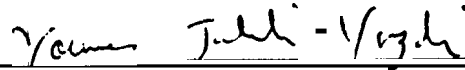


**RATE DECLINE ANALYSIS FOR
NATURALLY FRACTURED RESERVOIRS**

**A REPORT
SUBMITTED TO THE DEPARTMENT OF PETROLEUM
ENGINEERING
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE**

**By
Katsunori Fujiwara
June 1989**

I certify that I have read this report and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Petroleum Engineering.



Younes Jalali-Yazdi
(Principal advisor)

Acknowledgement

I would like to thank Dr. Younes Jalali-Yazdi for his essential help and his able guidance as my research advisor during the course of this research work and the preparation of this report.

I also would like to thank the faculty, students, and staff of the Department of Petroleum Engineering for providing me with encouragement and good basis for research work.

I would like to dedicate the present study to my wife, Rinko, for providing me patience and happiness.

Finally, I would like to thank Nippon Mining Co., Ltd. and Japan National Oil Corporation for having provided the financial support for my Master's program at Stanford University.

Abstract

In this work, transient rate analysis for constant pressure production in a naturally fractured reservoir is presented. The solution for the dimensionless flowrate is based on a model which treats interporosity flow as a function of a continuous matrix block size distribution. Several distributions of matrix block size are considered. This approach is similar to that of Ref. 1, which examined the pressure response.

The flowrate response is investigated for both pseudo-steady state (**PSS**) and unsteady state (**USS**) interporosity models, which include slab, cylindrical, and spherical matrix block geometries. It was found that the flowrate decline becomes smooth, specially for the unsteady state model, and approaches the decline behavior of a nonfractured reservoir when matrix block size Variability is large, i.e., when fracturing is extremely nonuniform. The difference in flowrate for various geometric models of blocks is not significant, with the spherical geometry yielding the highest and the slab yielding the lowest flowrate.

This work suggests why certain naturally fractured reservoirs do not exhibit a sudden rate decline followed by a period of constant flowrate as predicted by classical double porosity models. Also, the results indicate that reservoir producibility is directly proportional to fracture intensity and inversely proportional to the degree of fracture nonuniformity. Hence, the Warren and Root model which assumes fracturing is perfectly uniform, provides an upper bound of reservoir producibility and cumulative production.

Contents

Acknowledgement	iii
Abstract	iv
Table of Contents	vi
List of Tables	vii
List of Figures	ix
1 Introduction	1
2 Mathematical Model	4
2.1 Initial Boundary Value Problem	4
2.2 Probability Density Functions (PDF)	6
2.3 Solutions	8
2.3.1 General Solution	8
2.3.2 Solutions for Slab Geometry	8
2.3.3 Solutions for Cylindrical Geometry	11
2.3.4 Solutions for Spherical Geometry	11
2.3.5 Dimensionless Parameters	12
3 Discussion	15
3.1 Effect of Matrix Block Geometry	15
3.2 Effect of the Mode of Interporosity Flow	26

3.3	Effect of PDF Type	29
3.4	Effect of Fracture Intensity and Uniformity	34
4	Conclusion	39
	Nomenclature	41
	Bibliography	45
A	Derivation of Solution	48
A.1	Slab Geometry	48
A.1.1	Unsteady State	48
A.1.2	Pseudo-Steady State	51
A.2	Cylindrical Geometry	53
A.2.1	Unsteady State	53
A.2.2	Pseudo-Steady State	54
A.3	Spherical Geometry	55
A.3.1	Unsteady State	55
A.3.2	Pseudo-Steady State	57
B	Computer Programs	59

List of Tables

2.1 Constant α for various matrix geometries	13
---------------------------------------------------------------	----

List of Figures

1.1	Naturally Fractured Reservoir	3
2.1	Uniform Distribution	6
2.2	Positively Skewed Linear Distribution	7
2.3	Negatively Skewed Linear Distribution	7
2.4	Matrix Slab Geometry	13
2.5	Matrix Cylindrical Geometry	14
2.6	Matrix Spherical Geometry	14
3.1	Flowrate profile for slab geometry in USS model	17
3.2	Flowrate profile for cylindrical geometry in USS model	18
3.3	Flowrate profile for spherical geometry in USS model	19
3.4	Comparison of flowrate tesponse of various matrix block geometries .	20
3.5	Difference in flowrate of various matrix block geometries	21
3.6	Pressure and pressure derivative profile for slab geometry in USS model	22
3.7	Pressure and pressure derivative profile for cylindrical geometry in USS model	23
3.8	Pressure and pressure derivative profile for spherical geometry in USS model	24
3.9	Comparison of pressure derivative profile of various matrix block ge- ometries	25
3.10	Comparison of PSS and USS flowrate response	27
3.11	Comparison of PSS and USS pressure derivative profiles	28

3.12	PSS flowrate profile for various matrix block size distributions	30
3.13	PSS pressure and pressure derivative profile for various matrix block size distributions	31
3.14	USS flowrate profile for various matrix block size distributions . . .	32
3.15	USS pressure and pressure derivative profile for various matrix block size distributions	33
3.16	Cumulative production at $t_D = 10^5$ for PSS model	35
3.17	Cumulative production at $t_D = 10^5$ for USS model	36
3.18	Flowrate profile for various λ_{gmean}	37
3.19	Pressure derivative profile for various λ_{gmean}	38

Section 1

Introduction

Much work has been done on the pressure transient modeling of naturally fractured reservoirs. However, the rate response and producing capacity of these reservoirs have not received adequate attention. This work examines flowrate decline behavior of naturally fractured reservoirs.

Naturally fractured reservoirs are heterogeneous porous media which consist of fractures and matrix blocks. The matrix blocks store most of the fluid, but have low permeability. On the other hand, the fractures do not store much, but have extremely high permeability. Most of the reservoir fluid flows from the matrix blocks into the wellbore through the permeable fractures. Therefore, the producing capacity of a naturally fractured reservoir is governed by matrix-fracture fluid transport capacity, which is called interporosity flow. To describe flow in naturally fractured reservoirs, double porosity model has been widely used. Fig. 1.1 shows the schematic of a naturally fractured reservoir and its double porosity idealization. This concept was first proposed by Barenblatt *et al* [2,3]. Transient pressure behavior for this model has been studied by many researchers [2-16]. Mavor and Cinco-Ley [10] and Da Prat *et al* [17] examined the rate response of this model by applying the rate decline concept proposed by Fetkovich [18]. Raghavan and Ohaeri [19], and Sageev [20] also examined the rate decline behavior of naturally fractured reservoirs. These works indicate that the double porosity model predicts an initial high flowrate followed by a sudden rate decline and a period of constant

flowrate (the interporosity flow period). Although many naturally fractured reservoirs exhibit such behavior, many others do not. They exhibit a gradual rate decline throughout the life of the reservoir, similar to that of nonfractured reservoirs. Jalali-Yazdi *et al* [21] suggested the concept of a distributed interporosity flow strength to explain such behavior.

Here, a similar approach is adopted, and the effect of the variation of matrix block size on flowrate is investigated. This work demonstrates that the gradual rate decline occurs in nonuniformly fractured reservoirs where the variation in matrix block size is large. Also, this work shows that fracture nonuniformity has an adverse effect on the producing capacity of naturally fractured reservoirs.

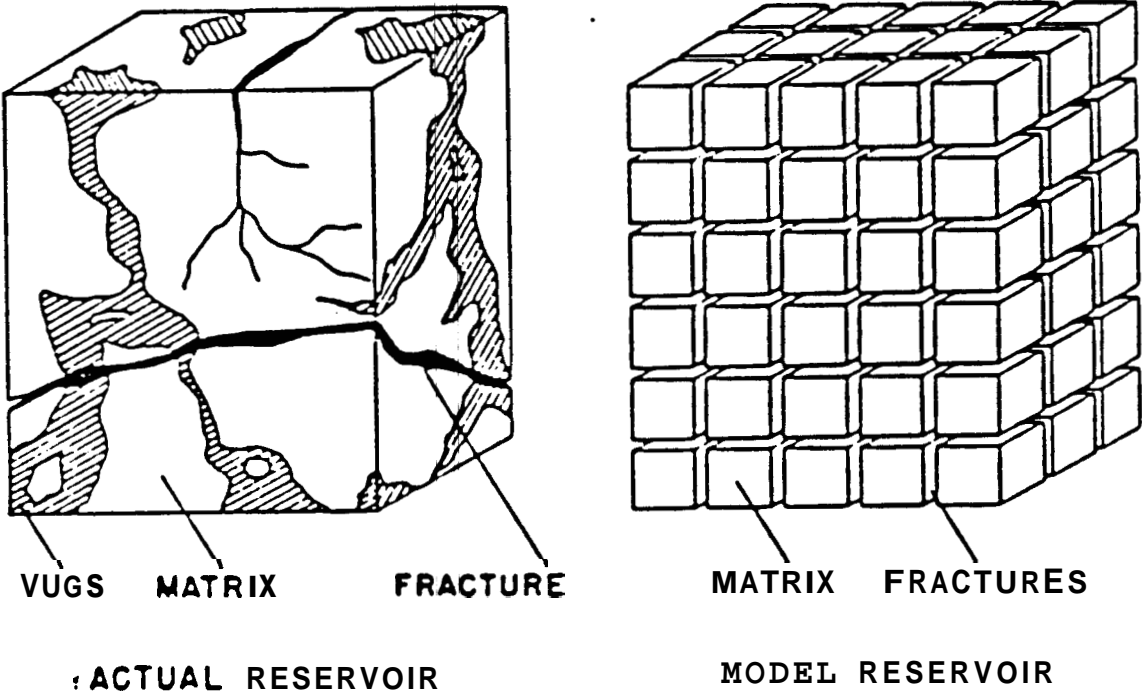


Figure 1.1: Naturally Fractured Reservoir (after Ref. 4)

Section 2

Mathematical Model

The partial differential equations and their solutions for pseudo-steady state and three unsteady state cases, which include slab, cylindrical, and spherical matrix block geometries, are presented in this section. Also, the probability density functions (PDF) used in this study to represent various distributions of matrix block size, are presented.

2.1 Initial Boundary Value Problem

The fracture diffusivity equation for a double porosity reservoir with a continuous matrix block size distribution is given by:

$$\nabla^2 P_f = \frac{\phi_f c_f \mu}{k_f} \frac{\partial P_f}{\partial t} + \frac{\mu}{k_f} \int_{h_{min}}^{h_{max}} U(h) P(h) dh, \quad (2.1)$$

and the matrix diffusivity equation is:

$$\nabla^2 P_m = \frac{\phi_m c_m \mu}{k_m} \frac{\partial P_m}{\partial t}. \quad (2.2)$$

In Eqn. 2.1, $P(h)$ is the probability density function of matrix block size and is discussed in the next section. $U(h)$ in Eqn. 2.1 is the flow contribution from a matrix block of size h , and is given by:

$$U(h) = -\frac{k_m A}{\mu V} \nabla \bar{P}_m|_{interface}, \quad (2.3)$$

where A/V is the specific surface area of the matrix block.

The main assumptions used to develop the equations and solutions are as follows:

- o flow is single phase and obeys Darcy's law,
- o reservoir fluid is slightly compressible,
- o reservoir is radial and infinite in extent ,
- matrix and fracture properties are homogeneous and isotropic, and
- o the well is either producing at constant pressure (rate decline) or constant rate (pressure decline).

For a reservoir at constant pressure, the initial condition is:

$$P_f = P_m = P_i ; \text{ at } t = 0. \quad (2.4)$$

The radial inner boundary condition for wellbore storage and skin are:

$$q = C \left(\frac{\partial P_w}{\partial t} \right) - \frac{2\pi k_f h}{\mu} \left(r \frac{\partial P_f}{\partial r} \right)_{r=r_w}, \quad (2.5)$$

$$P_w = \left[P_f - S_D \left(r \frac{\partial P_f}{\partial r} \right) \right]_{r=r_w}. \quad (2.6)$$

The radial outer boundary condition for an infinite acting reservoir is:

$$P_f = P_i ; \text{ as } r \rightarrow \infty. \quad (2.7)$$

Two boundary conditions are needed for flow in matrix blocks. One specifies that $P_m = P_f$ at the matrix-fracture interface. The other, depending on the geometry, specifies either no-flow boundary or bounded pressure at the center of the matrix block (see Appendix A for details).

2.2 Probability Density Functions (PDF)

Three probability density functions for the matrix block size distribution are considered — uniform, positively skewed linear, and negatively skewed linear distributions. Fig. 2.1 shows the schematic of a uniform distribution. The normalized probability density function for uniform distribution is given by:

$$P(h_D) = \frac{1}{1 - \frac{h_{min}}{h_{max}}} \quad (2.8)$$

Fig. 2.2 depicts a positively skewed linear distribution, where the normalized probability density function is given by:

$$P(h_D) = \frac{2(h_D - \frac{h_{min}}{h_{max}})}{(1 - \frac{h_{min}}{h_{max}})^2} \quad (2.9)$$

The negatively skewed linear distribution is shown in Fig. 2.3, and the normalized probability density function is given by:

$$P(h_D) = \frac{2(1 - h_D)}{(1 - \frac{h_{min}}{h_{max}})^2} \quad (2.10)$$

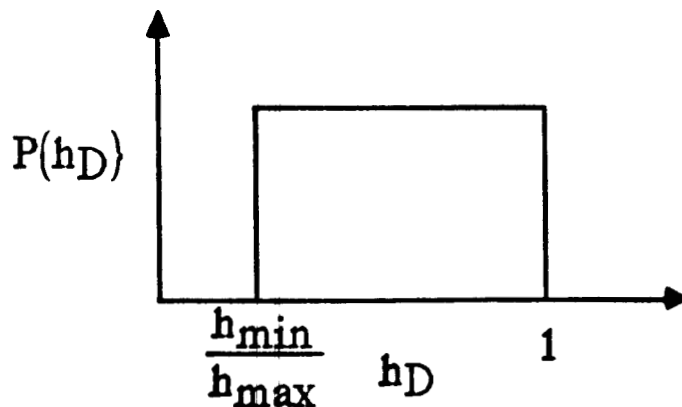


Figure 2.1: Uniform Distribution

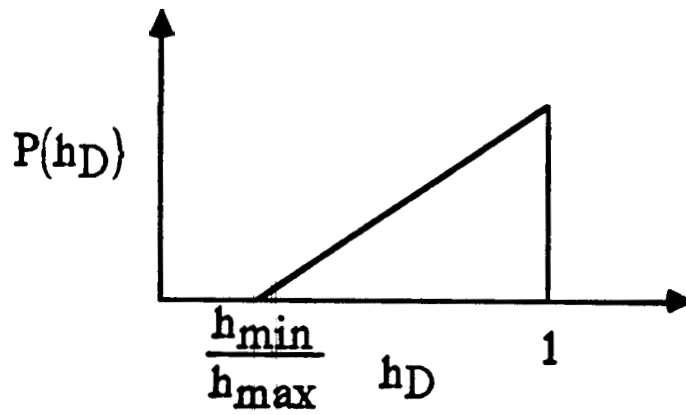


Figure 2.2: Positively Skewed Linear Distribution

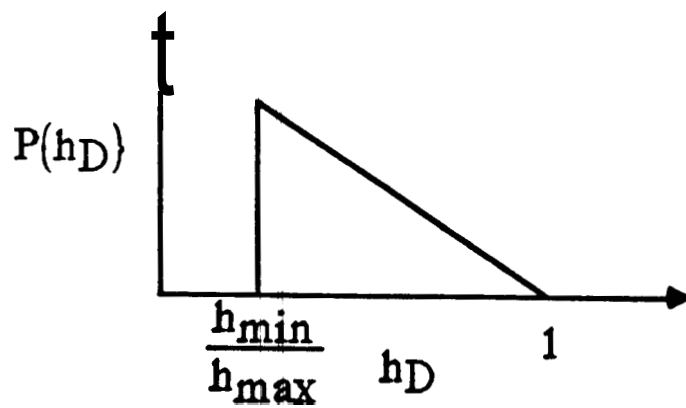


Figure 2.3: Negatively Skewed Linear Distribution

2.3 Solutions

2.3.1 General Solution

The initial boundary value problem is rendered dimensionless and solved using Laplace transforms. The procedure is stated in Appendix A. The constant pressure solution is obtained **from** the constant rate solution (with $C_D = 0$) using the result presented by van Everdingen and Hurst [22]:

$$\bar{q}_D = \frac{1}{s^2 \bar{P}_{Dw}}, \quad (2.11)$$

where:

$$\bar{P}_{Dw} = \frac{K_0(x) + S_D x K_1(x)}{s x K_1(x)}. \quad (2.12)$$

The cumulative production is obtained by integrating the flowrate over time:

$$Q_D = \int_0^{t_{Dp}} q_D dt_D. \quad (2.13)$$

Or, in Laplace space:

$$\bar{Q}_D = \frac{\bar{q}_D}{s}. \quad (2.14)$$

The argument x of modified **Bessel** functions is:

$$x = \sqrt{s g(s)}. \quad (2.15)$$

The function $g(s)$ can be obtained by specifying the mode of interporosity flow (**PSS** or **USS**), the geometry of matrix blocks (slab, cylinder, or sphere), and the distribution of matrix block size.

2.3.2 Solutions for Slab Geometry

Fig. 2.4 shows the schematic of the slab geometry. In this geometry, **matrix** blocks and fractures are assumed to be piled up on one another. No **flow** boundary exists at the center of the matrix slab due to flow symmetry. **Pseudo-steady** state and unsteady state **cases** are considered for the three probability density functions discussed in section 2.2.

Pseudo-Steady State Solutions

o Arbitrary Distribution

Let $P(h_D)$ denote any probability density function of matrix block size. From Appendix A, we have:

$$g(s) = \omega_f + \omega_m \int_{h_{min}}^1 \frac{\lambda}{\omega_m s + \lambda} P(h_D) dh_D. \quad (2.16)$$

o Uniform Distribution

By combining Eqns. 2.16 and 2.8, one obtains:

$$g(s) = \omega_f + \omega_m \frac{\sqrt{\frac{\lambda_{min}}{\omega_m s}}}{1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}} \left[\arctan \left(\sqrt{\frac{\lambda_{max}}{\omega_m s}} \right) - \arctan \left(\sqrt{\frac{\lambda_{min}}{\omega_m s}} \right) \right]. \quad (2.17)$$

o Positively Skewed Linear Distribution

By combining Eqns. 2.16 and 2.9, one obtains:

$$\begin{aligned} g(s) = & \omega_f + \frac{\omega_m \lambda_{min}}{\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)^2} \\ & \times \left[\frac{1}{\omega_m s} \ln \left(\frac{\lambda_{max} \omega_m s + \lambda_{min}}{\lambda_{min} \omega_m s + \lambda_{max}} \right) \right. \\ & \left. - \frac{2}{\sqrt{\omega_m s \lambda_{max}}} \left\{ \arctan \left(\sqrt{\frac{\lambda_{max}}{\omega_m s}} \right) - \arctan \left(\sqrt{\frac{\lambda_{min}}{\omega_m s}} \right) \right\} \right]. \quad (2.18) \end{aligned}$$

o Negatively Skewed Linear Distribution

By combining Eqns. 2.16 and 2.10, one obtains:

$$\begin{aligned} g(s) = & \omega_f + \frac{\omega_m \sqrt{\lambda_{min}}}{\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)^2} \\ & \times \left[\frac{2}{\sqrt{\omega_m s}} \left\{ \arctan \left(\sqrt{\frac{\lambda_{max}}{\omega_m s}} \right) - \arctan \left(\sqrt{\frac{\lambda_{min}}{\omega_m s}} \right) \right\} \right. \\ & \left. - \frac{\sqrt{\lambda_{min}}}{\omega_m s} \ln \left(\frac{\lambda_{max} \omega_m s + \lambda_{min}}{\lambda_{min} \omega_m s + \lambda_{max}} \right) \right]. \quad (2.19) \end{aligned}$$

Unsteady State Solutions

o Arbitrary Distribution

Similarly, let $P(h_D)$ denote any probability density function of matrix block size. One obtains:

$$g(s) = \omega_f + \omega_m \int_{\frac{h_{min}}{h_{max}}}^1 \sqrt{\frac{\lambda}{3\omega_m s}} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) P(h_D) dh_D. \quad (2.20)$$

o Uniform Distribution

Eqns. 2.20 and 2.8 yield:

$$g(s) = \omega_f + \omega_m \frac{\sqrt{\frac{\lambda_{min}}{3\omega_m s}}}{2\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)} \int_{\lambda_{min}}^{\lambda_{max}} \frac{1}{\lambda} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) d\lambda. \quad (2.21)$$

o Positively Skewed Linear Distribution

Eqns. 2.20 and 2.9 yield:

$$\begin{aligned} g(s) = & \omega_f + \frac{\omega_m}{\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)^2} \\ & \times \left[\int_{\lambda_{min}}^{\lambda_{max}} \frac{\lambda_{min}}{\sqrt{3\omega_m s}} \frac{1}{\lambda^{\frac{3}{2}}} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) d\lambda \right. \\ & \left. - \int_{\lambda_{min}}^{\lambda_{max}} \frac{\lambda_{min}}{\sqrt{3\omega_m s}} \frac{1}{\sqrt{\lambda_{max}}} \frac{1}{\lambda} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) d\lambda \right]. \quad (2.22) \end{aligned}$$

o Negatively Skewed Linear Distribution

Eqns. 2.20 and 2.10 yield:

$$\begin{aligned} g(s) = & \omega_f + \frac{\omega_m}{\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)^2} \\ & \times \left[\int_{\lambda_{min}}^{\lambda_{max}} \frac{\sqrt{\lambda_{min}}}{\sqrt{3\omega_m s}} \frac{1}{\lambda} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) d\lambda \right. \\ & \left. - \int_{\lambda_{min}}^{\lambda_{max}} \frac{\lambda_{min}}{\sqrt{3\omega_m s}} \frac{1}{\lambda^{\frac{3}{2}}} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) d\lambda \right]. \quad (2.23) \end{aligned}$$

2.3.3 Solutions for Cylindrical Geometry

In this model, the matrix blocks are assumed to be of cylindrical shape and are separated by vertical fractures. The schematic of this geometry is shown in Fig. 2.5.

