Stanford Geothermal Program
Interdisciplinary Research in
Engineering and Earth Sciences
STANFORD UNIVERSITY
Stanford, California

SGP-TR-86

COMPUTER GENERATION OF TYPE CURVES

By

Mary Elizabeth Eipper

February 1985

Financial support was provided through the Stanford Geothermal Program under Department of Energy Contract No. DE-AT03-80SF11459 and by the Department of Petroleum Engineering, Stanford University

ABSTRACT

Computer programs are written *to* generate several of the more common published type curves. The governing equations of the curves and method of programming are presented. Problems encountered in programming and solutions to overcome these problems are discussed.

Three previously unpublished type curves are developed. These include drawdown and buildup type curves for locating sealed and constant pressure linear boundaries and a generalized radial flow type curve. The governing equations for these curves are derived.

A section of the computer generated complete working type curves is Included in the Appendix.

TABLE **OF** CONTENTS

Page

ABSTRACT ii

LIST **OF** FIGURES iv

1. INTRODUCTION 1

2. LINE SOURCE SOLUTION TYPE CURVE 4

3. LINEAR FAULT TYPE CURVE FOR A DRAWDOWN INTERFERENCE TEST 7

4. LINEAR FAULT TYPE CURVES FOR A DRAWDOWN AND BUILDUP INTERFERENCE TEST 10

6. STORAGE AND SKIN TYPE CURVE 14

6. SLUG TEST TYPE CURVES 16

7. BOUNDED RADIAL FLOW TYPE CURVE 21

8. GENERALIZED RADIAL FLOW TYPE CURVE 24

9. VERTICAL FRACTURE TYPE CURVES 27

10. CONCLUSIONS AND RECOMMENDATIONS 32

NOMENCLATURE 33

REFERENCES 35

APPENDIX A. COMPUTER PROGRAMS 36

APPENDIX **B.** SELECTED DATA 68

APPENDIX C. WORKING TYPE CURVES 91

LIST OF FIGURES

Pane

Fig. 2.1 - P_D versus $\frac{t_D}{r_D^2}$ for the line source solution. (after Ramey, 1970) 6

Flg. 3.1 - Well geometry for linear fault interference test 7

Fig. 3.2 - P_D versus $\frac{t_D}{r_D^2}$ for **a** drawdown test in a linear fault system. (after Stallman, 1952) 9

Fig. **4.1** - P_D versus $\frac{t_D}{r_D^2}$ for **a** drawdown-buildup test in a closed linear fault system 12

Fig. 4.2 - P_D versus $\frac{t_D}{r_D^2}$ **for a** drawdown-buildup test in a constant pressure linear boundary system. 13

Fig. 6.1 - P_{wD} versus t_D for a well with storage and skin effect. (after Ramey, 1970) 16

Fig. 6.1 - P_{DR} versus $\frac{t_D}{r_D^2}$ for slug test early **or** late time flow period data. (after Ramey, et. al., 1975) 18

Fig. 6.2 - P_{DR} versus $\frac{t_D}{r_D^2}$ for slug test late time flow period data. (after Ramey, et. al., 1975) 19

Fig. **6.3** - $(1 - P_{DR})$ versus $\frac{t_D}{C_D}$ for slug test early time flow period flow data. (after Ramey, et. al., 1975) **20**

Fig. 7.1 - P_{wD} versus t_D for **a** well with constant rate production in **a** finite circular reservoir. (after Fiock and Aziz, 1963) **23**

Fig. 8.1 - $P_{wD}^{\bullet}$ versus $t_D^{\bullet}$ for **a** well with constant rate production in a generalized finite circular reservoir. (Curves collapsed to r_{e_D} = 100.) 26

Fig. 9.1 - Definitions of drainage area, X_e, and X_f **for** a vertical fracture system 27

Fig. **9.2** - P_{wD} versus t_{Df} for a uniform flux vertical fracture. (after Gringarten, et. al., 1974) 30

