUNC/ETA,S
      EXTERNAL BESKl

      SS = DSQRT(S)
      R = DSQRT(X*X + ETA*ETA)
      TERM = X*BESK1(R*SS)/R
      XIKl = TERM
      RETURN
      END

        DOUBLE PRECISION FUNCTION AINTK(X)

c
c       Evaluates the integral of KO over the singular interval
c       0 --> x  by means of ascending series upto x=11.2 and
c       above that by means of asymptotic series.
c               Abramowitz and Stegun Pg. 480-81
c
c
        IMPLICIT REAL*8(A-H,O-Z)
          AINTK=0.0D+00
          PI=3.14159265358979323846264 3D+00
          GAMMA=0.57721566490153D+00

        I=1
        Y=1.0d+00

          IF(X.GE.12.0D+00) GOTO 100

        AHARMA0.0d+00
        AINTK=X - X*GAMMA + X*DLOG(2.0D+00/X)
 1      AHARM=AHARM+ 1.0D+00/Y
        AKIT = 2.0d+00*(D/2.0D+00)**(2*I+1)/FACT(Y)/FACT(Y)
     &          /(2.0D+00*Y+ 1.0D+00)*(DLOG(2.0D+00/X)-GAMMA
     &          +1.0D+00/(2.0D+00*Y+ 1.0d+00) + AHARM)
        AINTK=AINTK + AKIT
          IF (AINTK/AKIT.LE. 1.OD+16) THEN
        Y=Y+1.0D+00
        I=I+1
        GOTO 1
        END
        RETURN

 100      IF (X .ge. 50.D+00) GO TO 200
        AMIN=1.0D+00
        AZER=0.625D+00
      K=2
      Y=1.0D+00
      ASSYM=DSQRT(PI/
     &        (2.0D+00*X))*DEXP(-X)
      AINKIT=ASSYM*(AMIN - AZER/X)

 3     APLUS=(1.5D+00*AZER*(Y+0.5D+00)*(Y+5.0D 00/6.0D+00)
     &     -0.5D+00*AMIN*(Y+0.5D+00)**2*(Y-0.5D+00))/(Y+1.0D+00)

      AKIT=(-1.0D+00)**K*APLUS/(X**K)*ASSYM
        IF(DABS(PI/(2.0D+00*AKIT)).LE.1.OD+10) THEN
          AINKIT=AINKIT + AKIT
          AMIN=AZER
          AZER=APLUS
          Y=Y+1.0D+00
          K=K+1
          GOTO 3
        ENDIF

      AINKIT=AINKIT + AKIT
      AINTK=PI/2.0D+00 - AINKIT
      RETURN
 200    AINTK=PI/2.0D+00
      RETURN
      END

      DOUBLE PRECISION FUNCTION FACT(W)
       IMPLICIT REAL*8(A-H,O-Z)
C
      IF (W.GT.56) THEN
        print*,'Factorial Argument Too High', w
      ENDIF
      FACT=1.0D+00
      X=0.0D+00
      IF(W.LE.1.0D-04) RETURN
 1     X=X+1.0D+00
      FACT=FACT*X
      IF(DABS(X-W).GE.1.0D-04) GOTO 1
      RETURN
      END
```