The PSS solutions are the same as those given for the slab geometry, Eqns. 2.16 through 2.19 (but different definition of λ as discussed in section 2.3.5 is used). The USS solutions are given below.

o Arbitrary Distribution

Let $P(r_{mD})$ denote any probability density function of matrix block size. One obtains:

$$g(s) = \omega_f + \omega_m \int_{\frac{r_{mmin}}{r_{max}}}^1 \sqrt{\frac{2\lambda}{3\omega_m s}} \frac{I_1\left(\sqrt{\frac{6\omega_m s}{\lambda}}\right)}{I_0\left(\sqrt{\frac{6\omega_m s}{\lambda}}\right)} P(r_{mD}) dr_{mD}. \quad (2.24)$$

o Uniform Distribution

Eqns. 2.24 and 2.8 yield:

$$g(s) = \omega_f + \omega_m \frac{\sqrt{\frac{2\lambda_{min}}{3\omega_m s}}}{2\left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)} \int_{\lambda_{min}}^{\lambda_{max}} \frac{1}{\lambda} \frac{I_1\left(\sqrt{\frac{6\omega_m s}{\lambda}}\right)}{I_0\left(\sqrt{\frac{6\omega_m s}{\lambda}}\right)} d\lambda. \quad (2.25)$$

2.3.4 Solutions for Spherical Geometry

This model is shown in Fig. 2.6. Matrix blocks have spherical shape and are surrounded by fractures. The PSS solutions are the same as those given for the slab or the cylindrical geometry, Eqns. 2.16 through 2.19 (but different definition of λ as discussed in section 2.3.5 is used). The USS solutions are given below.

• Arbitrary Distribution

Let $P(r_{mD})$ denote any probability density function of matrix block size. One obtains:

$$g(s) = \omega_f + \omega_m \int_{\frac{r_{mmin}}{r_{max}}}^1 \sqrt{\frac{\lambda}{\omega_m s}} \coth\left(\sqrt{\frac{9\omega_m s}{\lambda}}\right) P(r_{mD}) dr_{mD} - \frac{1}{3s} \int_{\frac{r_{mmin}}{r_{max}}}^1 \lambda P(r_{mD}) dr_{mD}. \quad (2.26)$$

- Uniform Distribution

Eqns. 2.26 and 2.8 yield:

$$g(s) = \omega_f + \omega_m \frac{\sqrt{\frac{\lambda_{min}}{\omega_m s}}}{2 \left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)} \int_{\lambda_{min}}^{\lambda_{max}} \frac{1}{\lambda} \coth \left(\sqrt{\frac{9\omega_m s}{\lambda}} \right) d\lambda - \frac{\sqrt{\lambda_{min}}}{6s \left(1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}\right)} \int_{\lambda_{min}}^{\lambda_{max}} \frac{d\lambda}{\sqrt{\lambda}}. \quad (2.27)$$

2.3.5 Dimensionless Parameters

The following dimensionless parameters are used:

$$P_{Df} = \frac{2\pi k_f h_f (P_i - P_f)}{q\mu B}, \quad (2.28)$$

$$P_{Dm} = \frac{2\pi k_f h_f (P_i - P_m)}{q\mu B}, \quad (2.29)$$

$$t_D = \frac{k_f t}{(\phi_f c_f + \phi_m c_m) \mu r_w^2}, \quad (2.30)$$

$$r_D = \frac{r}{r_w}, \quad (2.31)$$

$$\omega_m = \frac{\phi_m c_m}{\phi_f c_f + \phi_m c_m}, \quad (2.32)$$

$$\omega_f = 1 - \omega_m. \quad (2.33)$$

For slab geometry:

$$\lambda = \alpha \frac{k_m r_w^2}{k_f h^2}, \quad (2.34)$$

$$z_D = \frac{z}{h}, \quad (2.35)$$

$$h_D = \frac{h}{h_{max}}. \quad (2.36)$$

For cylindrical and spherical geometry:

$$\lambda = \alpha \frac{k_m r_w^2}{k_f r_m^2}, \quad (2.37)$$

$$\eta_D = \frac{\eta}{r_m}, \quad (2.38)$$

$$r_{mD} = \frac{r_m}{r_{maz}} \quad (2.39)$$

The constant a in the definition of interporosity flow coefficient X is given in Table 2-1.

Table 2.1: Constant a for various matrix geometries

Geometry	α
Slab	3
Cylinder	6
Sphere	9

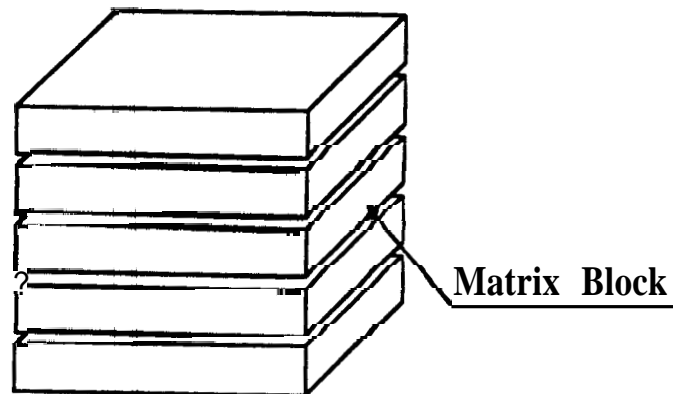


Figure 2.4: Matrix Slab Geometry

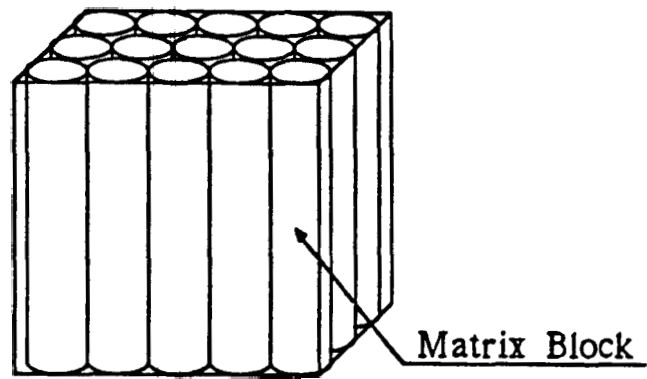


Figure 2.5: Matrix Cylindrical Geometry

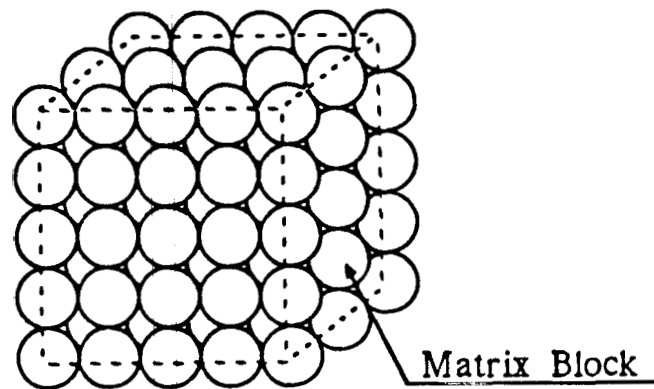


Figure 2.6: Matrix Spherical Geometry

Section 3

Discussion

In this work, flowrate and pressure response are considered and the factors which affect them are investigated. To convert the rate and pressure solutions from Laplace space into **real** space, Stehfest Algorithm [23] is used. The unsteady state solutions are evaluated using numerical integration.

3.1 Effect of Matrix Block Geometry

Figs. 3.1 through 3.3 show the rate response for the uniform distribution and **USS** interporosity flow for slab, cylindrical, and spherical matrix block geometries, respectively. Each figure is for **one** value of ω_m and λ_{max} , and several values of λ_{ratio} . The parameter λ_{ratio} is defined **as**:

$$\lambda_{ratio} = \frac{\lambda_{max}}{\lambda_{min}}, \quad (3.1)$$

which is a measure of the spread (variance) of the matrix block size distribution and determines the duration of the interporosity **flow** period. In each of the Figs. 3.1 through 3.3, the interporosity flow period for a given λ_{ratio} begins and ends at the same time for the three geometries. This occurs because the duration of the interporosity flow period depends **only** on λ_{ratio} . The effect **of** **block** geometry appears only in the interporosity flow period. Fig. 3.4 shows that the spherical model yields a higher flowrate than the Cylindrical model, which in **turn** yields a

higher flowrate than the slab model. The differences are, however, not significant. **Also, as Fig. 3.5 indicates, as λ_{ratio} increases, the difference in flowrate decreases.**

Figs. 3.6 through 3.8 show the pressure and the pressure derivative response for constant rate production for slab, cylindrical, and spherical geometries, respectively. Fig. 3.9 compares the pressure derivative response for the three geometries. The spherical model shows a smoother derivative profile than the cylindrical or the slab models, especially for small λ_{ratio} . Due to the striking similarity of the rate and the pressure response for the three block geometries, **only the slab model will be used for the remainder of this study.**