Flg. 9.3 - P_{wD} versus t_{Df} for **an** infinite conductivity vertical fracture. (after Gringarten, et. al., 1874) 31

1: INTRODUCTION

Type curve matching techniques have proven to be a useful method of analyzing transient well test ddta. There are many different type curves available. The method can be used for drawdown, buildup, interference, constant pressure, or any transient well test with a known P_D and t_D. The purpose of this research is to generate P_D versus t_D values by means of a computer in order to produce several working type curves. The ultimate objective is the publication of a complete type curve manual. Unless otherwise stated t_D and P_D are defined as in Eqs. 1.1 and 1.2.

$$t_D = \frac{0.000264kt}{\varphi\mu C_t r_w^2} \qquad (1.1)$$

$$P_D = \frac{kh}{141.2qB\mu}\left(P_i - P_{r,t}\right) \qquad (1.2)$$

Type curve matching is most appropriate for well tests too short for the semilog straight line to develop. For single well tests, this method should only be used when other more conventional techniques such as semilog analysis methods cannot be used. Simplicity of model identification and speed are some of the advantages of the method.

Many of the type curves are derived from the diffusivity equation with boundary conditions depending upon the situation. The equation is solved using Laplace transforms. The equation in Laplace space is then analytically inverted resulting in an Integral equation. Some analytical inversions are complicated at best and impractical at worst. In the past, many of the type curves' data points were generated by numerically solving the integrals. The data points were then plotted by a draftsman. This method is time consuming and prone to human error.

Now, with the Stehfest algorithm **(1970)** and more advanced computer capabilities, the Laplace equation$ governing the many type curves may be inverted numerically. The graphics routines available are capable **of** plotting the points and duplicating the log-log grid background **of** the type curves.

In this report, twelve type curves are presented. Nine of the type curves are from the literature. These include:

1) Line Source Solution Type Curve

2) Linear Fault Type Curve for a Drawdown Interference Test

3) Storage and Skin Type Curve

4) **Slug** Test Type Curve **for** Early and Late **Flow** Data

6) Slug Test Type Curve for Late Time Flow Data

6) Slug Test Type Curve for Early Time Flow Data

7) Bounded Radial Flow Type Curve

8) Infinite Conductivity Vertical Fracture Type Curve

9) Uniform Flux Vertical Fracture Type Curve

The governing equations are presented for each of these cases. These equations are used in the computer programs to generate the type curves. Small versions of the finished type curves are found in the body of the report. The more complete, working curves are found in Appendix C.

Three of the type curves generated are previously unpublished. The first two are modifications **of** the linear fault type curve **of** Stallman (1952) for a drawdown interference test. They are type curves for drawdown and buildup interference **test for 1)** locating a sealed' linear fault and 2) locating a constant pressure linear boundary. The third type curve is one which mathematically collapses to one curve the Flock and Aziz (1963) curves for the unsteady state radial flow equation with a constant terminal rate.

2: LINE SOURCE SOLUTION TYPE CURVE

The line source or Exponential Integral solution type curve represents the pressure response of a well, producing at a constant rate, in an infinite reservoir of constant thickness. The total compressibility of the reservoir and viscosity of the single phase reservoir fluid are assumed constant during the drawdown period.

The line source solution is derived from the radial diffusivity equation assuming that the radius of the wellbore is insignificant compared to the size of the reservoir. The solution gives the pressure response at any point in the reservoir where the influence of the finite wellbore radius is insignificant. Mueller and Witherspoon have shown that the influence of the producing wellbore radius is insignificant when $r_D > 20$ and $\frac{t_D}{r_D^2} > 0.5$, or when $\frac{t_D}{r_D^2} > 25$ for any value of r_D.

The solution does not account for skin or wellbore storage making it more appropriate for an interference test than a single well test.