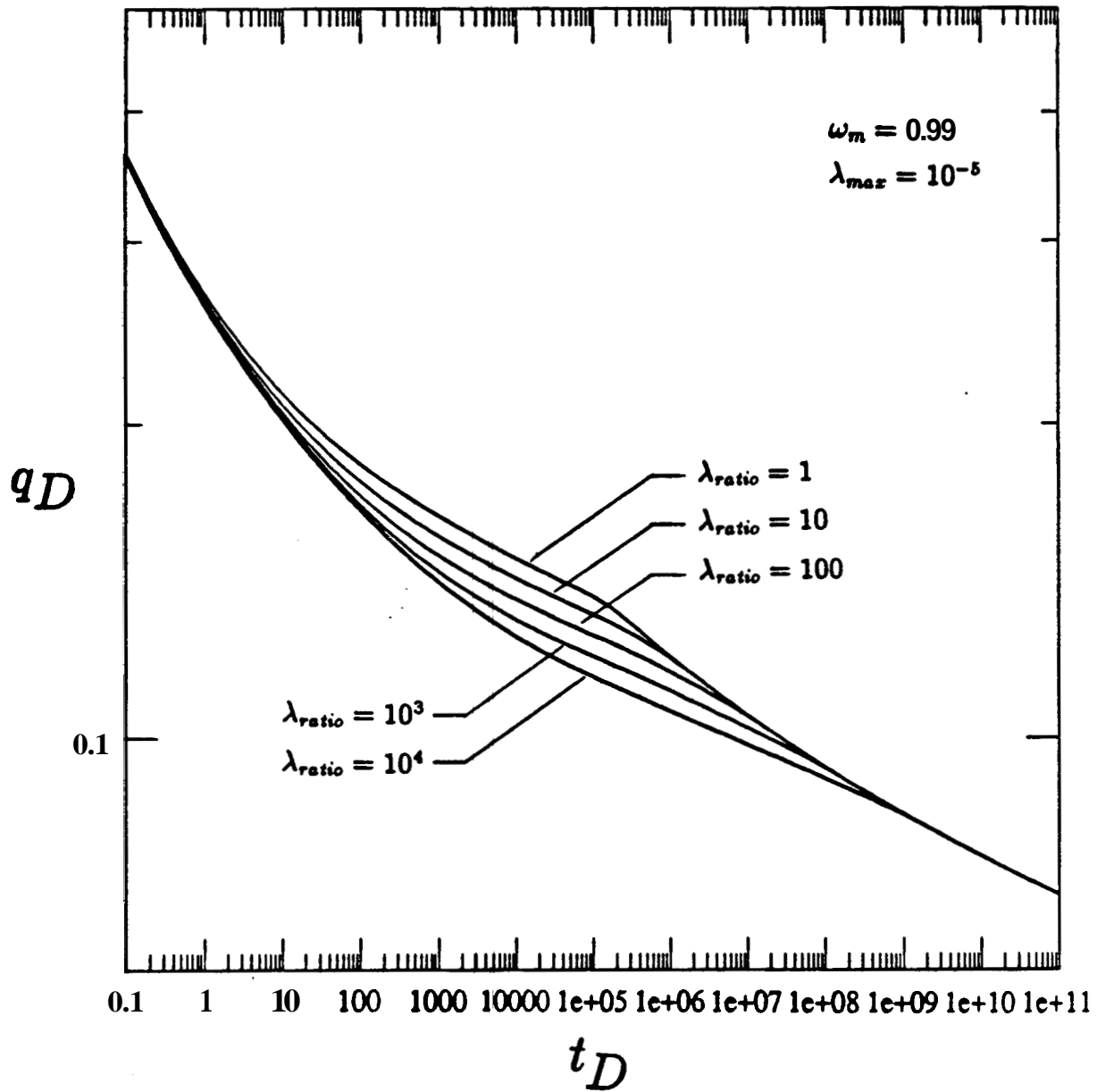


Figure 3.1: Flowrate profile for slab geometry in USS model

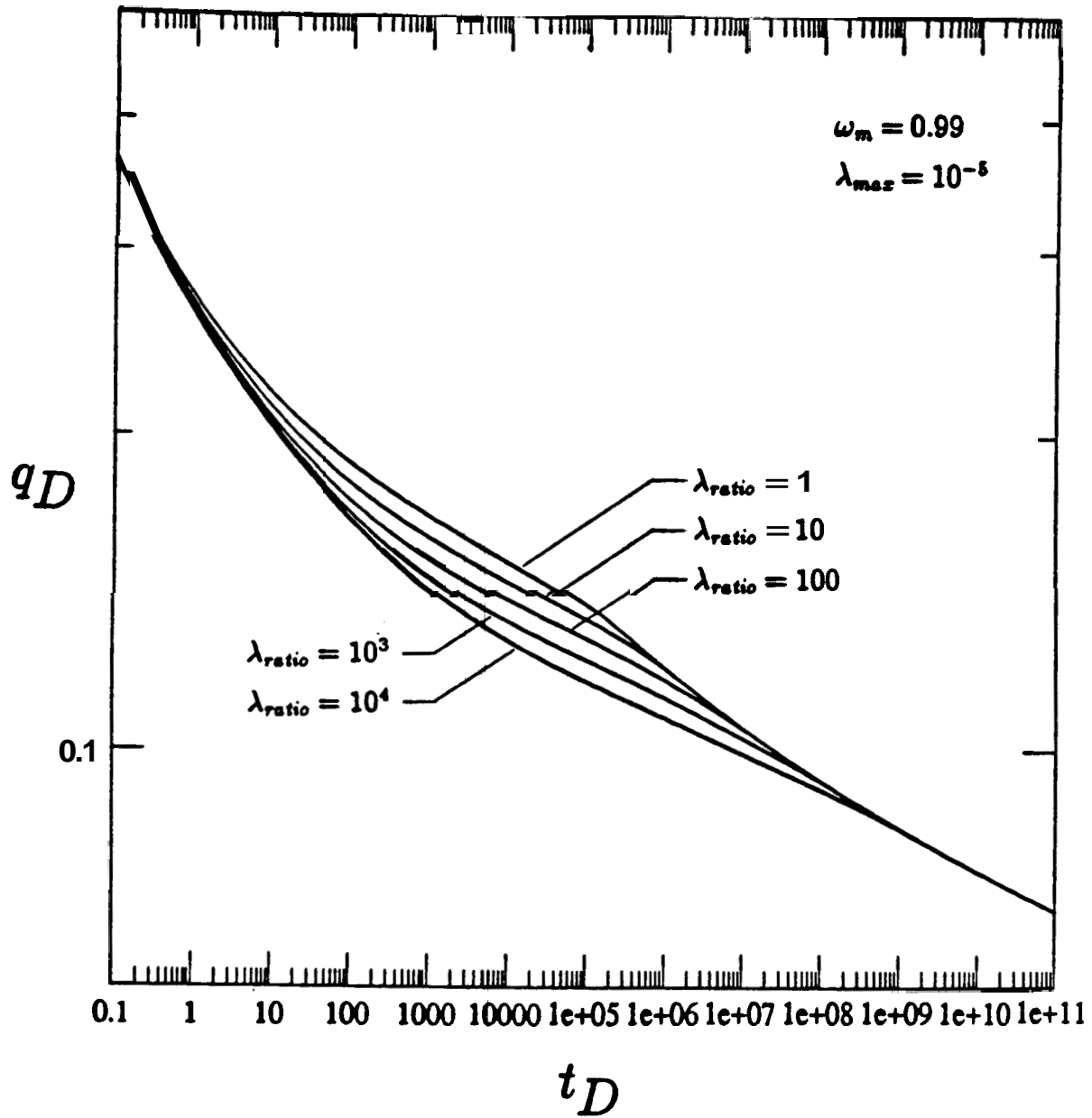


Figure 3.2: Flowrate profile for cylindrical geometry in USS model

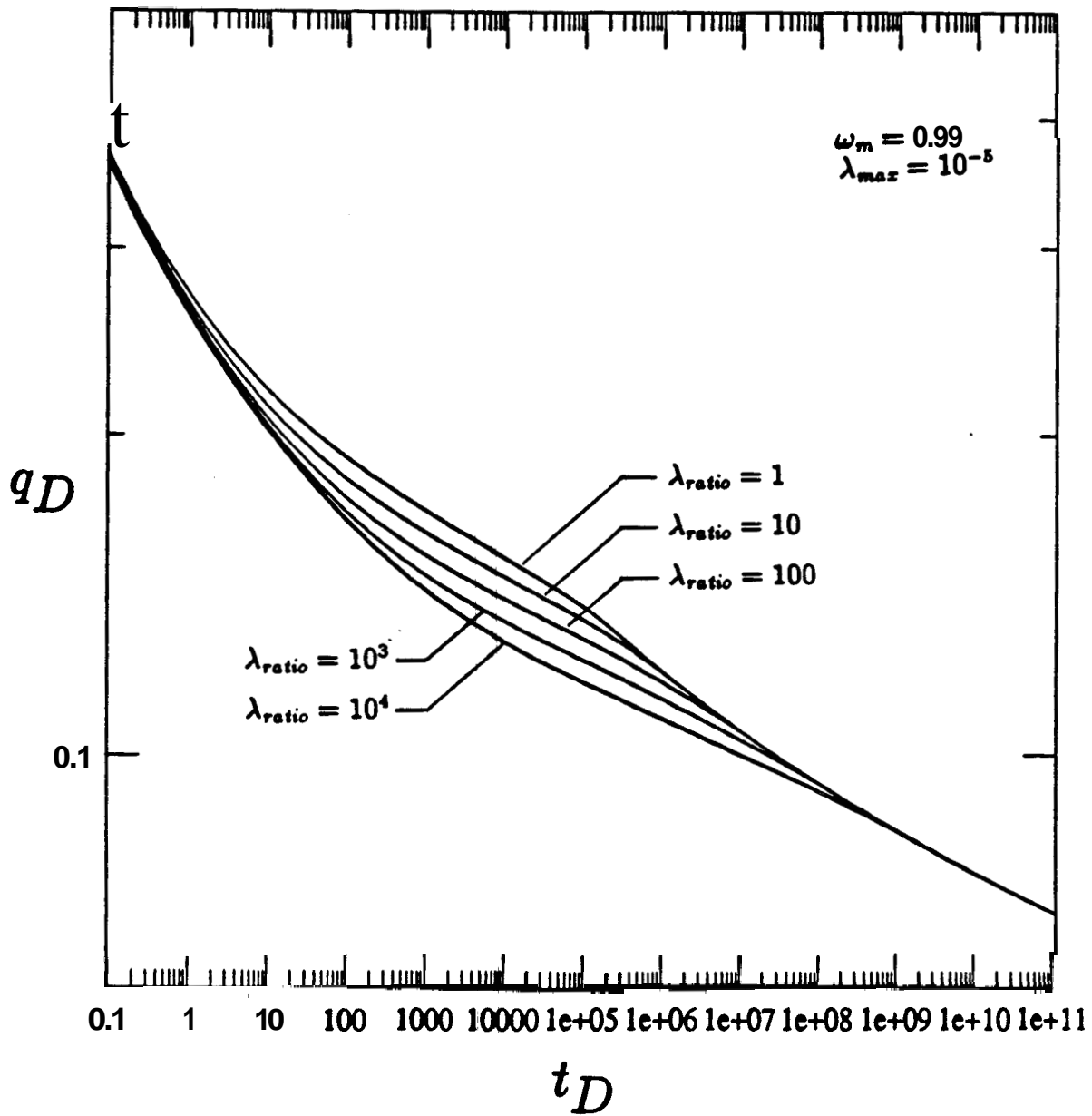


Figure 3.3: Flowrate profile for spherical geometry in USS model

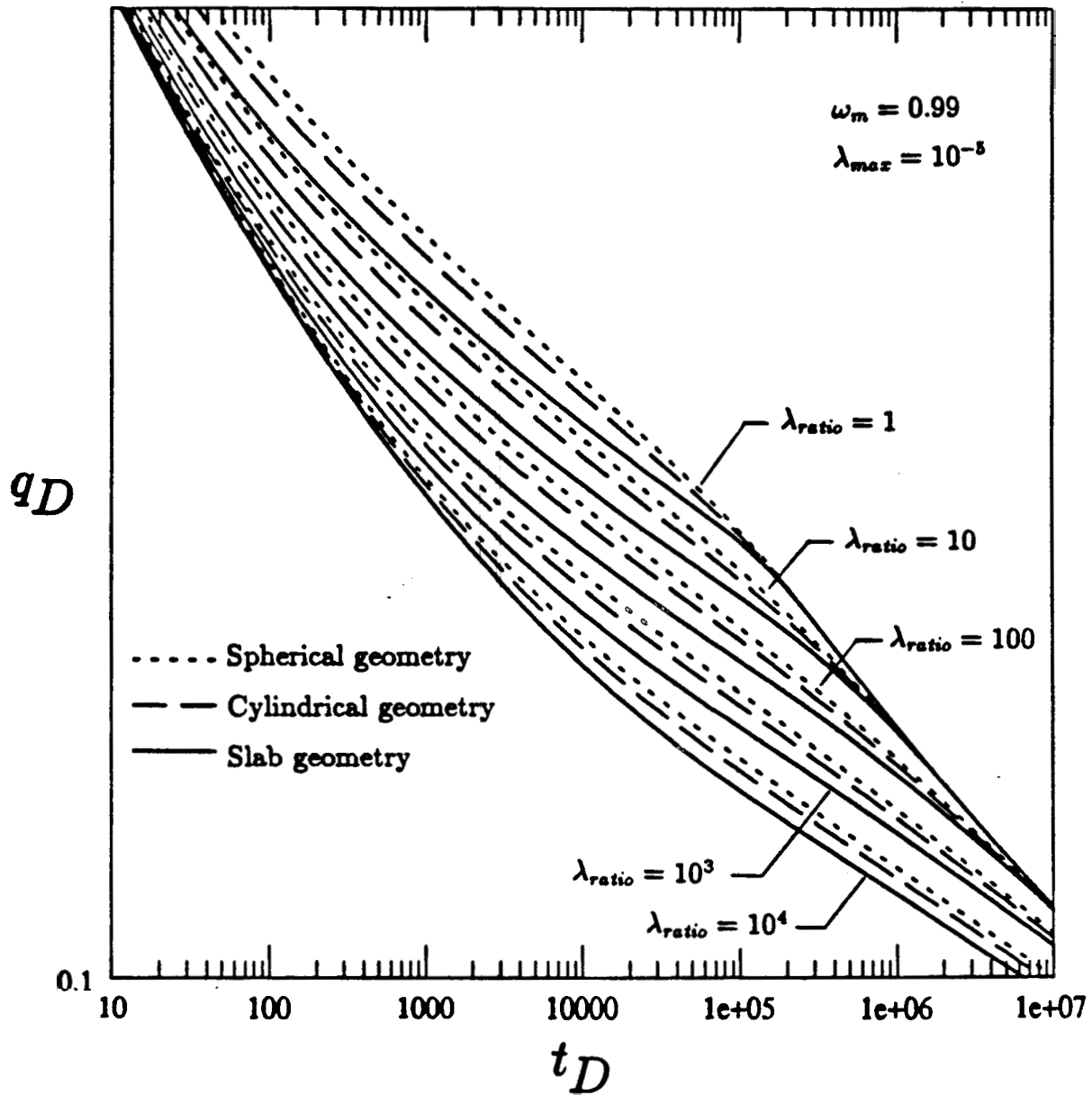


Figure 3.4: Comparison of flowrate response of various matrix block geometries

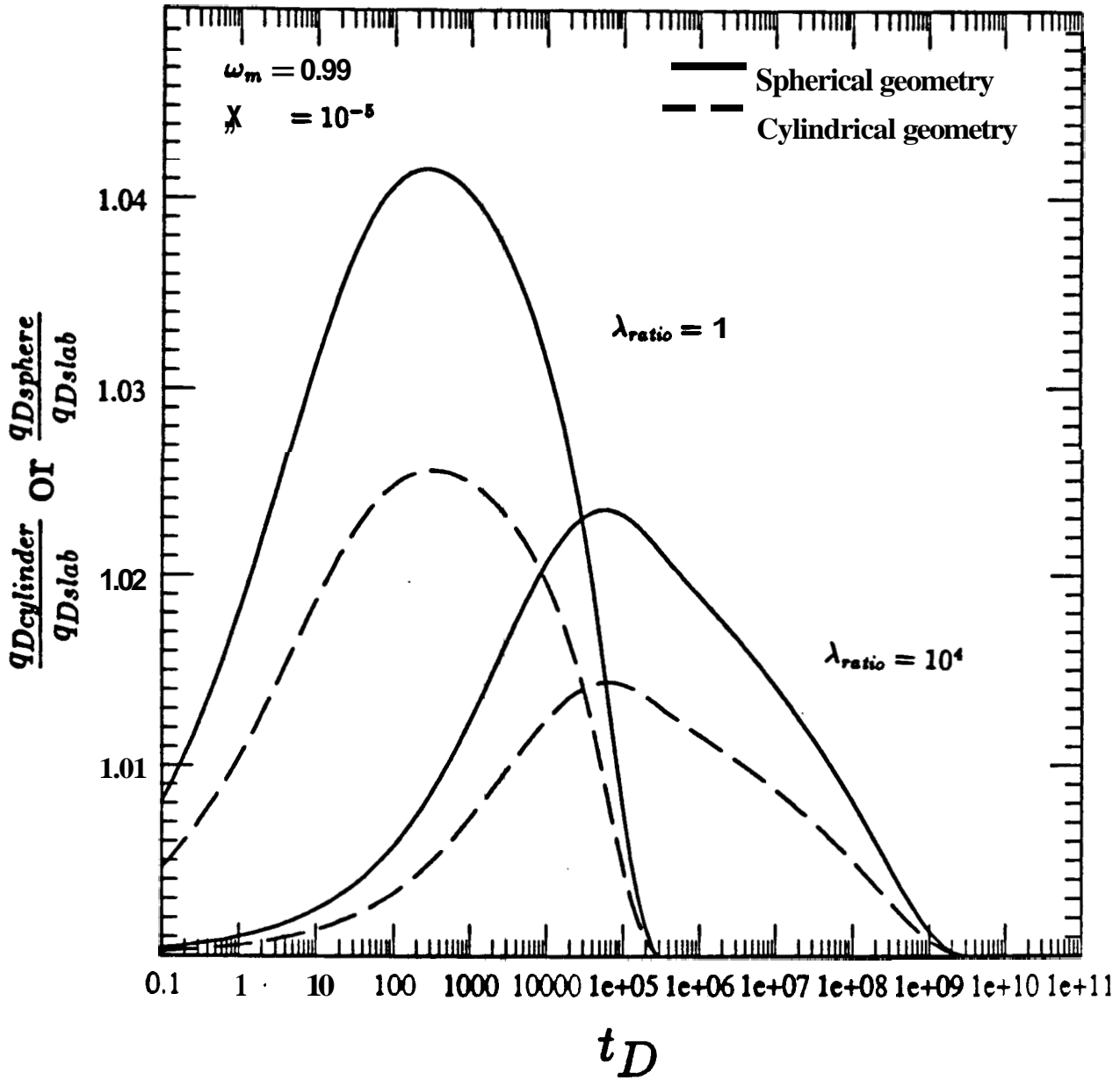


Figure 3.5: Difference in flowrate of various matrix block geometries

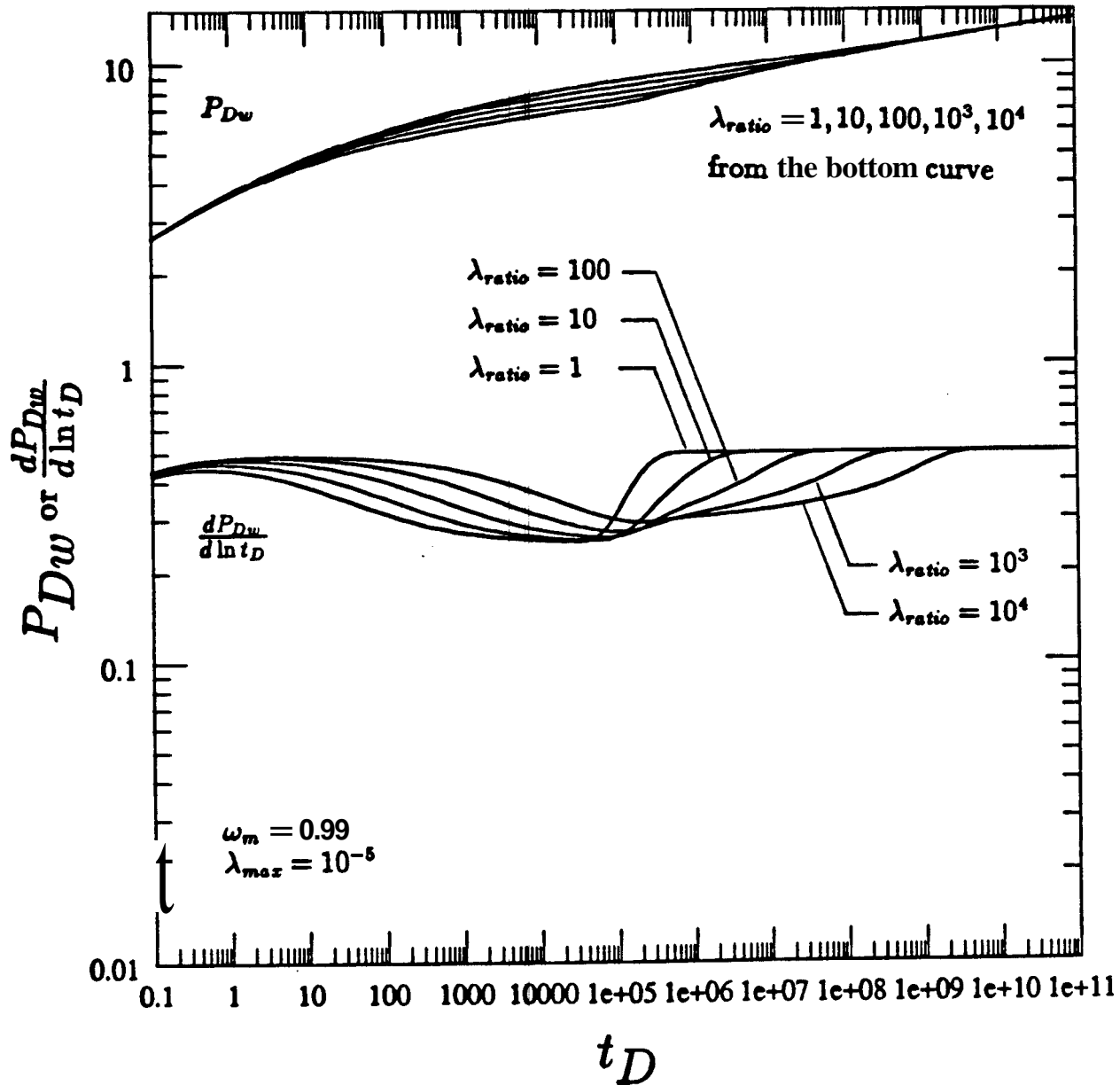


Figure 3.6: Pressure and pressure derivative profile for slab geometry in USS model

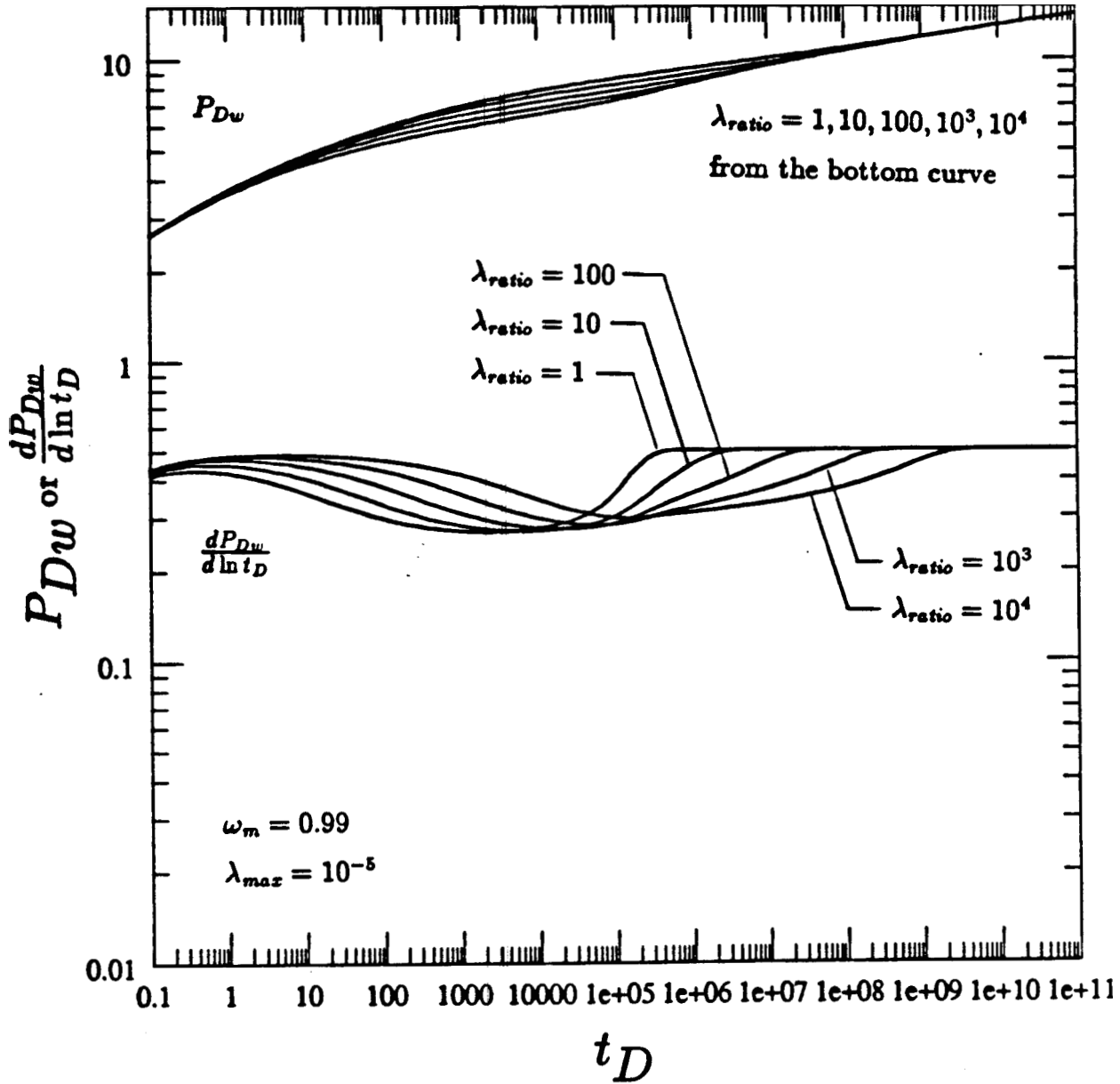


Figure 3.7: Pressure and pressure derivative profile for cylindrical geometry in USS model

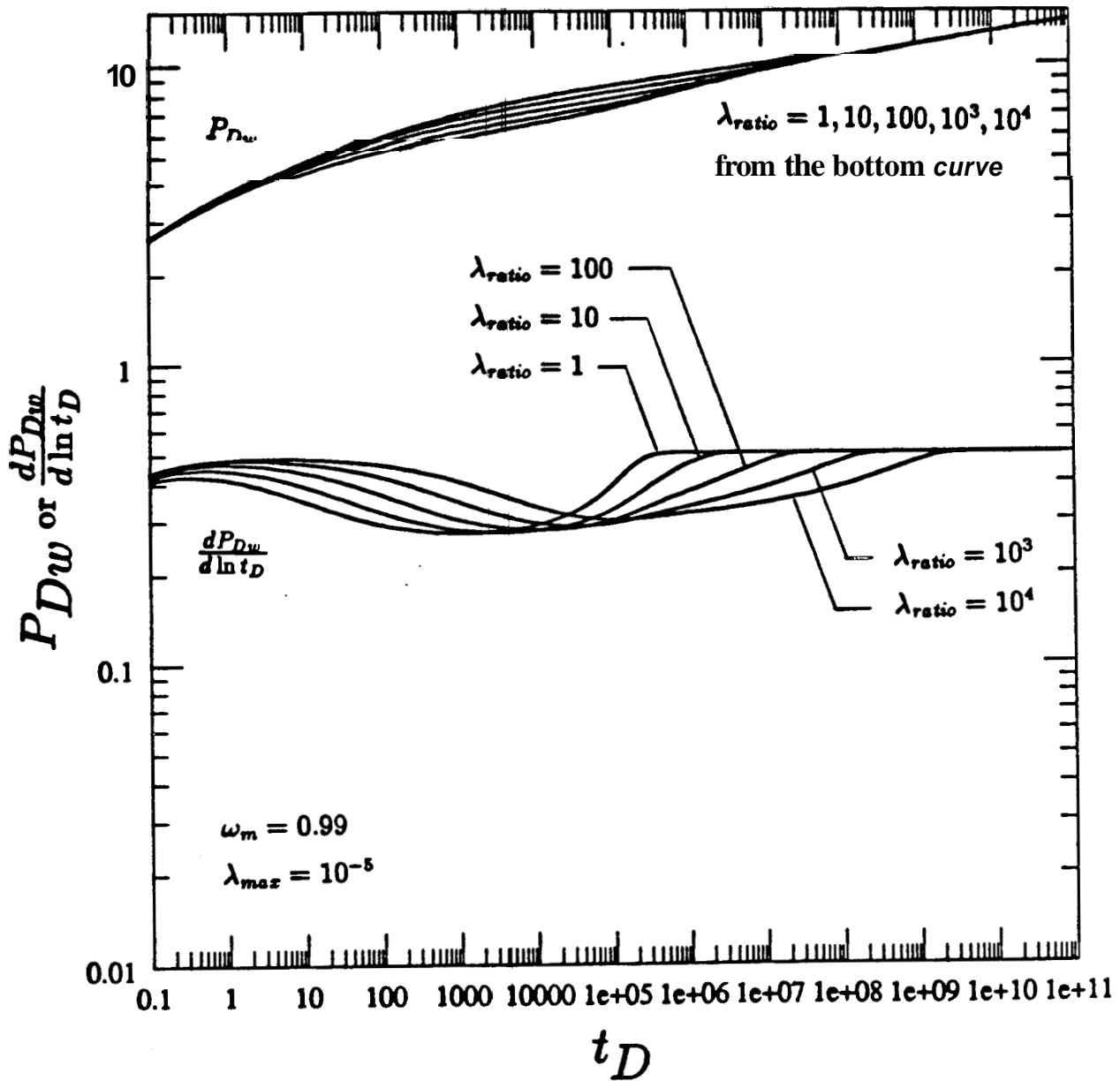


Figure 3.8: Pressure and pressure derivative profile for spherical geometry in USS model

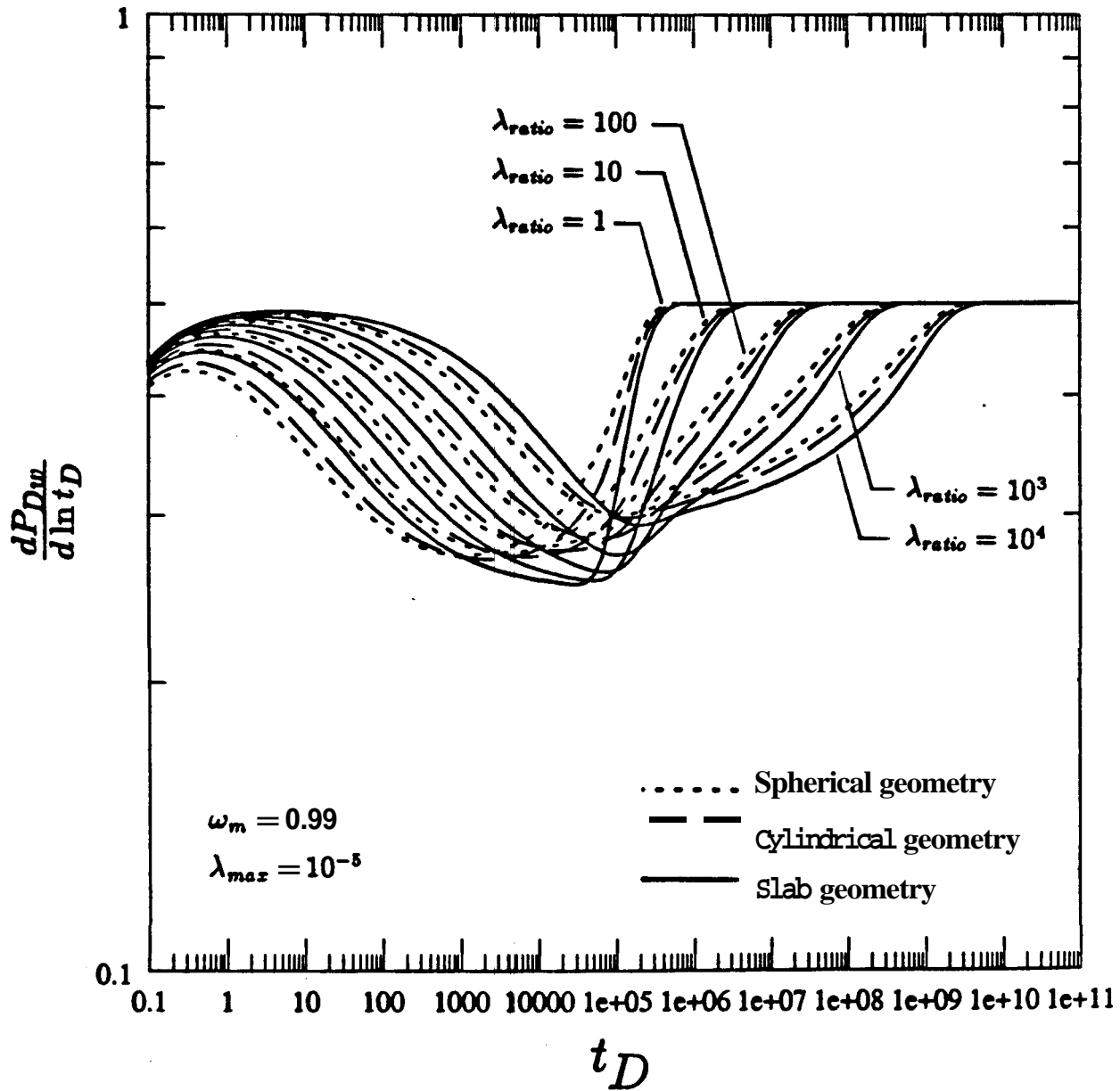


Figure 3.9: Comparison of pressure derivative profile of various matrix block geometries

3.2 Effect of the Mode of Interporosity Flow

Fig. 3.10 is a flowrate profile that compares the **USS** and **PSS** response for several values of λ_{ratio} . The interporosity flow period ends at the same time for these two models, however, the beginning time is different. In the **USS** model, the interporosity flow period begins much earlier than in the **PSS** model. Thus, the duration of the interporosity flow period of the **USS** model is much longer than that of the **PSS** model. Also, the **USS** model yields a higher flowrate than the **PSS** model during the interporosity flow period. Both models produce at exactly the same flowrate at the end of this period. Thus, the **USS** model yields a larger cumulative production. In the **PSS** model, the flowrate during the interporosity flow period is almost constant for small λ_{ratio} values. In the **USS** model, on the other hand, a period of constant flowrate does not occur and a gradual decline is observed throughout the interporosity flow period. This gradual flowrate decline is particularly pronounced for large λ_{ratio} values. Fig. 3.11 shows the comparison of pressure derivative profiles for the **PSS** and the **USS** models. The figure shows the sharp character of the **PSS** profile compared to the **USS** profile.

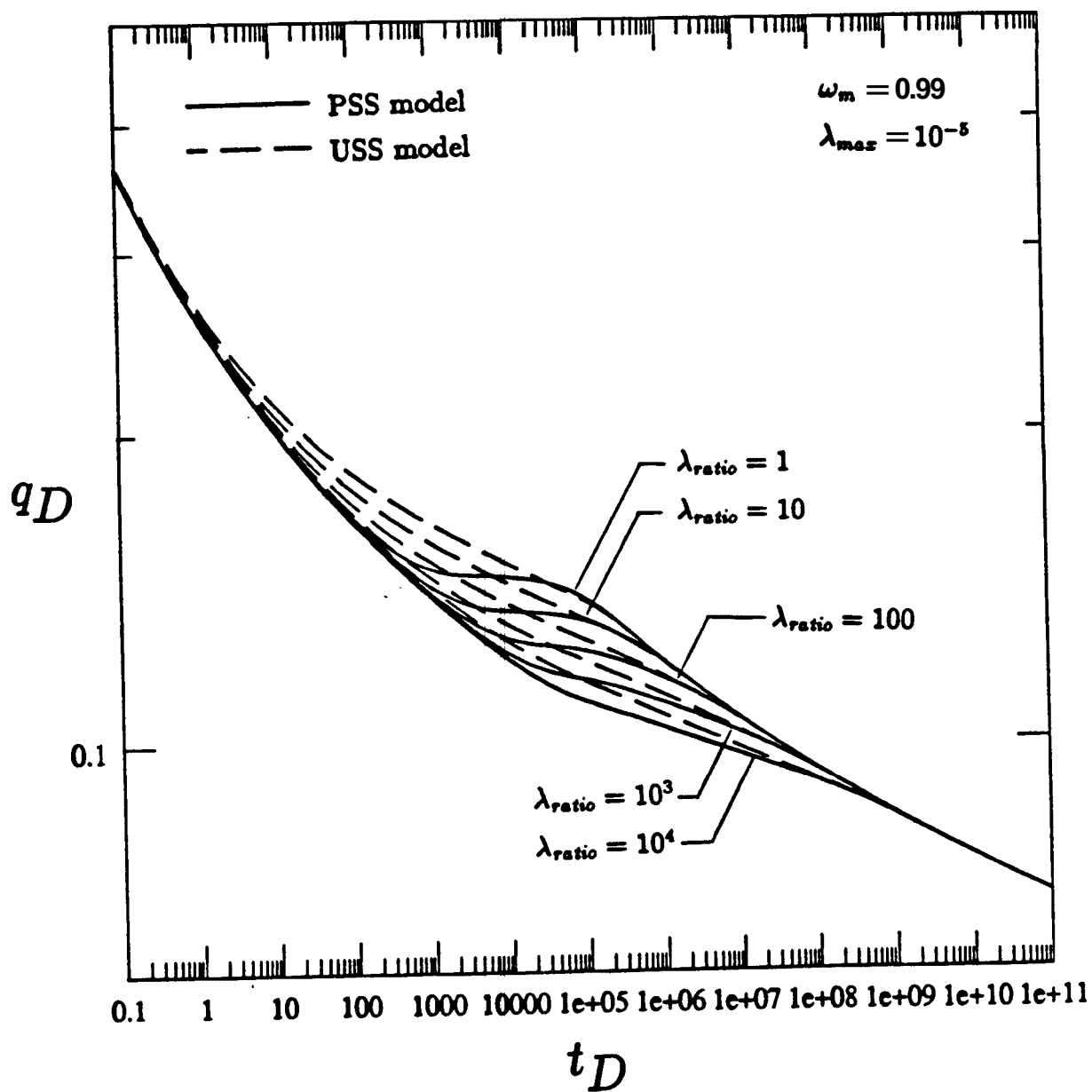


Figure 3.10: Comparison of PSS and USS flowrate response

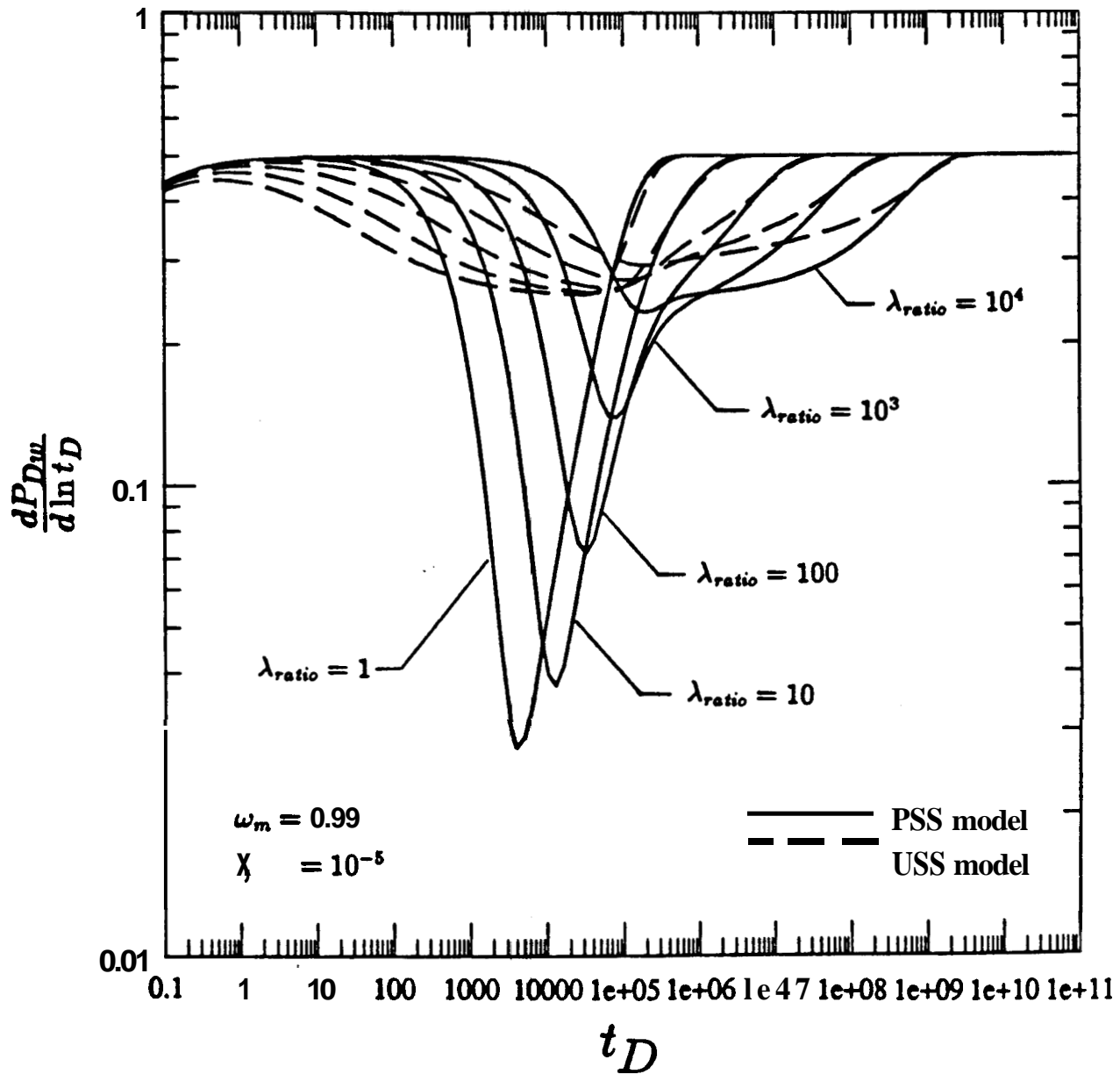


Figure 3.11: Comparison of PSS and USS pressure derivative profiles

3.3 Effect of PDF Type

In this section, the rate and the pressure response for three matrix block size distributions — uniform, positively skewed linear, and negatively skewed linear distributions, are examined. Both **PSS** and **USS** modes of interporosity flow are considered. Only the slab matrix geometry is used, since the transient response is not very sensitive to the matrix block geometry (see section 3.1).

Fig. 3.12 shows the PSS flowrate profile for the three probability density functions. The duration of the interporosity flow period is the same for the three models. The probability distribution function does not affect the duration of this period as long as the limits of the distribution (λ_{min} and λ_{max}) are the same. The response shows the lowest flowrate for the positively skewed linear distribution and the highest flowrate for the negatively skewed linear distribution. This indicates that the higher frequency of smaller matrix block sizes yields a higher flowrate. The flowrate for the uniform distribution is between that of the positively and the negatively skewed linear distributions, but is closer to the latter. This implies that the adverse effect of large blocks on reservoir producibility outweighs the advantage of small blocks. Fig. 3.13 shows the PSS pressure and the pressure derivative response for the three distributions. The positively skewed distribution, which corresponds to a higher frequency of larger blocks, shows a delayed response.

Figs. 3.14 and 3.15 show the rate profile and the pressure derivative profile for the **USS** model, respectively. The effect of the probability density function is less significant compared to the **PSS** model. However, the features which were mentioned with regard to the **PSS** model are observed here as well.

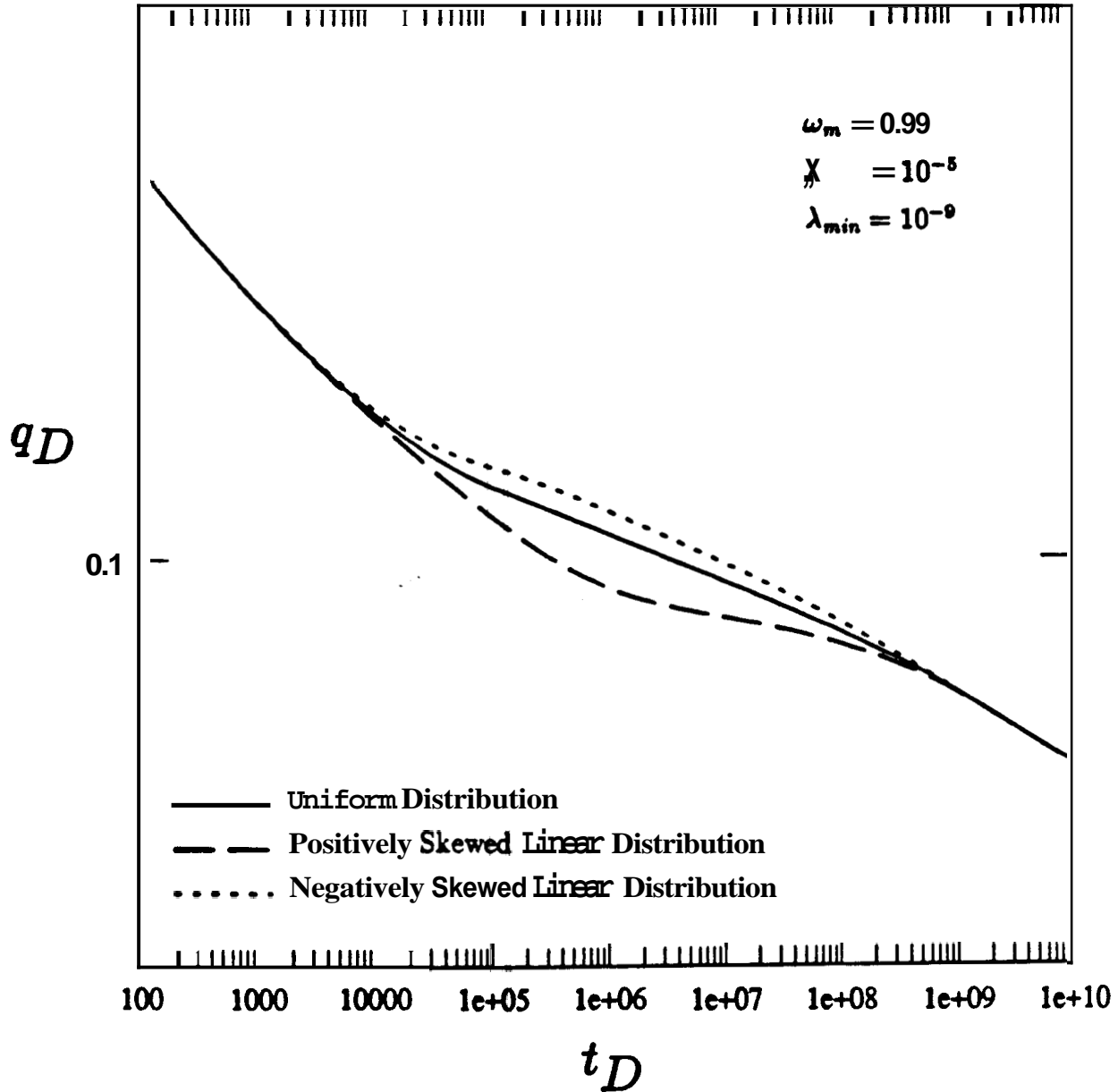


Figure 3.12: PSS flowrate profile for various matrix block size distributions

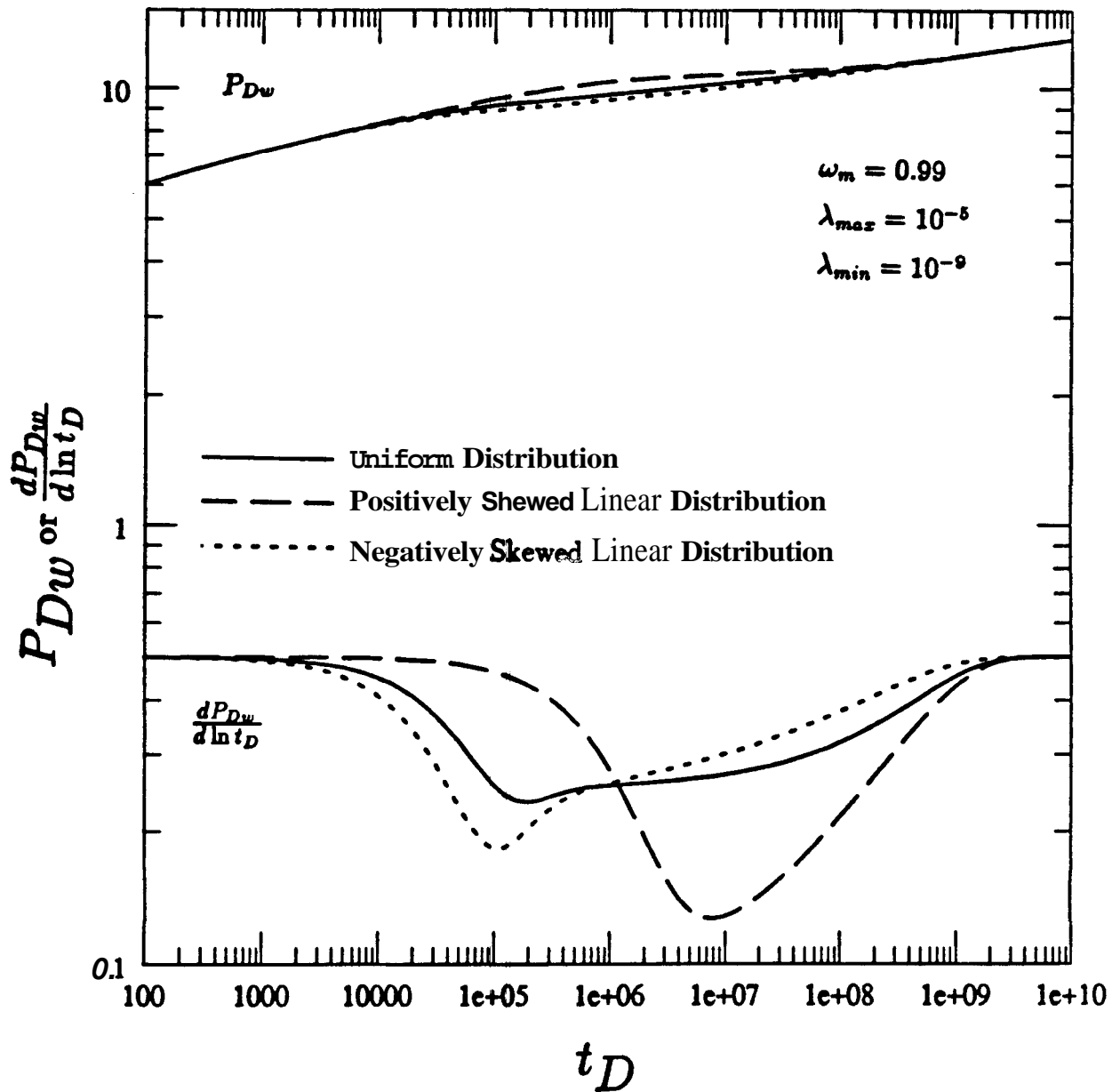


Figure 3.13: PSS pressure and pressure derivative profile for various matrix block size distributions