Theis (1935) has given the governing equation for the line source solution:

$$P_D = -\frac{1}{2} Ei\left(\frac{-r_D^2}{4t_D}\right) \tag{2.1}$$

The Exponential Integral is defined as:

$$Ei(-x) = \int_x^\infty \frac{e^{-u}}{u} du \tag{2.2}$$

Values of $\frac{t_D}{r_D^2}$ in the desired range are generated for which the corresponding P_D values are calculated. Fig. 2.1 presents the line source solution.

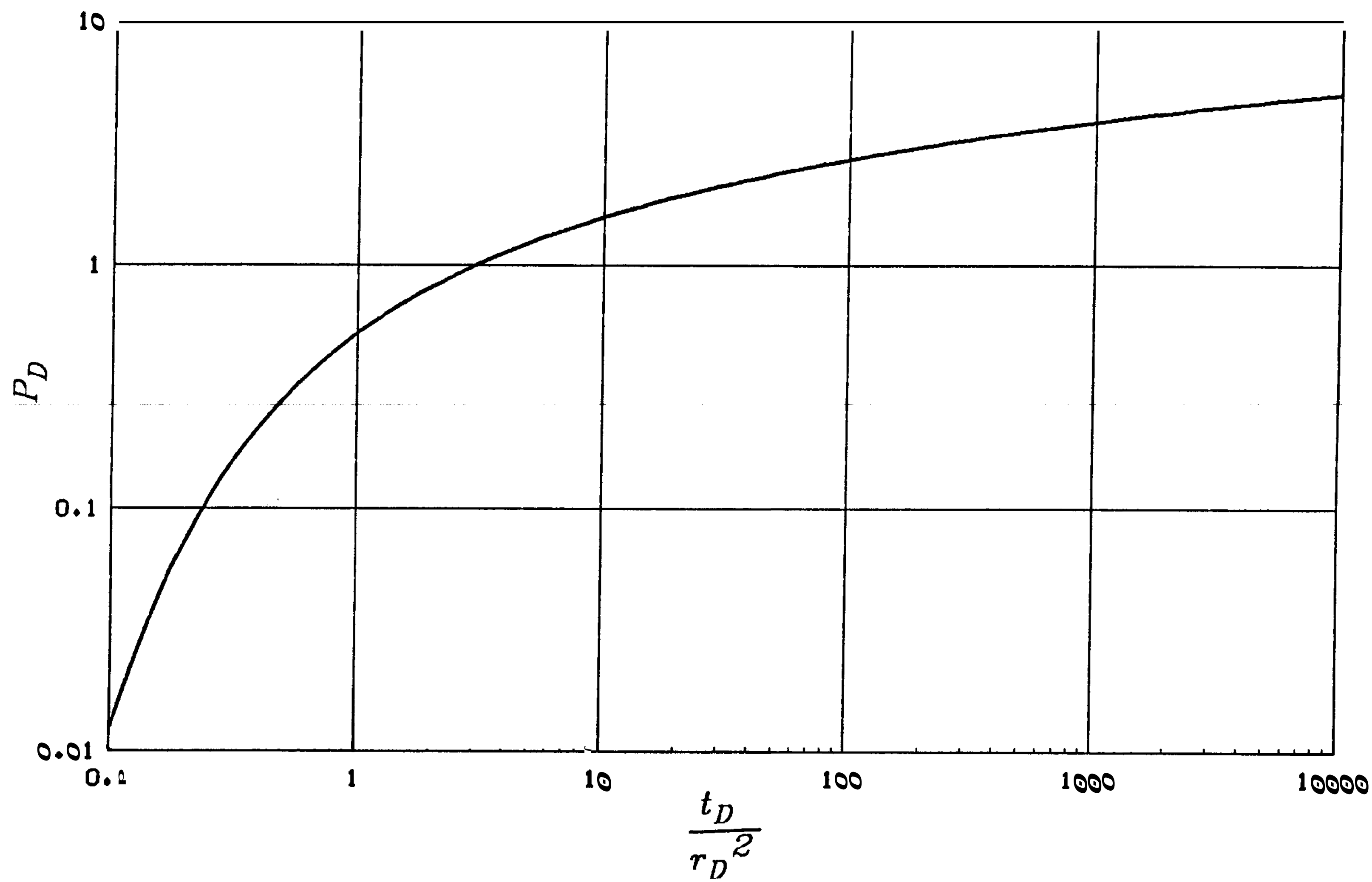

Fig. 2.1 - P_D versus $\frac{t_D}{r_D^2}$ for the line source solution. (after Ramey, 1970)

3: LINEAR FAULT TYPE CURVE FOR A DRAWDOWN INTERFERENCE TEST

The drawdown test linear fault type curve utilizes superposition in space of an image well to produce the effect of a linear fault. Fig. **3.1** illustrates the well geometry.

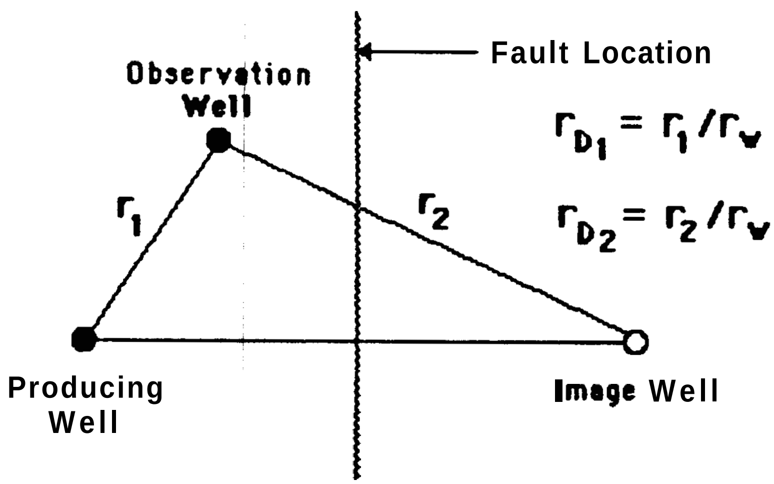

Fig. **3.1** - Well geometry for linear fault interference test.

The Image well either injects or produces at the same rate as the producing well. Production at the image well creates the effect of a sealed linear fault halfway between the image and producing wells. Injection at the image well creates the effect of a constant pressure'linear boundary between the image and producing **wells.**