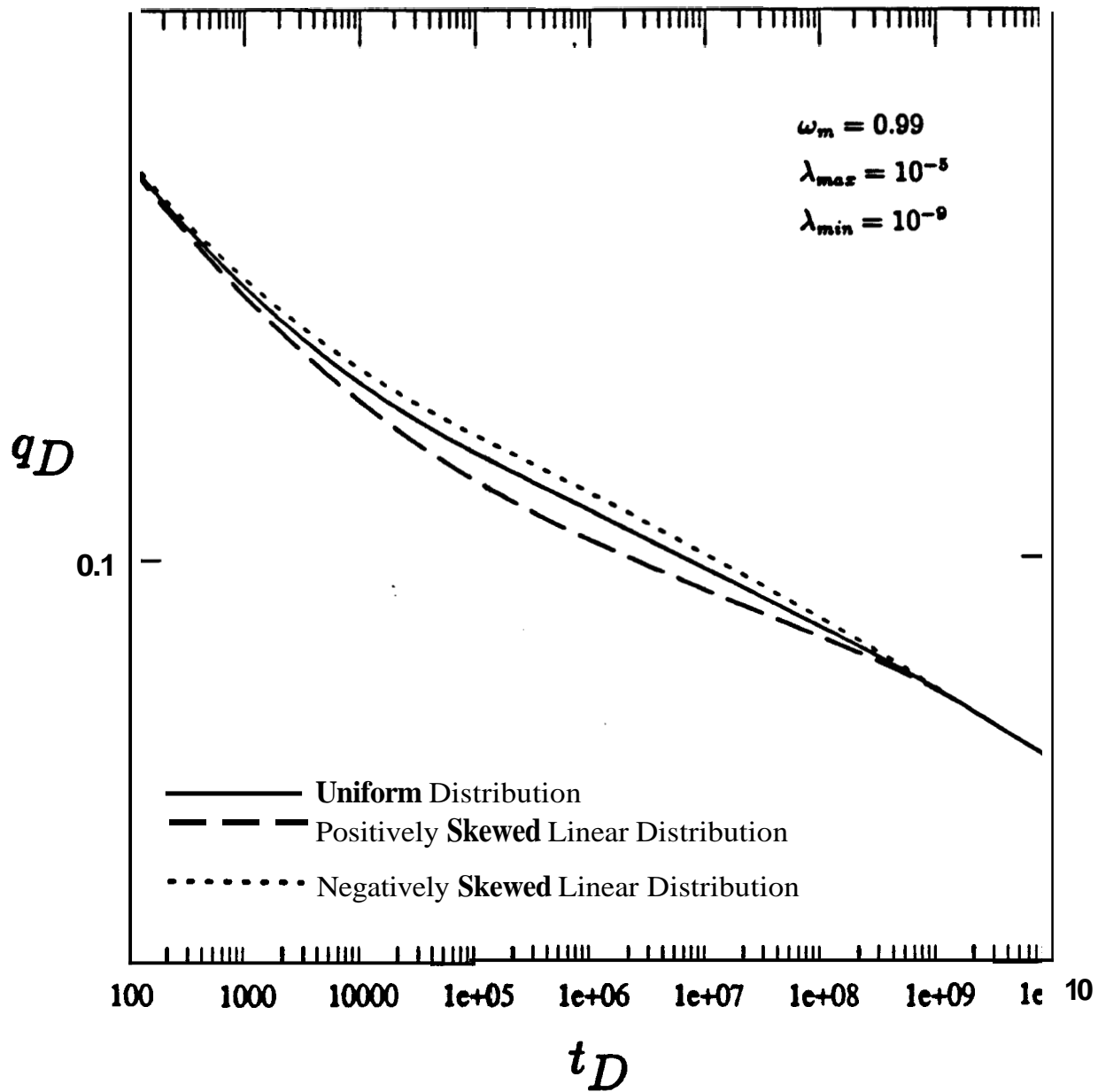


Figure 3.14: USS flowrate profile for various matrix block size distributions

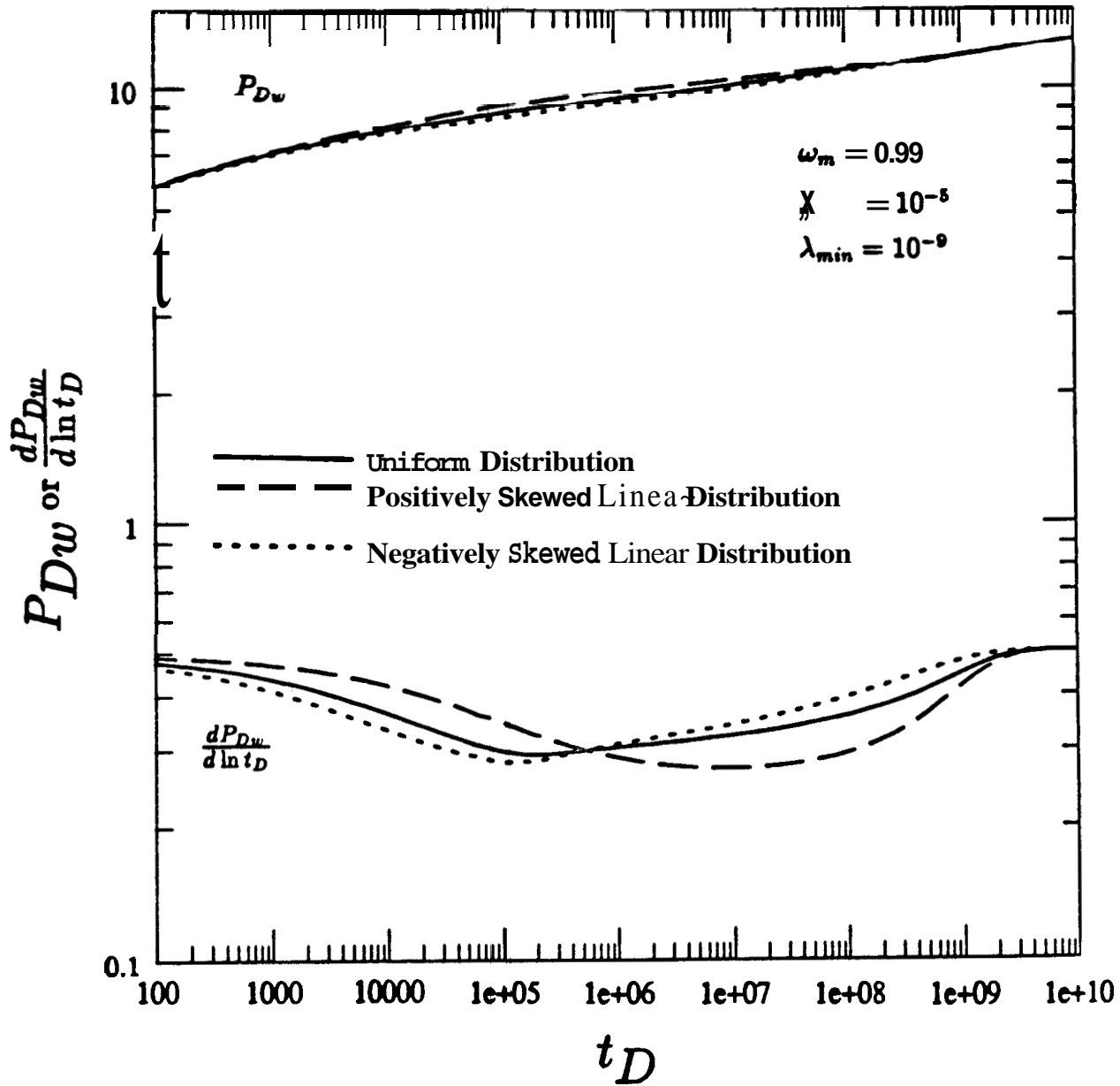


Figure 3.15: USS pressure and pressure derivative profile for various matrix block size distributions

3.4 Effect of Fracture Intensity and Uniformity

Here, the effect of the mean and the variance of the matrix block size distribution on flowrate and cumulative recovery is considered. This is done for the uniform distribution and slab geometry for both PSS and USS models.

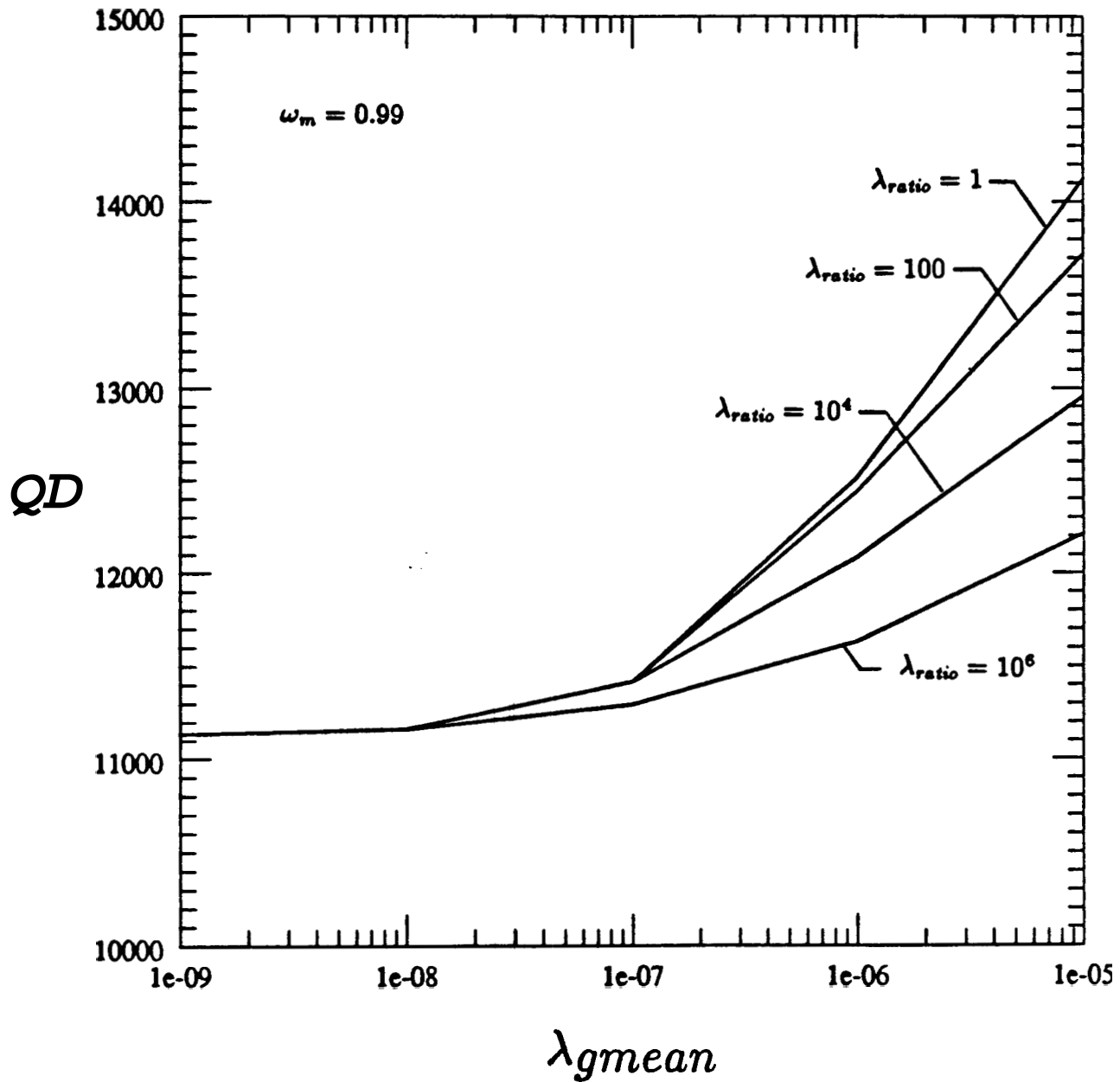
For PSS interporosity flow model, Fig. 3.16 shows the cumulative production at $t_D = 10^5$ versus λ_{gmean} with λ_{ratio} as a parameter. The parameter λ_{gmean} is the geometrical mean of the distributed interporosity flow coefficient and is given by:

$$\lambda_{gmean} = \sqrt{\lambda_{max}\lambda_{min}}. \quad (3.2)$$

Two trends are apparent. First, cumulative recovery is directly proportional to λ_{gmean} . That is, as λ_{gmean} increases, i.e., as matrix blocks become smaller or fracture intensity becomes larger, then cumulative recovery becomes larger for a given λ_{ratio} .

Second, for a given λ_{gmean} , as λ_{ratio} increases, cumulative recovery decreases. That is, as matrix block size variability increases or fracturing becomes more nonuniform, then cumulative recovery decreases. For a given λ_{gmean} , the maximum cumulative recovery is given by a λ_{ratio} of unity, which is the smallest possible value for λ_{ratio} and represents a reservoir with perfectly uniform fracturing, i.e., a Warren and Root type model. The effect of λ_{ratio} is more pronounced when λ_{gmean} is large. On the other hand, the effect of λ_{ratio} is negligible for small λ_{gmean} . For the USS interporosity flow model, Fig. 3.17 shows identical results.

Figs. 3.18 and 3.19 show the PSS rate response and the pressure derivative profile, respectively. It is observed that the time at which the interporosity flow period begins is determined by λ_{gmean} . Also, the time at which this period ends is determined by λ_{min} . In Fig. 3.19, the shape of the pressure derivative profile is determined by λ_{ratio} , whereas its temporal position is determined by λ_{gmean} . Thus, variations in fracture intensity (or λ_{gmean}) only affect the temporal position of the response, whereas variations in fracture uniformity (or λ_{ratio}) affect the shape of the transient response.

Figure 3.16: Cumulative production at $t_D = 10^5$ for PSS model

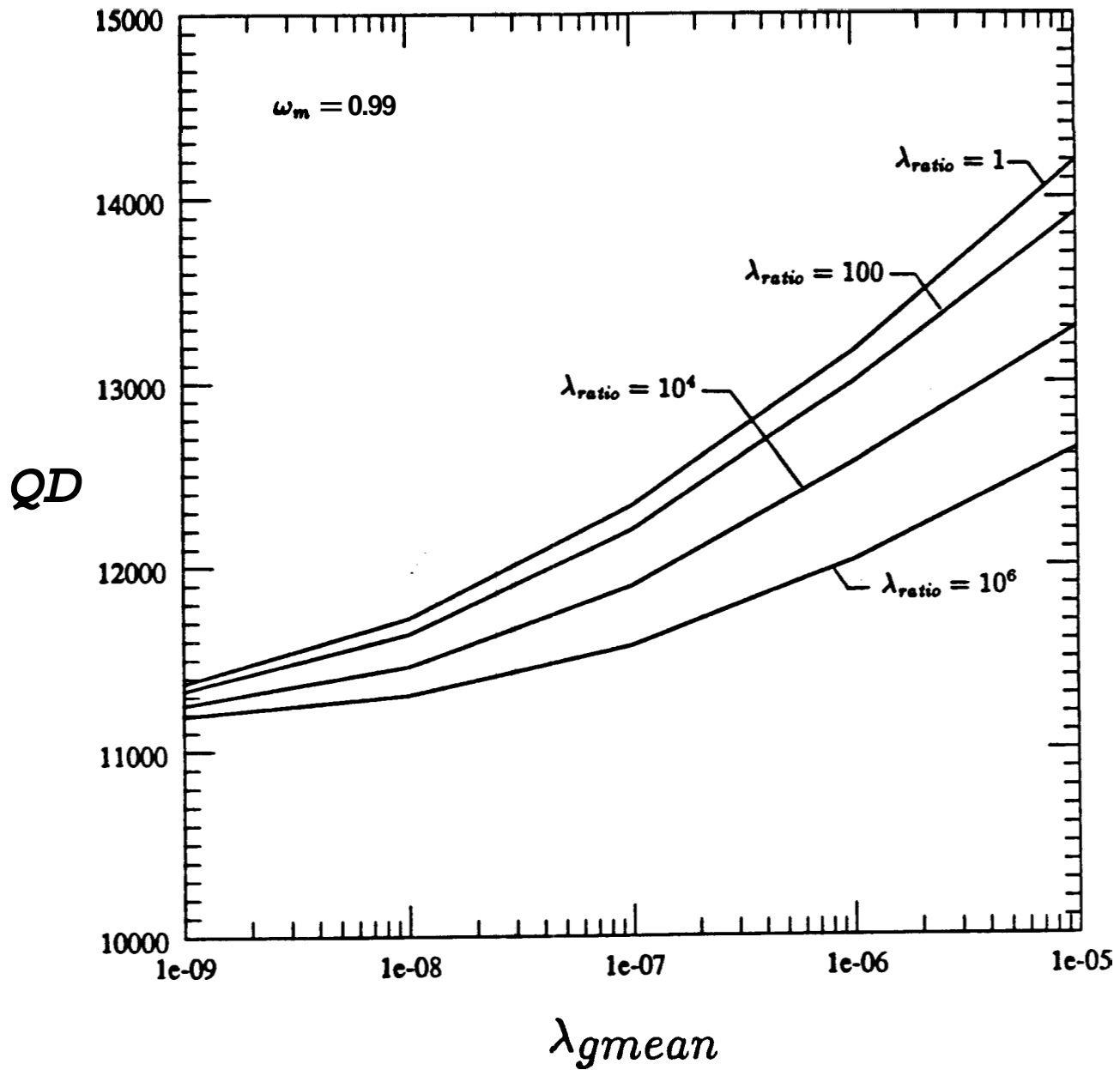
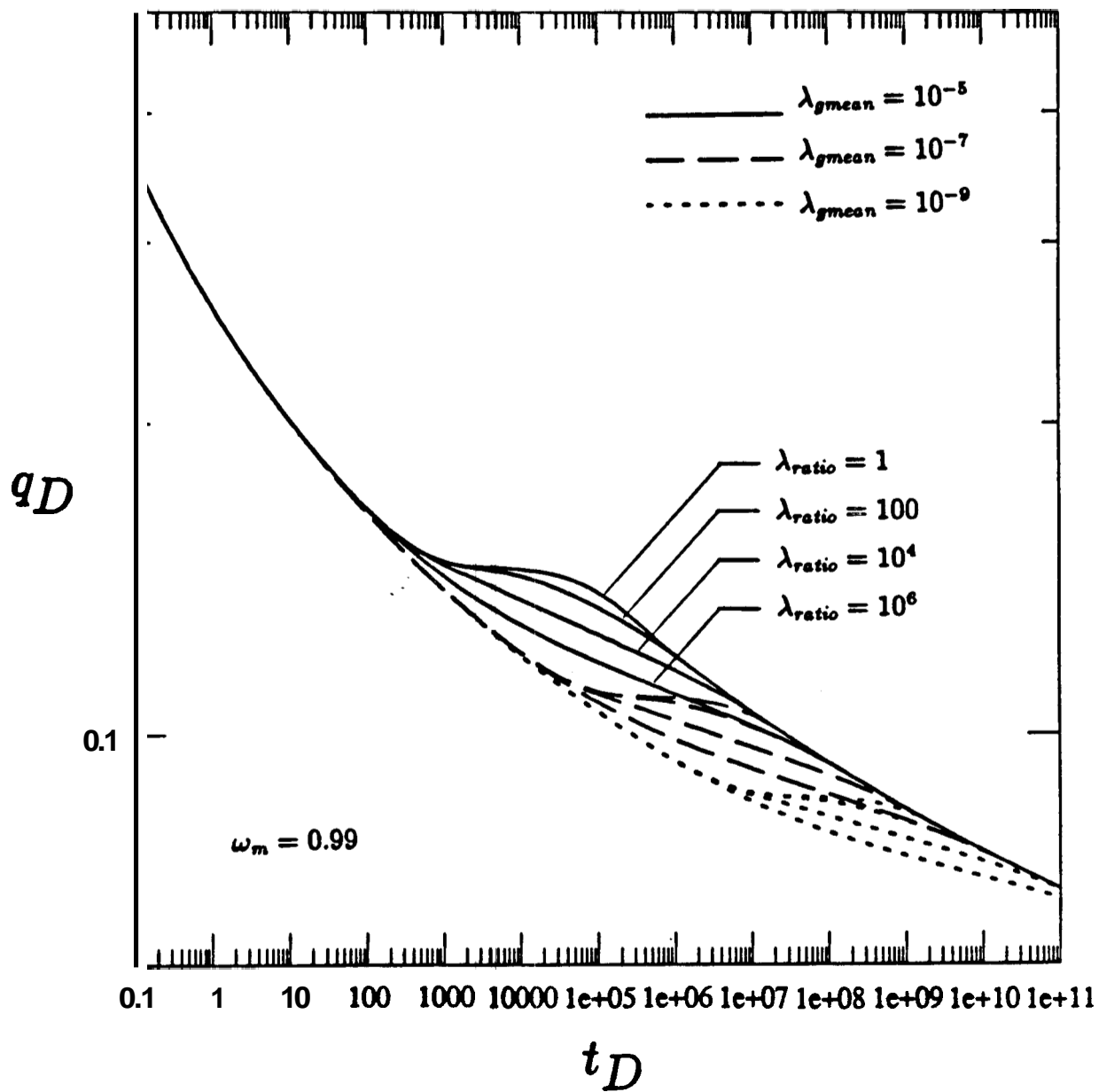


Figure 3.17: Cumulative production at $t_D = 10^5$ for USS model

Figure 3.18: Flowrate profile for various λ_{gmean}

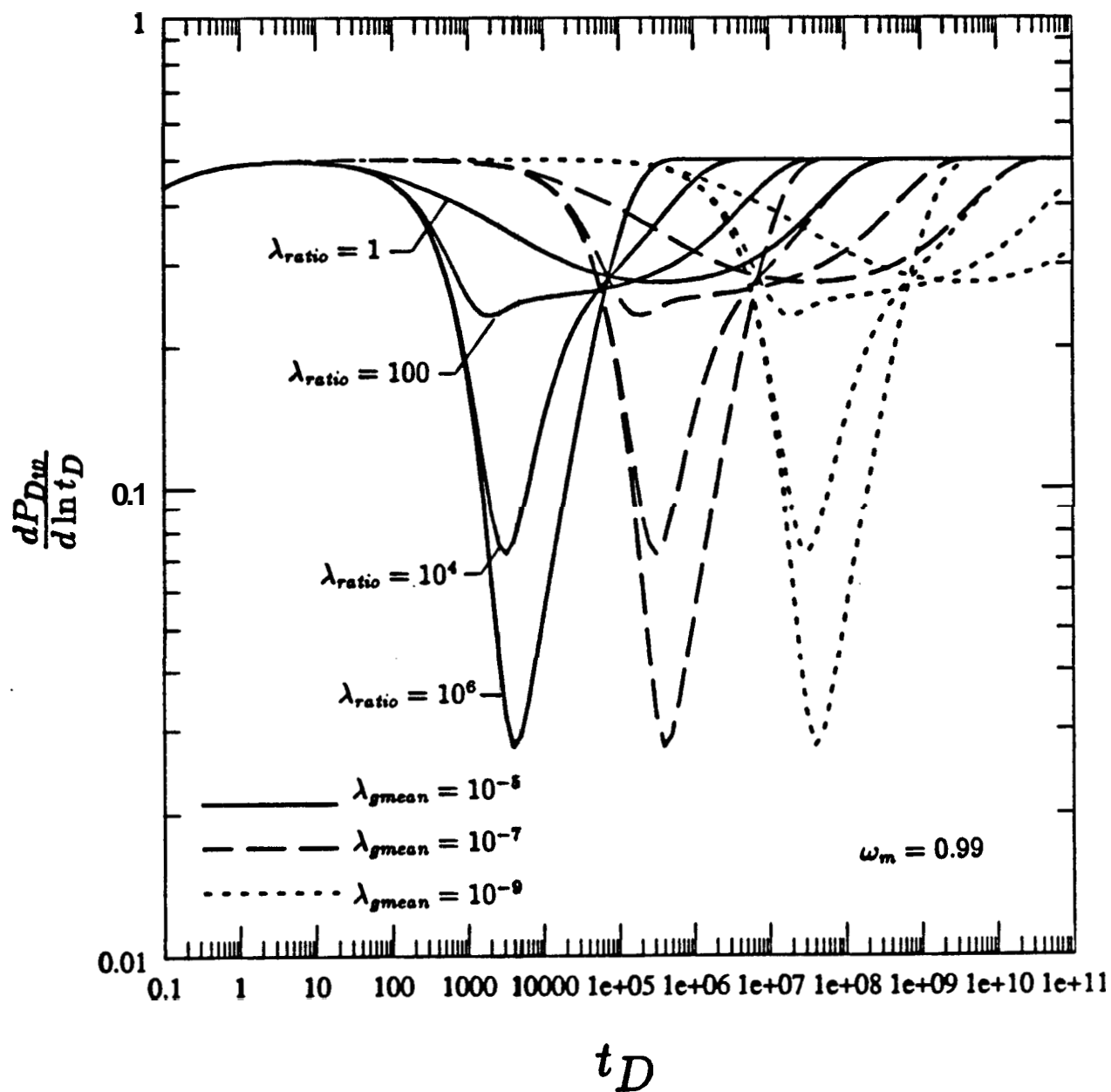


Figure 3.19: Pressure derivative profile for various λ_{gmean}

Section 4

Conclusion

1. The distributed formulation of interporosity flow predicts a gradual decline of the flowrate unlike the classical double porosity models which predict a sudden rate decline followed by a period of constant flowrate. This gradualness is more pronounced for larger matrix block size variabilities, i.e., cases of extremely nonuniform fracturing. Also, the gradualness is more apparent in the unsteady state model of interporosity flow.
2. Matrix block geometry does not have a significant effect on the rate or the pressure response. Spherical geometry yields slightly higher flowrate than the cylindrical geometry. The latter yields slightly higher flowrate than the slab geometry. Matrix block geometry only enters the formulation for unsteady state interporosity flow and not for pseudo-steady state interporosity flow.
3. The unsteady state formulation of interporosity flow yields higher flowrate (and hence higher cumulative recovery) than the pseudo-steady state formulation.
4. The negatively skewed linear distribution of matrix block size yields higher flowrate than the uniform distribution. The latter, in turn, yields higher flowrate than the positively skewed linear distribution. Thus, from the viewpoint of reservoir producibility, it is more advantageous to have a high frequency of small blocks than large blocks.

- 5. Reservoir producibility is directly proportional to fracture intensity and inversely proportional to the degree of fracture nonuniformity. Hence, the Warren and Root model which assumes perfectly uniform fracturing, yields an upper bound of reservoir producibility.**

Nomenclature

A	=	area of matrix fracture interface, ft^2
A_1	=	constant
A_2	=	constant
A_3	=	constant
B	=	formation volume factor, RB/STB
B_1	=	constant
B_2	=	constant
B_3	=	constant
c_f	=	fracture total compressibility, psi^{-1}
c_m	=	matrix total Compressibility, psi^{-1}
C_D	=	wellbore storage coefficient, dimensionless
C_1	=	constant
D_1	=	constant
E_1	=	constant
E_2	=	constant
E_3	=	constant
F_1	=	constant
G_1	=	constant
$g(s)$	=	a parameter in the Bessel function argument
h	=	matrix block size for slab geometry, ft
h_D	=	matrix block size for slab geometry, dimensionless

h_{max}	=	maximum matrix block size, ft
h_{min}	=	minimum matrix block size, ft
$I_0(x)$	=	modified Bessel function, first kind, zero order
$I_1(x)$	=	modified Bessel function, first kind, first order
k_f	=	fracture permeability, md
k_m	=	matrix permeability, md
$K_0(x)$	=	modified Bessel function, second kind, zero order
$K_1(x)$	=	modified Bessel function, second kind, first order
P_{Df}	=	fracture pressure, dimensionless
\bar{P}_{Df}	=	Laplace transformed fracture pressure
P_{Dm}	=	matrix pressure, dimensionless
\bar{P}_{Dm}	=	Laplace transformed matrix pressure
P_{Dw}	=	wellbore pressure response, dimensionless
\bar{P}_{Dw}	=	Laplace transformed wellbore pressure response
P_f	=	fracture fluid pressure, psi
$P(h)$	=	probability density function of matrix block size distribution
$P(h_D)$	=	probability density function of normalized matrix block size distribution in slab geometry
$P(r_{mD})$	=	probability density function of normalized matrix block size distribution in cylindrical and spherical geometries
P_i	=	initial pressure, psi
P_m	=	matrix fluid pressure, psi
P_w	=	wellbore pressure response, psi
q	=	volumetric flow rate, STB/D
q_D	=	volumetric flow rate, dimensionless
\bar{q}_D	=	Laplace transformed flow rate
Q_D	=	cumulative production, dimensionless

\bar{Q}_D	=	Laplace transformed cumulative production
r	=	radial coordinate, ft
r_D	=	radial Coordinate, dimensionless
r_m	=	matrix block radius for cylindrical and spherical geometries, ft
r_{mD}	=	matrix block radius for cylindrical and spherical geometries, dimensionless
r_{mmax}	=	maximum matrix block radius, ft
r_{mmin}	=	minimum matrix block radius, ft
r_w	=	wellbore radius, ft
s	=	Laplace parameter
S_D	=	skin factor, dimensionless
t	=	time, hour
t_D	=	time, dimensionless
t_{Dp}	=	production time, dimensionless
$U(h)$	=	interporosity flow contribution from matrix size h
$U(h_D)$	=	interporosity flow contribution from dimensionless matrix size h_D
$U(r_{mD})$	=	interporosity flow contribution from dimensionless matrix radius r_{mD}
V	=	volume of matrix block, ft ³
x	=	Bessel function argument
z	=	coordinate for slab matrix block
z_D	=	dimensionless coordinate for slab matrix block
α	=	constant in the definition of interporosity flow coefficient
η	=	coordinate for matrix block radius
η_D	=	dimensionless coordinate for matrix block radius
λ	=	interporosity flow coefficient, dimensionless

- λ_{gmean} = geometrical mean of interporosity flow coefficient, dimensionless
- λ_{max} = maximum interporosity flow coefficient, dimensionless
- λ_{min} = minimum interporosity flow coefficient, dimensionless
- λ_{ratio} = ratio of λ_{max} to λ_{min}
- μ = viscosity, cp
- ϕ_f = fracture porosity, dimensionless
- ϕ_m = matrix porosity, dimensionless
- ω_f = fracture storativity ratio, dimensionless
- ω_m = matrix storativity ratio, dimensionless

SI METRIC CONVERSION FACTORS

- bbf x 1.589873 E-01 = m³
- cp x 1.0* E-03 = pa.s
- ft x 3.048' E-01 = m
- psi x 6.894757 E-01 = kpa
- psi⁻¹ x 1.450 E-01 = kpa⁻¹

* Conversion factor is exact.

Bibliography

- [1] Belani, A.K. and Jalali-Yazdi, **Y.**: “Estimation of Matrix Block Size Distribution in Naturally Fractured Reservoirs,” **paper** SPE 18171 presented at the SPE **63rd** Annual Technical Conference and Exhibition, Houston, Texas, Oct. **2-5, 1988.**
- [2] Barenblatt, G.E., Zheltov, **I.P.**, and Kochina, I.N.: “Basic Concepts in the Theory of Homogeneous Liquids in Fissured Rocks,” *J. Appl. Math. Mech.* **24,** (1960) **1286 - 1303.**
- [3] Barenblatt, G.E.: “On Certain Boundary-Value Problems for the Equations of Seepage of a Liquid in Fissured Rocks,” *J. Appl. Math. Mech.* **27,** (1963) **513 - 518.**
- [4] Warren, J.E and Root, **P.J.**: “The Behavior of Naturally Fractured Reservoirs,” *Soc. Pet. Eng. J.* (Sept. **1963**) **245 - 255.** *Trans. , AIME,* **228**
- [5] Odeh, **A.S.**: “Unsteady-State Behavior of Naturally Fractured Reservoirs,” *Soc. Pet. Eng. J.* (Mar. **1965**) **60 - 64.**
- [6] Kazemi, H.: “Pressure Transient Analysis of Naturally Fractured Reservoirs with Uniform Fracture Distribution,” *Soc. Pet. Eng. J.* (Dec. **1969**) **451 - 462.**
- [7] Kazemi, H., Seth, **M.S.**, and Thomas, G.W.: “The Interpretation of Interference Tests in Naturally Fractured Reservoirs with Uniform Fracture Distribution,” *Soc. Pet. Eng. J.* (Dec. **1969**) **463 - 472.**

- [8] DeSwaan, O.A.: "Analytic Solution for Determining Naturally Fractured Reservoir Properties by Well Testing," *Soc.Pet.Eng.J.* (June 1976) 117 - 122.
- [9] Najurieta, H.L.: "A Theory for Pressure Transient Analysis in Naturally Fractured Reservoirs," *J.Pet.Tech.* (July 1980) 1241 - 1250.
- [10] Mavor, M.J. and Cinco-Ley, H.: "Transient Pressure Behavior of Naturally Fractured Reservoirs," paper SPE 7977 presented at the SPE California Regional Meeting, Ventura, California, Apr. 18-20, 1979.
- [11] Kucuk, F. and Sawyer, W.K.: "Transient Flow in Naturally Fractured Reservoirs and Its Application to Devonian Gas Shales," paper SPE 9397 presented at the SPE 55th Annual Technical Conference and Exhibition, Dallas, Texas, Sept. 21-24, 1980.
- [12] Deruyck, B.G., Bourdet, D.P., DaPrat, G., and Ramey, H.J.Jr.: "Interpretation of Interference Tests in Reservoirs With Double Porosity Behavior - Theory and Field **Examples**," paper SPE 11025 presented at the SPE 57th Annual Technical Conference and Exhibition, New Orleans, Louisiana, Sept. 26-29, 1982.
- [13] Ohaeri, C.U.: "Pressure Buildup Analysis for a Well Produced at a Constant Pressure in a Naturally Fractured Reservoir;" paper SPE 12009 presented at the SPE 58th Annual Technical Conference and Exhibition, San Francisco, California, Oct. 5-8, 1983.
- [14] Streltsova, T.D.: "Well Pressure Behavior of a Naturally Fractured Reservoir," *Soc.Pet.Eng.J.* (Oct. 1983) 769 - 780.
- [15] Cinco-Ley, H., Samaniego, F.V., and Kucuk, F.: "The Pressure Transient Behavior for Naturally Fractured Reservoirs With Multiple Block **Size**," paper SPE 14168 presented at the SPE 60th Annual Technical Conference and **Ex**hibition, Las Vegas, Nevada, Sept. 22-25, 1985.

- [16] Jalali-Yazdi, Y. and Ershaghi, I.: "A Unified Type Curve Approach for Pressure Transient Analysis of Naturally Fractured Reservoirs," paper SPE 16778 presented at the SPE 62nd Annual Technical Conference and Exhibition, Dallas, Texas, Sept. 27-30, 1987,
- [17] DaPrat, G., Cinco-Ley, H., and Ramey, H.J.Jr.: "Decline Curve Analysis Using Type Curves for Two-Porosity System," *Soc.Pet.Eng.J.* (June 1981) 354 - 362.
- [18] Fetkovich, M.J.: "Decline Curve Analysis Using Type Curves," *J.Pet.Tech.* (June 1980) 1065 - 1077.
- [19] Raghavan, R. and Ohaeri, C.U.: "Unsteady Flow to A Well Produced at Constant Pressure in a Fractured Reservoir," paper SPE 9902 presented at the SPE California Regional Meeting, Bakersfield, California, Mar. 25-26, 1981.
- [20] Sageev, A., DaPrat, G., and Ramey, H.J.Jr.: "Decline Curve Analysis for Double-Porosity Systems," paper SPE 13630 presented at the SPE California Regional Meeting, Bakersfield, California, Mar. 27-29, 1985.
- [21] Jalali-Yazdi, Y., Belani, A.K., and Fujiwara, K.: "An Interporosity Flow Model for Naturally Fractured Reservoirs," paper SPE 18749 presented at the SPE California Regional Meeting, Bakersfield, California, Apr. 5-7, 1989.
- [22] van Everdingen, A.F. and Burst, W.: "The Application of the Laplace Transformation to Flow Problems in Reservoirs," *Trans. , AIME(1949)*, 186 305 - 324B.
- [23] Stehfest, H.: "Algorithm 368, Numerical Inversion of Laplace Transforms," *Communications of the ACM D-5* (Jan. 1970) 13, No.1, 47 - 49.

Appendix A

Derivation of Solution

This appendix contains a derivation of the constant rate and constant pressure solutions.