The solution is obtained by superposition of the line source solution. Although t_D **is** the same for both the production and image wells, r_D is different for the two wells because the distances between the production and image wells and the observation well are not the same. Equation 3.1 describes the solution for the sealed linear boundary case.

$$P_D = P_D\left[\frac{t_D}{r_{D_1}^{\ 2}}\right] + P_D\left[\frac{t_D}{r_{D_2}^{\ 2}}\right] \tag{3.1}$$

Because the radial diffusivity equation is linear, the net effect at the observation well is the sum of the individual effects of the producing and image wells. Substitution of the exponential integral solution, Eq. 2.1, into Eq. 3.1 yields the sealed fault solution, Eq. 3.2.

$$P_D = -\frac{1}{2}\left[Ei\left[\frac{-r_{D_1}^{\ 2}}{4t_D}\right] + Ei\left[\frac{-(r_2/r_1)^2\, r_{D_1}^{\ 2}}{4t_D}\right]\right] \tag{3.2}$$

For the case of the constant pressure linear boundary, Eq. 3.3 represents the superposition.

$$P_D = P_D\left[\frac{t_D}{r_{D_1}^{\ 2}}\right] - P_D\left[\frac{t_D}{r_{D_2}^{\ 2}}\right] \tag{3.3}$$

Again substitution of Eq. 2.1 yields Eq. 3.4.

$$P_D = -\frac{1}{2}\left[Ei\left[-\frac{r_{D_1}^{\ 2}}{4t_D}\right] - Ei\left[\frac{-(r_2/r_1)^2\, r_{D_1}}{4t_D}\right]\right] \tag{3.4}$$

Eqs. 3.2 and 3.4 are solved for various values of $\frac{r_2}{r_1}$ and are plotted in Fig. 3.2. When $\frac{r_2}{r_1}$ is infinite the solution becomes the line source, This figure was first developed by Stallman (1952).

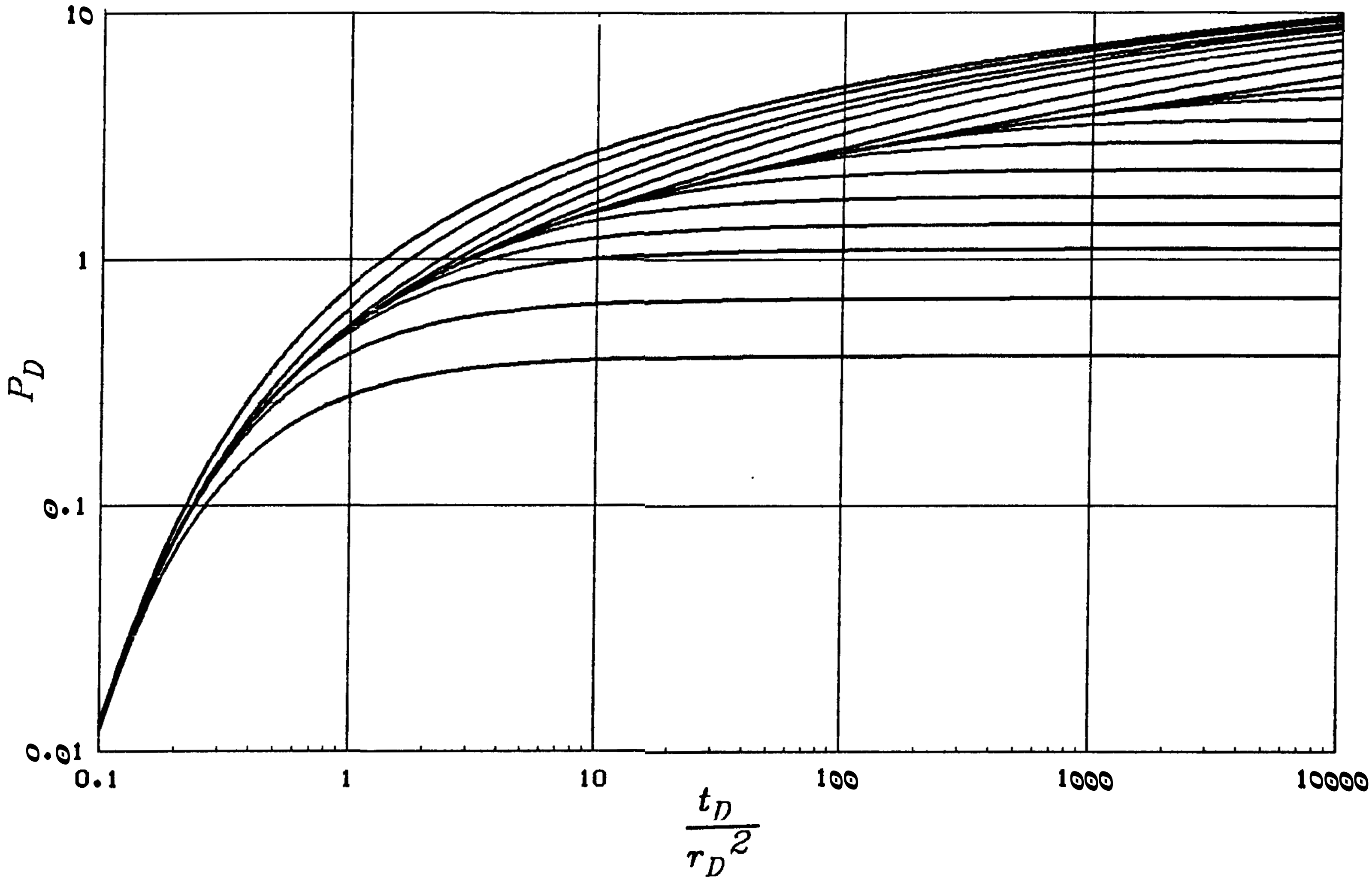

Fig 3.2 - P_D versus $\frac{t_D}{r_D^2}$ for a drawdown test in a linear fault system. (after Stallman, 1952)

4: LINEAR FAULT TYPE CURVES FOR A DRAWDOWN AND BUILDUP INTERFERENCE TEST

These type curves are **a** modification of the original type curve of Stallman (1952) discussed in Chapter **3.** Superposition in time is utilized to create the effect of shutting in the producing well after a dimensionless production time, t_{pD}. This type of test **has** the advantage of producing more detectable pressure change onces the producing well is shut in.

Equation **4.1** describes the drawdown-buildup solution **for** the sealed linear fault case.

$$P_D = \left[P_D\left(\frac{t_D}{{r_{D_1}}^2}\right) + P_D\left(\frac{t_D}{{r_{D_2}}^2}\right)\right] + \left[P_D\left(\frac{t_D - t_{pD}}{{r_{D_1}}^2}\right) + P_D\left(\frac{t_D - t_{pD}}{{r_{D_2}}^2}\right)\right] \qquad (4.1)$$

Substituting the exponential integral into the equation yields:

$$P_D = -\frac{1}{2}\left[Ei\left(\frac{-{r_{D_1}}^2}{4t_D}\right) + Ei\left(\frac{-(r_2/r_1)^2\,{r_{D_1}}^2}{4t_D}\right)\right] +$$