A.1 Slab Geometry

A.1.1 Unsteady State

In dimensionless form, Eqns. 2.1 through 2.3 become:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \frac{\mu}{k_f} \left(-\frac{2\pi k_f h_f r_w^2}{q\mu B} \right) \int_{h_{max}}^{h_{min}} U(h_D) P(h_D) dh_D, \quad (\text{A.1})$$

$$\frac{\partial^2 P_{Dm}}{\partial z_D^2} = \frac{3\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D}, \quad (\text{A.2})$$

$$U(h_D) = -\frac{k_m}{\mu h^2} \left(-\frac{q\mu B}{2\pi k_f h_f} \right) \frac{\partial P_{Dm}}{\partial z_D} \Big|_{z_D=0}. \quad (\text{A.3})$$

Combining Eqns. A.1 and A.3 yields:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} - \int_{h_{max}}^{h_{min}} \frac{\lambda}{3} \frac{\partial P_{Dm}}{\partial z_D} \Big|_{z_D=0} P(h_D) dh_D. \quad (\text{A.4})$$

In dimensionless form, Eqns. 2.4 through 2.7 are:

$$P_{Df} = P_{Dm} = 0 \quad ; \quad \text{at } t_D = 0, \quad (\text{A.5})$$

$$C_D \frac{\partial P_{Dw}}{\partial t_D} - \frac{\partial P_{Df}}{\partial r_D} = 1 ; \text{ at } r_D = 1, \quad (\text{A.6})$$

$$P_{Dw} = P_{Df} - S_D \frac{\partial P_{Df}}{\partial r_D} ; \text{ at } r_D = 1, \quad (\text{A.7})$$

$$P_{Df} = 0 ; \text{ as } r_D \rightarrow \infty. \quad (\text{A.8})$$

Matrix boundary conditions are:

$$P_{Dm} = P_{Df} ; \text{ at } z_D = 0, \quad (\text{A.9})$$

and

$$\frac{\partial P_{Dm}}{\partial z_D} = 0 ; \text{ at } z_D = 1. \quad (\text{A.10})$$

Eqns. A.2 and A.4, subject to the six conditions, Eqns. A.5 through A.10 are to be solved. Applying Laplace transformation to these equations yields:

$$\frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial \bar{P}_{Df}}{\partial r_D} = \omega_f s \bar{P}_{Df} - \int_{h_{\min}}^1 \frac{\lambda}{3} \frac{\partial \bar{P}_{Dm}}{\partial z_D} \Big|_{z_D=0} P(h_D) dh_D, \quad (\text{A.11})$$

$$\frac{\partial^2 \bar{P}_{Dm}}{\partial z_D^2} = \frac{3\omega_m s}{\lambda} \bar{P}_{Dm}, \quad (\text{A.12})$$

$$\bar{P}_{Df} = \bar{P}_{Dm} = 0 ; \text{ at } t_D = 0, \quad (\text{A.13})$$

$$C_D s \bar{P}_{Dw} - \frac{\partial \bar{P}_{Df}}{\partial r_D} = \frac{1}{s} ; \text{ at } r_D = 1, \quad (\text{A.14})$$

$$\bar{P}_{Dw} = \bar{P}_{Df} - S_D \frac{\partial \bar{P}_{Df}}{\partial r_D} ; \text{ at } r_D = 1, \quad (\text{A.15})$$

$$\bar{P}_{Df} = 0 ; \text{ as } r_D \rightarrow \infty, \quad (\text{A.16})$$

$$\bar{P}_{Dm} = \bar{P}_{Df} ; \text{ at } z_D = 0, \quad (\text{A.17})$$

$$\frac{\partial \bar{P}_{Dm}}{\partial z_D} = 0 ; \text{ at } z_D = 1. \quad (\text{A.18})$$

The general solution of Eqn. A.12 is

$$\bar{P}_{Dm} = A_1 \cosh \left(\sqrt{\frac{3\omega_m s}{\lambda}} z_D \right) + B_1 \sinh \left(\sqrt{\frac{3\omega_m s}{\lambda}} z_D \right). \quad (\text{A.19})$$

Applying Eqn. A.17:

$$A_1 = \bar{P}_{Dm} = \bar{P}_{Df}. \quad (\text{A.20})$$

Next, differentiating Eqn. A.19 and using Eqn. A.18:

$$B_1 = -\tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) \bar{P}_{Df}. \quad (\text{A.21})$$

Thus:

$$\frac{\partial \bar{P}_{Dm}}{\partial z_D} \Big|_{z_D=0} = -\sqrt{\frac{3\omega_m s}{\lambda}} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) \bar{P}_{Df}. \quad (\text{A.22})$$

Substituting Eqn. A.22 into Eqn. A.11, and rearranging yields:

$$r_D^2 \frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + r_D \frac{\partial \bar{P}_{Df}}{\partial r_D} - r_D^2 s g(s) \bar{P}_{Df} = 0, \quad (\text{A.23})$$

where:

$$g(s) = \omega_f + \omega_m \int_{h_{\max}}^1 \sqrt{\frac{\lambda}{3\omega_m s}} \tanh\left(\sqrt{\frac{3\omega_m s}{\lambda}}\right) P(h_D) dh_D. \quad (\text{A.24})$$

The general solution for Eqn. A.23 is:

$$\bar{P}_{Df} = C_1 K_0(\sqrt{sg(s)r_D}) + D_1 I_0(\sqrt{sg(s)r_D}). \quad (\text{A.25})$$

Applying boundary condition, Eqn. A.16:

$$\bar{P}_{Df} = C_1 K_0(\sqrt{sg(s)r_D}). \quad (\text{A.26})$$

Differentiating Eqn. A.26:

$$\frac{\partial \bar{P}_{Df}}{\partial r_D} = -C_1 \sqrt{sg(s)} K_1(\sqrt{sg(s)r_D}). \quad (\text{A.27})$$

From the skin and storage conditions, Eqns. A.14 and A.15:

$$C_1 = \frac{r}{s [C_D s \{K_0(\sqrt{sg(s)}) + S_D \sqrt{sg(s)} K_1(\sqrt{sg(s)})\} + \sqrt{sg(s)} K_1(\sqrt{sg(s)})]}. \quad (\text{A.28})$$

Then, from Eqns. A.26 and A.28, \bar{P}_{Df} is obtained as:

$$\bar{P}_{Df} = \frac{K_0(xr_D)}{s [C_D s \{K_0(x) + S_D x K_1(x)\} + x K_1(x)]}, \quad (\text{A.29})$$

where:

$$x = \sqrt{sg(s)}, \quad (\text{A.30})$$

and:

$$\bar{P}_{Dw} = \frac{K_0(x) + S_D x K_1(x)}{s [C_D s \{K_0(x) + S_D x K_1(x)\} + x K_1(x)]}. \quad (\text{A.31})$$

The relation between flowrate and wellbore pressure in Laplace space is [22]:

$$\bar{q}_D = \frac{1}{s^2 \bar{P}_{Dw}}. \quad (\text{A.32})$$

Then, flowrate for no storage case is:

$$\bar{q}_D = \frac{x K_1(x)}{s \{K_0(x) + S_D x K_1(x)\}}. \quad (\text{A.33})$$

Cumulative production is given as:

$$\bar{Q}_D = \frac{x K_1(x)}{s^2 \{K_0(x) + S_D x K_1(x)\}}. \quad (\text{A.34})$$

A.1.2 Pseudo-Steady State

Solution for pseudo-steady state interporosity flow is obtained by eliminating the spacial dependency of time rate of change of pressure. That is, **Eqn. A.2** becomes:

$$\frac{\partial^2 P_{Dm}}{\partial z_D^2} = \frac{3\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D} \equiv E_1(t_D), \quad (\text{A.35})$$

where $E_1(t_D)$ is independent of z_D . This partial differential equation is solved as:

$$\frac{\partial P_{Dm}}{\partial z_D} = E_1(t_D) z_D + F_1, \quad (\text{A.36})$$

$$P_{Dm} = \frac{1}{2} E_1(t_D) z_D^2 + F_1 z_D + G_1. \quad (\text{A.37})$$

Applying boundary condition, **Eqns. A.9**, and **A.10**:

$$F_1 = -E_1(t_D), \quad (\text{A.38})$$

$$G_1 = P_{Df}. \quad (\text{A.39})$$

Then:

$$P_{Dm} = \frac{1}{2} E_1(t_D) z_D^2 - E_1(t_D) z_D + P_{Df}. \quad (\text{A.40})$$

By differentiating Eqn. A.40 and using Eqn. A.35:

$$\frac{\partial P_{Dm}}{\partial z_D} \Big|_{z_D=0} = -E_1(t_D) = -\frac{3\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D}. \quad (\text{A.41})$$

Substituting Eqn. A.41 into Eqn. A.4, the diffusivity equation becomes:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \omega_m \int_{h_{max}}^1 \frac{\partial P_{Dm}}{\partial t_D} P(h_D) dh_D. \quad (\text{A.42})$$

In Eqn. A.40, by averaging $E_1(t_D)$ across the matrix block thickness, the following equation is obtained:

$$\begin{aligned} P_{Dm} - P_{Df} &= \left(\frac{1}{2} z_D^2 - z_D \right) E_1(t_D) \\ &= -\frac{1}{3} E_1(t_D) |_{avg}. \end{aligned} \quad (\text{A.43})$$

Then:

$$E_1(t_D) |_{avg} = 3(P_{Df} - P_{Dm}). \quad (\text{A.44})$$

Eqns. A.44 and A.35 give:

$$\frac{\partial P_{Dm}}{\partial t_D} = \frac{\lambda}{\omega_m} (P_{Df} - P_{Dm}). \quad (\text{A.45})$$

Applying Laplace transformation to Eqns. A.42 and A.45:

$$\frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial \bar{P}_{Df}}{\partial r_D} = \omega_f s \bar{P}_{Df} + \int_{h_{max}}^1 s \bar{P}_{Dm} P(h_D) dh_D, \quad (\text{A.46})$$

and

$$s \bar{P}_{Dm} = \frac{\lambda}{\omega_m} (\bar{P}_{Df} - \bar{P}_{Dm}). \quad (\text{A.47})$$

From Eqn. A.47:

$$\bar{P}_{Dm} = \frac{\lambda}{\omega_m s + \lambda} \bar{P}_{Df}. \quad (\text{A.48})$$

Substituting Eqn. A.48 into Eqn. A.46, the following equation is obtained:

$$r_D^2 \frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + r_D \frac{\partial \bar{P}_{Df}}{\partial r_D} - r_D^2 s g(s) \bar{P}_{Df} = 0, \quad (\text{A.49})$$

where:

$$g(s) = \omega_f + \omega_m \int_{h_{max}}^1 \frac{\lambda}{\omega_m s + \lambda} P(h_D) dh_D. \quad (\text{A.50})$$

Eqn. A.49 is the same as Eqn. A.23 in the unsteady state model, which yields the same general solution. The procedure and solutions of Eqns. A.25 through A.34 are valid for the pseudo-steady state model also.

A.2 Cylindrical Geometry

A.2.1 Unsteady State

In dimensionless form, Eqns. 2.1 through 2.3 become:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \frac{\mu}{k_f} \left(-\frac{2\pi k_f h_f r_w^2}{q\mu B} \right) \int_{r_{m\max}}^1 U(r_{mD}) P(r_{mD}) dr_{mD}, \quad (\text{A.51})$$

$$\frac{\partial^2 P_{Dm}}{\partial \eta_D^2} + \frac{1}{\eta_D} \frac{\partial P_{Dm}}{\partial \eta_D} = \frac{6\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D}, \quad (\text{A.52})$$

$$U(r_{mD}) = \frac{2k_m}{\mu r_m^2} \left(-\frac{q\mu B}{2\pi k_f h_f} \right) \frac{\partial P_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1}. \quad (\text{A.53})$$

Combining Eqn. A.51 and Eqn. A.53 yields:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \int_{r_{m\max}}^1 \frac{\lambda}{3} \frac{\partial P_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} P(r_{mD}) dr_{mD}. \quad (\text{A.54})$$

Eqns. A.5 through A.8 are used as initial and boundary conditions in cylindrical geometry, too. As for the boundary conditions for the matrix coordinate, the following are used instead of Eqns. A.9 and A.10:

$$P_{Dm} = P_{Df} ; \text{ at } \eta_D = 1, \quad (\text{A.55})$$

$$P_{Dm} \text{ is finite at } \eta_D = 0. \quad (\text{A.56})$$

Applying Laplace transform to Eqns. A.54 and A.52:

$$\frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial \bar{P}_{Df}}{\partial r_D} = \omega_f s \bar{P}_{Df} + \int_{r_{m\max}}^1 \frac{\lambda}{3} \frac{\partial \bar{P}_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} P(r_{mD}) dr_{mD}, \quad (\text{A.57})$$

and

$$\frac{\partial^2 \bar{P}_{Dm}}{\partial \eta_D^2} + \frac{1}{\eta_D} \frac{\partial \bar{P}_{Dm}}{\partial \eta_D} = \frac{6\omega_m}{\lambda} s \bar{P}_{Dm}. \quad (\text{A.58})$$

The boundary conditions in Laplace space are given by Eqns. A.13 through A.16 and the following two equations:

$$\bar{P}_{Dm} = \bar{P}_{Df} ; \text{ at } \eta_D = 1, \quad (\text{A.59})$$

$$\bar{P}_{Dm} \text{ is finite at } \eta_D = 0. \quad (\text{A.60})$$

The general solution for Eqn. A.58 is given as:

$$\bar{P}_{Dm} = A_2 K_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \eta_D \right) + B_2 I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \eta_D \right). \quad (\text{A.61})$$

From boundary condition, Eqn. A.60, \bar{P}_{Dm} is finite at $\eta_D = 0$, therefore:

$$A_2 = 0. \quad (\text{A.62})$$

From the other boundary condition, Eqn. A.59:

$$B_2 = \frac{\bar{P}_{Df}}{I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)}. \quad (\text{A.63})$$

Then:

$$\bar{P}_{Dm} = \frac{I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \eta_D \right)}{I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)} \bar{P}_{Df}. \quad (\text{A.64})$$

By differentiating Eqn. A.64 and evaluating at $\eta_D = 1$:

$$\frac{\partial \bar{P}_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} = \frac{\sqrt{\frac{6\omega_m s}{\lambda}} I_1 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)}{I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)} \bar{P}_{Df}. \quad (\text{A.65})$$

Substituting Eqn. A.65 into Eqn. A.57:

$$r_D^2 \frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + r_D \frac{\partial \bar{P}_{Df}}{\partial r_D} - r_D^2 s g(s) \bar{P}_{Df} = 0, \quad (\text{A.66})$$

where:

$$g(s) = \omega_f + \omega_m \int_{\frac{r_{mmin}}{r_{mmax}}}^1 \sqrt{\frac{2\lambda}{3\omega_m s}} \frac{I_1 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)}{I_0 \left(\sqrt{\frac{6\omega_m s}{\lambda}} \right)} P(r_{mD}) dr_{mD}. \quad (\text{A.67})$$

Eqn. A.65 is the same as Eqn. A.23 in the slab geometry, so solutions shown by Eqns. A.29 through A.34 are valid for this geometry too, while $g(s)$ is different.

A.2.2 Pseudo-Steady State

The procedure is exactly the same as the case of slab geometry. Eqn. A.52 is assumed to be independent of space, which yields:

$$\frac{\partial^2 P_{Dm}}{\partial \eta_D^2} + \frac{1}{\eta_D} \frac{\partial P_{Dm}}{\partial \eta_D} = \frac{6\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D} \equiv E_2(t_D), \quad (\text{A.68})$$

where $E_2(t_D)$ is independent of η_D . Solving this partial differential equation yields:

$$P_{Dm} = \frac{1}{4}E_2(t_D)(\eta_D^2 - 1) + P_{Df}. \quad (\text{A.69})$$

Differentiating, evaluating at $\eta_D = 1$, and considering Eqn. A.68:

$$\left. \frac{\partial P_{Dm}}{\partial \eta_D} \right|_{\eta_D=1} = \frac{3\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D}. \quad (\text{A.70})$$

Substituting Eqn. A.70 into Eqn. A.54, the diffusivity equation becomes:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \omega_m \int_{r_{mmin}}^1 \frac{\partial P_{Dm}}{\partial t_D} P(r_{mD}) dr_{mD}. \quad (\text{A.71})$$

In Eqn. A.69, averaging $E_2(t_D)$ across the matrix radius:

$$\begin{aligned} P_{Dm} - P_{Df} &= \frac{1}{4}(\eta_D^2 - 1) E_2(t_D) \\ &= -\frac{1}{6} E_2(t_D)|_{avg}. \end{aligned} \quad (\text{A.72})$$

Then,

$$E_2(t_D)|_{avg} = 6(P_{Df} - P_{Dm}). \quad (\text{A.73})$$

Eqns. A.68 and A.73 give:

$$\frac{\partial P_{Dm}}{\partial t_D} = \frac{\lambda}{\omega_m} (P_{Df} - P_{Dm}). \quad (\text{A.74})$$

This equation is the same as Eqn. A.45 and fracture flow equation, Eqn. A.71, is also the same as Eqn. A.42 for the slab geometry. Then, the procedure to obtain the function $g(s)$ follows that of Eqns. A.46 through A.50. The function $g(s)$ obtained is the same as that of the slab geometry (except r_m should be used in these equations instead of h).

A.3 Spherical Geometry

A.3.1 Unsteady State

In this case, the fracture flow equation, Eqn. A.51, is valid. Matrix flow equation, Eqn. 2.2, becomes:

$$\frac{\partial^2 P_{Dm}}{\partial \eta_D^2} + \frac{2}{\eta_D} \frac{\partial P_{Dm}}{\partial \eta_D} = \frac{9\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D}, \quad (\text{A.75})$$

and Eqn. 2.3 becomes:

$$U(r_{mD}) = \frac{3k_m}{\mu r_m^2} \left(-\frac{q\mu B}{2\pi k_f h_f} \right) \frac{\partial P_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1}. \quad (\text{A.76})$$

Combining Eqn. A.51 and Eqn. A.76 yields:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \int_{r_{mmin}}^1 \frac{\lambda}{3} \frac{\partial P_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} P(r_{mD}) dr_{mD}. \quad (\text{A.77})$$

Initial and boundary conditions are the same as the case of cylindrical geometry. Eqns. A.5 through A.8 and Eqns. A.55 and A.56 are used. Applying Laplace transform to Eqns. A.77 and A.75:

$$\frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial \bar{P}_{Df}}{\partial r_D} = \omega_f s \bar{P}_{Df} + \int_{r_{mmin}}^1 \frac{\lambda}{3} \frac{\partial \bar{P}_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} P(r_{mD}) dr_{mD}, \quad (\text{A.78})$$

and

$$\frac{\partial^2 \bar{P}_{Dm}}{\partial \eta_D^2} + \frac{2}{\eta_D} \frac{\partial \bar{P}_{Dm}}{\partial \eta_D} = \frac{9\omega_m s}{\lambda} \bar{P}_{Df}. \quad (\text{A.79})$$

The conditions in Laplace space are shown in Eqns. A.13 through A.16 and Eqns. A.59 and A.60.

The general solution for Eqn. A.79 is given as:

$$\bar{P}_{Dm} = \frac{1}{\eta_D} \left\{ A_3 \cosh \left(\sqrt{\frac{9\omega_m s}{\lambda}} \eta_D \right) + B_3 \sinh \left(\sqrt{\frac{9\omega_m s}{\lambda}} \eta_D \right) \right\}. \quad (\text{A.80})$$

From boundary condition, Eqn. A.60, \bar{P}_{Dm} is finite at $\eta_D = 0$, therefore:

$$A_3 = 0. \quad (\text{A.81})$$

From another boundary condition, Eqn. A.59:

$$B_3 = \frac{\bar{P}_{Df}}{\sinh \left(\sqrt{\frac{9\omega_m s}{\lambda}} \right)}. \quad (\text{A.82})$$

Then:

$$\bar{P}_{Dm} = \frac{1}{\eta_D} \frac{\sinh \left(\sqrt{\frac{9\omega_m s}{\lambda}} \eta_D \right)}{\sinh \left(\sqrt{\frac{9\omega_m s}{\lambda}} \right)} \bar{P}_{Df}. \quad (\text{A.83})$$

By differentiating Eqn. A.83 and evaluating at $\eta_D = 1$:

$$\frac{\partial \bar{P}_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} = \left\{ \sqrt{\frac{9\omega_m s}{\lambda}} \coth \left(\sqrt{\frac{9\omega_m s}{\lambda}} \right) - 1 \right\} \bar{P}_{Df}. \quad (\text{A.84})$$

Substituting Eqn. A.84 into Eqn. A.78:

$$r_D^2 \frac{\partial^2 \bar{P}_{Df}}{\partial r_D^2} + r_D \frac{\partial \bar{P}_{Df}}{\partial r_D} - r_D^2 s g(s) \bar{P}_{Df} = 0, \quad (\text{A.85})$$

where:

$$g(s) = \omega_f \int_{\frac{r_{mmin}}{r_{mmax}}}^1 \frac{\lambda}{3s} \left\{ \sqrt{\frac{9\omega_m s}{\lambda}} \coth \left(\sqrt{\frac{9\omega_m s}{\lambda}} \right) - 1 \right\} P(r_{mD}) dr_{mD}. \quad (\text{A.86})$$

Eqn. A.85 is the same as Eqn. A.25 for the slab geometry, so solutions obtained by Eqns. A.29 through A.34 are also valid for this geometry, while $g(s)$ is different.

A.3.2 Pseudo-Steady State

The procedure is exactly the same as the case of slab geometry. Eqn. A.75 is assumed to be independent of space, which yields:

$$\frac{\partial^2 P_{Dm}}{\partial \eta_D^2} + \frac{2}{\eta_D} \frac{\partial P_{Dm}}{\partial \eta_D} = \frac{9\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D} \equiv E_3(t_D), \quad (\text{A.87})$$

where $E_3(t_D)$ is independent of η_D . Solving this partial differential equation gives:

$$P_{Dm} = \frac{1}{6} E_3(t_D) (\eta_D^2 - 1) + P_{Df}. \quad (\text{A.88})$$

Differentiating Eqn. A.88, evaluating at $\eta_D = 1$, and using Eqn. A.87:

$$\frac{\partial P_{Dm}}{\partial \eta_D} \Big|_{\eta_D=1} = \frac{3\omega_m}{\lambda} \frac{\partial P_{Dm}}{\partial t_D} \quad (\text{A.89})$$

Substituting Eqn. A.89 into Eqn. A.77, the diffusivity equation becomes:

$$\frac{\partial^2 P_{Df}}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial P_{Df}}{\partial r_D} = \omega_f \frac{\partial P_{Df}}{\partial t_D} + \omega_m \int_{\frac{r_{mmin}}{r_{mmax}}}^1 \frac{\partial P_{Dm}}{\partial t_D} P(r_{mD}) dr_{mD}. \quad (\text{A.90})$$

In Eqn. A.88, averaging $E_3(t_D)$ across the matrix radius:

$$\begin{aligned} P_{Dm} - P_{Df} &= \frac{1}{6} (\eta_D^2 - 1) E_3(t_D) \\ &= -\frac{1}{9} E_3(t_D) |_{avg}. \end{aligned} \quad (\text{A.91})$$

Then:

$$E_3(t_D)|_{avg} = 9(P_{Df} - P_{Dm}). \quad (\text{A.92})$$

Eqns. A.87 and A.92 give:

$$\frac{\partial P_{Dm}}{\partial t_D} = \frac{\lambda}{\omega_m} (P_{Df} - P_{Dm}). \quad (\text{A.93})$$

This equation is the same as Eqn. A.45 and the fracture flow equation, Eqn. A.90, is also the same as Eqn. A.42 for the slab and the cylindrical cases. Then, the procedure for obtaining the function $g(s)$ follows that of Eqns. A.46 through A.50. The function $g(s)$ is the same as that for slab and cylindrical cases (except r_m should be used in these equations instead of h).

Appendix B

Computer Programs

This section contains the computer programs which are used in this study. To solve some of the equations, IMSL fortran routines were used.

```

C
C *****
C
C * Calculation of PD, qD, and QD
C
C * for Slab, Cylindrical, and Spherical Geometries
C
C * ----- P.S.S. & U.S.S. -----
C
C *
C * Katsunori Fujiwara Apr.29, 89
C *****
C
C implicit real*8 (a-h,o-z)
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C dimension td(121),pd(121),xlmd(10),ylmd(10),dpd(120),
C rate(121),cum(121)
C
C open (unit=3,file='pl.dat')
C open (unit=4,file='dp1.dat')
C open (unit=7,file='rate1.dat')
C open (unit=8,file='cum1.dat')
C
C rewind (unit=3)
C rewind (unit=4)
C rewind (unit=7)
C rewind (unit=8)
C
C n=10
C m=1
C
C read(5,*) sn
C read(5,*) cd
C read(5,*) omgm
C
C read(5,*) nxlmd
C do 60 i=1, nxlmd
C 60 read(5,*) xlmd(i)
C read(5,*) nylmd
C do 70 i=1, nylmd
C 70 read(5,*) ylmd(i)
C
C td(1)=0.1
C do 40 l=1, 120
C td(l+1)=(10.0**0.1)*td(l)
C 40 continue
C
C
C do 20 i=1, nxlmd
C xlamda=xlmd(i)
C
C do 30 k=1, nylmd
C ylamda=ylmd(k)
C
C write(3,*) 121
C write(4,*) 120
C write(7,*) 121
C write(8,*) 121
C
C do 50 l=1, 121
C call pwd(td(l), n, pd(l), rate(l), cum(l))
C write(3,*) td(l), pd(l)
C write(7,*) td(l), rate(l)
C write(8,*) td(l), cum(l)

```

```
SO continue
do 55 l=1,120
  dpd(l)=(pd(l+1)-pd(l))/(dlog(td(l+1))-dlog(td(l)))
  write(4,*) td(l),dpd(l)
55 continue
30 continue
20 continue
```

```
C
C
```

```
stop
end
```

```

C
C *****
C
C *      Calculation of PD, qD, and QD
C
C •      for Slab, Cylindrical, and Spherical Geometries
C
C •
C *      ----- P.S.S. -----
C
C *
C
C          Katsunori Fujiwara   Apr.29, 89
C *****
C
C function plap1(s)
C implicit real*8(a-h,o-z)
C double precision k0,k1,gg,xx
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C if(xlamda.eq. ylamda) then
C   gg=1.0-omgm+omgm*xlamda/(omgm*s+xlamda)
C else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
C & dsqrt(xlamda/ylamda))
C & (datan(dsqrt(ylamda/omgm/s))-datan(sqrt(xlamda/omgm/
C & s)))
C endif
C
C   xx=dsqrt(s*gg)
C   k0=dbsk0(xx)
C   k1=dbsk1(xx)
C   plap1=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C   return
C   end
C
C
C function plap2(s)
C implicit real*8(a-h,o-z)
C double precision k0,k1,gg,plap,xx
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C if(xlamda.eq. ylamda) then
C   gg=1.0-omgm+omgm*xlamda/(omgm*s+xlamda)
C else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
C & dsqrt(xlamda/ylamda))
C & (datan(dsqrt(ylamda/omgm/s))-datan(sqrt(xlamda/omgm/
C & s)))
C endif
C
C   xx=dsqrt(s*gg)
C   k0=dbsk0(xx)
C   k1=dbsk1(xx)
C   plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C   plap2=1./s/s/plap
C   return
C   end
C
C
C function plap3(s)
C implicit real*8(a-h,o-z)
C double precision k0,k1,gg,plap,xx

```

c
c

```
common m, /a/sn, cd, xlamda, ylamda, omgm
```

```
if(xlamda .eq. ylamda) then  
  gg=1.0-omgm+omgm*xlamda/(omgm*s+xlamda)  
else  
  gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-  
  & dsqrt(xlamda/ylamda))*  
  & (datan(dsqrt(ylamda/omgm/s))-datan(sqrt(xlamda/omgm/  
  & s)))  
endif
```

c

```
xx=dsqrt(s*gg)  
k0=dbsk0(xx)  
k1=dbsk1(xx)  
plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))  
plap3=1./s/s/s/plap  
return  
end
```



```

C
C
C *****
C
C • Calculation of PD, qD, and QD
C *
C * for Slab Geometry ----- U.S.S. -----
C *
C Katsunori Pujiwara
C *****
C
C function plap1(s,ans)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,xx
C common m, /a/sn, cd,xlamda,ylamda,omgm
C
C if (xlamda .ne. ylamda) then
C   gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans
C else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)*
& dtanh(dsqrt(3.*omgm*s/xlamda))
C endif
C
C ut=dsqrt(s*gg)
C k0=dbsk0(xx)
C k1=dbsk1(xx)
C plap1=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C return
C end
C
C
C function plap2(s,ans)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,plap,xx
C common m, /a/sn, cd,xlamda,ylamda,omgm
C
C if (xlamda .ne. ylamda) then
C   gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans
C else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)*
& dtanh(dsqrt(3.*omgm*s/xlamda))
C endif
C
C xx=dsqrt(s*gg)
C k0=dbsk0(xx)
C k1=dbsk1(xx)
C plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C plap2=1./s/s/plap
C return
C end
C
C function plap3(s,ans)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,plap,xx
C common m, /a/sn, cd,xlamda,ylamda,omgm
C

```

```

if (xlamda .ne. ylamda) then
  gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans
  else
  gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)*
& dtanh(dsqrt(3.*omgm*s/xlamda))
endif

```

```

c
ut=dsqrt(s*gg)
k0=dbsk0(xx)
k1=dbsk1(xx)
plap(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
plap3=1./s/s/s/plap
return
end

```

c Definition of function f which is used for integration

```

function f(x)
implicit real*8 (a-h,o-z)

```

```

c
f=dtanh(x)/x
return
end

```

```

C
C *****
C
C * Calculation of PD,qD, and QD
C *
C * for Cylindrical Geometry ----- U.S.S. -----
C
C *
C * Katsunori Fujiwara
C ***** ** **
C *****

function plap1(s,ans)
implicit real*8 (a-h,o-z)
double precision k0,kl,gg,xx
common m, /a/sn, cd, xlamda, ylamda, omgm

if(xlamda .ne. ylamda) then
  gg=1.0-omgm+omgm*dsqrt(2.*xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans
  else
  gg=1.0-omgm+omgm*dsqrt(2.*xlamda/3./omgm/s)*
& dbsile(dsqrt(6.*omgm*s/xlamda))/
& dbsi0e(dsqrt(6.*omgm*s/xlamda))
endif

xx=dsqrt(s*gg)
k0=dbsk0(xx)
kl=dbskl(xx)
plap1=(k0+sn*xx*kl)/(s*(cd*s*(k0+sn*xx*kl)+xx*kl))
return
end

C
C
C function plap2(s,ans)
implicit real*8 (a-h,o-z)
double precision k0,kl,gg,xx
common m, /a/sn, cd, xlamda, ylamda, omgm

if(xlamda .ne. ylamda) then
  gg=1.0-omgm+omgm*dsqrt(2.*xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans
  else
  gg=1.0-omgm+omgm*dsqrt(2.*xlamda/3./omgm/s)*
& dbsile(dsqrt(6.*omgm*s/xlamda))/
& dbsi0e(dsqrt(6.*omgm*s/xlamda))
endif

xx=dsqrt(s*gg)
k0=dbsk0(xx)
kl=dbskl(xx)
plap=(k0+sn*xx*kl)/(s*(cd*s*(k0+sn*xx*kl)+xx*kl))
plap2=1./s/s/plap
return
end

C
C
C function plap3(s,ans)
implicit real*8 (a-h,o-z)
double precision k0,kl,gg,xx
common m, /a/sn, cd, xlamda, ylamda, omgm

if(xlamda .ne. ylamda) then

```

```

    gg=1.0-omgm+omgm*dsqrt (2.*xlamda/3./omgm/s)/(1.0-
&   dsqrt(xlamda/ylamda))*ans
    else
    gg=1.0-omgm+omgm*dsqrt (2.*xlamda/3./omgm/s)*
&   dbsile (dsqrt(6.*omgm*s/xlamda))/
&   dbsi0e (dsqrt(6.*omgm*s/xlamda))
    endif

```

```

c
xx=dsqrt (s*gg)
k0=dbsk0 (xx)
k1=dbsk1 (xx)
plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
plap3=1./s/s/s/plap
return
end

```

```

c   Definition of function f which is used for integration
function f(x)
implicit real*8 (a-h,o-z)
double precision i0e,ile

```

```

c
i0e=dbsi0e (x)
ile=dbsile(x)
f=ile/i0e/x

```

```

c
return
end

```

```

C
C *****
C
C * Calculation of PD, qD, and QD
C *
C * for Spherical Geometry ----- U.S.S. -----
C *
C *
C * Katsunori Fujiwara
C *****
C
C function plap1(s,ans1,ans2)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,xx
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C if(xlamda .ne. ylamda) then
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
C & dsqrt(xlamda/ylamda))*ans1
C & -dsqrt(xlamda)/6./s/(1.0-dsqrt(xlamda/ylamda))*ans2
C   else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/dtanh
C & (dsqrt(9.*omgm*s/xlamda))-xlamda/3./s
C   endif
C
C   xx=dsqrt(s*gg)
C   k0=dbsk0(xx)
C   k1=dbsk1(xx)
C   plap1=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C   return
C   end
C
C function plap2(s,ans1,ans2)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,xx
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C if(xlamda .ne. ylamda) then
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
C & dsqrt(xlamda/ylamda))*ans1
C & -dsqrt(xlamda)/6./s/(1.0-dsqrt(xlamda/ylamda))*ans2
C   else
C   gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/dtanh
C & (dsqrt(9.*omgm*s/xlamda))-xlamda/3./s
C   endif
C
C   xx=dsqrt(s*gg)
C   k0=dbsk0(xx)
C   k1=dbsk1(xx)
C   plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
C   plap2=1./s/s/plap
C   return
C   end
C
C function plap3(s,ans1,ans2)
C implicit real*8 (a-h,o-z)
C double precision k0,k1,gg,xx
C common m, /a/sn, cd, xlamda, ylamda, omgm
C
C if(xlamda .ne. ylamda) then

```

```

    gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans1
& -dsqrt(xlamda)/6./s/(1.0-dsqrt(xlamda/ylamda))*ans2
    else
    gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/dtanh
& (dsqrt(9.*omgm*s/xlamda))-xlamda/3./s
    endif

```

```

c
    ut=dsqrt(s*gg)
    k0=dbsk0(xx)
    k1=dbsk1(xx)
    plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
    plap3=1./s/s/s/plap
    return
end

```

c Definition of function f which is used for integration

```

function f1(x)
implicit real*8(a-h,o-z)

```

```

c
    f1=1./dtanh(x)/x
    return
end

```

```

function f2(x)
implicit real*8(a-h,o-z)

```

```

c
    f2=1./dsqrt(x)
    return
end

```



```

gg=1.0-omgm+omgm*xlamda/(omgm*s+xlamda)
else
  if( nflag .eq. 1) then
    gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
&      dsqrt(xlamda/ylamda))*
&      (datan(dsqrt(ylamda/omgm/s))-datan(sqrt(xlamda/omgm/
&      s)))
  else
    endif
    ratio=xlamda/ylamda
    arg1=dsqrt(xlamda/omgm/s)
    arg2=dsqrt(ylamda/omgm/s)
    arg3=1.0-ratio
    aaa=omgm*xlamda/arg3/arg3
    bbb=dlog(ylamda*(omgm*s+xlamda)/xlamda/(omgm*s+
&      ylamda))
    ccc=datan(arg2)-datan(arg1)
    if( nflag .eq. 2) then
      gg=1.0-omgm+aaa*(bbb/omgm/s-2.*ccc/dsqrt(omgm*s*ylamda))
    else
      endif
    if( nflag .eq. 3) then
      gg=1.0-omgm+aaa*(2.*ccc/dsqrt(omgm*s*xlamda)-bbb/omgm/s)
    else
      endif
    endif
  endif
  c
  xx=dsqrt(s*gg)
  k0=dbsk0(xx)
  kl=dbskl(xx)
  plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
  plap2=1./s/s/plap
  return
  end

```

```

c
function plap3(s)
implicit real*8(a-h,o-z)
double precision k0,kl,gg,plap,xx
common m, /a/sn, cd, xlamda, ylamda, omgm
common /b/nflag

```

```

c
c
if(xlamda.eq. ylamda) then
gg=1.0-omgm+omgm*xlamda/(omgm*s+xlamda)
else
  if( nflag .eq. 1) then
    gg=1.0-omgm+omgm*dsqrt(xlamda/omgm/s)/(1.0-
&      dsqrt(xlamda/ylamda))
&      (datan(dsqrt(ylamda/omgm/s))-datan(sqrt(xlamda/omgm/
&      s)))
  else
    endif
    ratio=xlamda/ylamda
    arg1=dsqrt(xlamda/omgm/s)
    arg2=dsqrt(ylamda/omgm/s)
    arg3=1.0-ratio
    aaa=omgm*xlamda/arg3/arg3
    bbb=dlog(ylamda*(omgm*s+xlamda)/xlamda/(omgm*s+
&      ylamda))
    ccc=datan(arg2)-datan(arg1)
    if( nflag .eq. 2) then
      gg=1.0-omgm+aaa*(bbb/omgm/s-2.*ccc/dsqrt(omgm*s*ylamda))
    else
      endif
    endif
  endif

```



```
if( nflag .eq. 3) then
  gg=1.0-omgm+aaa*(2.*ccc/dsqrt(omgm*s*xlamda)-bbb/omgm/s)
else
endif
endif
```

c

```
xx=dsqrt(s*gg)
k0=dbsk0(xx)
k1=dbsk1(xx)
plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
plap3=1./s/s/s/plap
return
end
```



```

&      dsqrt(xlamda/ylamda))*ans1
      else
      endif
      if(nflag.eq. 2) then
        gg=1.0-omgm+omgm/((1.0-dsqrt(xlamda/ylamda))**2.)
&        *(2.*xlamda/3./omgm/s*ans2
&        -2.*xlamda/dsqrt(3.*omgm*s*ylamda))*ans1
      else
      endif
      if(nflag.eq. 3) then
        gg=1.0-omgm+omgm/((1.0-dsqrt(xlamda/ylamda))**2.)
&        *(2.*dsqrt(xlamda/3./omgm/s)*ans1
&        -2.*xlamda/3./omgm/s*ans2)
      else
      endif
    endif
  endif

```

```

c
xx=dsqrt(s*gg)
k0=dbsk0(xx)
k1=dbsk1(xx)
plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
plap2=1./s/s/plap
return
end

```

```

function plap3(s,ans1,ans2)
implicit real*8 (a-h,o-z)
double precision k0,k1,gg,plap,xx
common m,/a/sn,cd,xlamda,ylamda,omgm
common /b/nflag

```

```

c
c
if (xlamda.eq. ylamda) then
  gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)*
& dtanh(dsqrt(3.*omgm*s/xlamda))
else
  if(nflag.eq. 1) then
    gg=1.0-omgm+omgm*dsqrt(xlamda/3./omgm/s)/(1.0-
& dsqrt(xlamda/ylamda))*ans1
  else
  endif
  if(nflag.eq. 2) then
    gg=1.0-omgm+omgm/((1.0-dsqrt(xlamda/ylamda))**2.)
& *(2.*xlamda/3./omgm/s*ans2
& -2.*xlamda/dsqrt(3.*omgm*s*ylamda))*ans1
  else
  endif
  if(nflag.eq. 3) then
    gg=1.0-omgm+omgm/((1.0-dsqrt(xlamda/ylamda))**2.)
& *(2.*dsqrt(xlamda/3./omgm/s)*ans1
& -2.*xlamda/3./omgm/s*ans2)
  else
  endif
endif
endif

```

```

c
xx=dsqrt(s*gg)
k0=dbsk0(xx)
k1=dbsk1(xx)
plap=(k0+sn*xx*k1)/(s*(cd*s*(k0+sn*xx*k1)+xx*k1))
plap3=1./s/s/s/plap
return
end

```

c Definition of function **f** which **is** used for integration

```
function f1(x)
implicit real*8 (a-h,o-z)
```

c

```
f1=dtanh(x)/x
return
end
```

```
function f2(x)
implicit real*8 (a-h,o-z)
```

c

```
f2=dtanh(x)
return
end
```

```

C          THE STEHFEST ALGORITHM
C          *****
C
C          SUBROUTINE PWD(TD,N,PD,rate,cum)
C          THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
C          INVERSE OF F(S) .
C          IMPLICIT REAL*8 (A-H,O-Z)
C          DIMENSION G(50),V(50),H(25)
C          COMMON M,/a/sn,cd,xlamda,ylamda,omgm
C
C          NOW IF THE ARRAY V(I) WAS COMPUTED BEFORE THE PROGRAM
C          GOES DIRECTLY TO THE END OF THE SUBRUTINE TO CALCULATE
C          F(S) .
C          IF (N.EQ.M) GO TO 17
C          M=N
C          DLOGTW=0.6931471805599
C          NH=N/2
C
C          THE FACTORIALS OF 1 TO N ARE 'CALCULATEDINTO ARRAY G.
C          G(1)=1
C          DO 1 I=2,N
C             G(I)=G(I-1)*I
C          CONTINUE
C
C          TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H.
C          H(1)=2./G(NH-1)
C          DO 6 I=2,NH
C             FI=I
C             IF (I-NH) 4,5,6
C             H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
C             GO TO 6
C             H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
C          CONTINUE
C
C          THE TERMS (-1)**NH+1 ARE CALCULATED.
C          FIRST THE TERM FOR I=1
C          SN=2*(NH-NH/2*2)-1
C
C          THE REST OF THE SN'S ARECALCULAXED IN THE MAIN ROUTINE.
C
C          THE ARRAY V(I) IS CALCULATED.
C          DO 7 I=1,N
C
C             FIRST SET V(I)=0
C             V(I)=0.
C
C             THE LIMITS FOR K ARE ESTABLISHED.
C             THE LOWER LIMIT IS K1=INTEG((I+1/2))
C             K1=(I+1)/2
C
C             THE UPPER LIMIT IS K2=MIN(I,N/2)
C             K2=I
C             IF (K2-NH) 8,8,9
C             K2=NH
C
C             THE SUMMATION TERM IN V(I) IS CALCULATED.
C             DO 10 K=K1,K2
C                IF (2*K-I) 12,13,12
C                IF (I-K) 11,14,11
C                V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
C                GO TO 10
C                V(I)=V(I)+H(K)/G(I-K)
C                GO TO 10
C                V(I)=V(I)+H(K)/G(2*K-I)
C             CONTINUE
C
C          9
C          10

```

```
C
C      THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C      ACCORDING TO SN.
C      V(I)=SN*V(I)
C
C      THE TERM SN CHANGES ITS SIGN EACH ITERATION.
C      SN=-SN
7 CONTINUE
C
C      THE NUMERICAL APPROXIMATION IS CALCULATED.
17 A=DLOGTW/TD
   PD=0
   rate=0.
   cum=0.
   DO 15 I=1,N
     ARG=A*I
     PD=PD+V(I)*plap1(ARG)
     rate=rate+v(i)*plap2(arg)
     cum=cum+v(i)*plap3(arg)
15 CONTINUE
   PD=PD*A
   rate=rate*a
   cum=cum*a
18 RETURN
   END
```