$$\frac{1}{2}\left[Ei\left(\frac{-{r_{D_1}}^2}{4t_D} - \frac{{r_{D_1}}^2}{4t_{P_D}}\right) - Ei\left(\frac{-(r_2/r_1)^2\,{r_{D_1}}^2}{4t_D} - \frac{(r_2/r_1)^2\,{r_{D_1}}^2}{4t_{P_D}}\right)\right] \qquad (4.2)$$

This equation is solved for various values of $\frac{r_2}{r_1}$

Figure **4.1** presents the log-log type curves for the drawdown-buildup tests **for** a sealed linear fault. Equadion **4.3** describes the drawdown-buildup solution **for** the constant pressure fault case:

$$P_D = \left[P_D\left(\frac{t_D}{r_{D_1}^{\;2}}\right) + P_D\left(\frac{t_D}{r_{D_2}^{\;2}}\right)\right] - \left[P_D\left(\frac{t_D - t_{pD}}{r_{D_1}^{\;2}}\right) + P_D\left(\frac{t_D - t_{pD}}{r_{D_2}^{\;2}}\right)\right] \qquad (4.3)$$

Again substituting the exponehtial integral solution yields:

$$P_D = -\frac{1}{2}\left[Ei\left(\frac{-r_{D_1}^{\;2}}{4t_D}\right) - Ei\left(\frac{-(r_2/r_1)^2\, r_{D_1}^{\;2}}{4t_D}\right)\right] +$$

$$\frac{1}{2}\left[Ei\left(\frac{-r_{D_1}^{\;2}}{4t_D} + \frac{r_{D_1}^{\;2}}{4t_{P_D}}\right) + Ei\left(\frac{-(r_2/r_1)^2\, r_{D_1}^{\;2}}{4t_D} + \frac{(r_2/r_1)^2\, r_{D_1}^{\;2}}{4t_{P_D}}\right)\right] \qquad (4.4)$$

Figure 4.2 presents the log-log type curve for the drawdown-buildup tests for constant pressure linear boundaries.

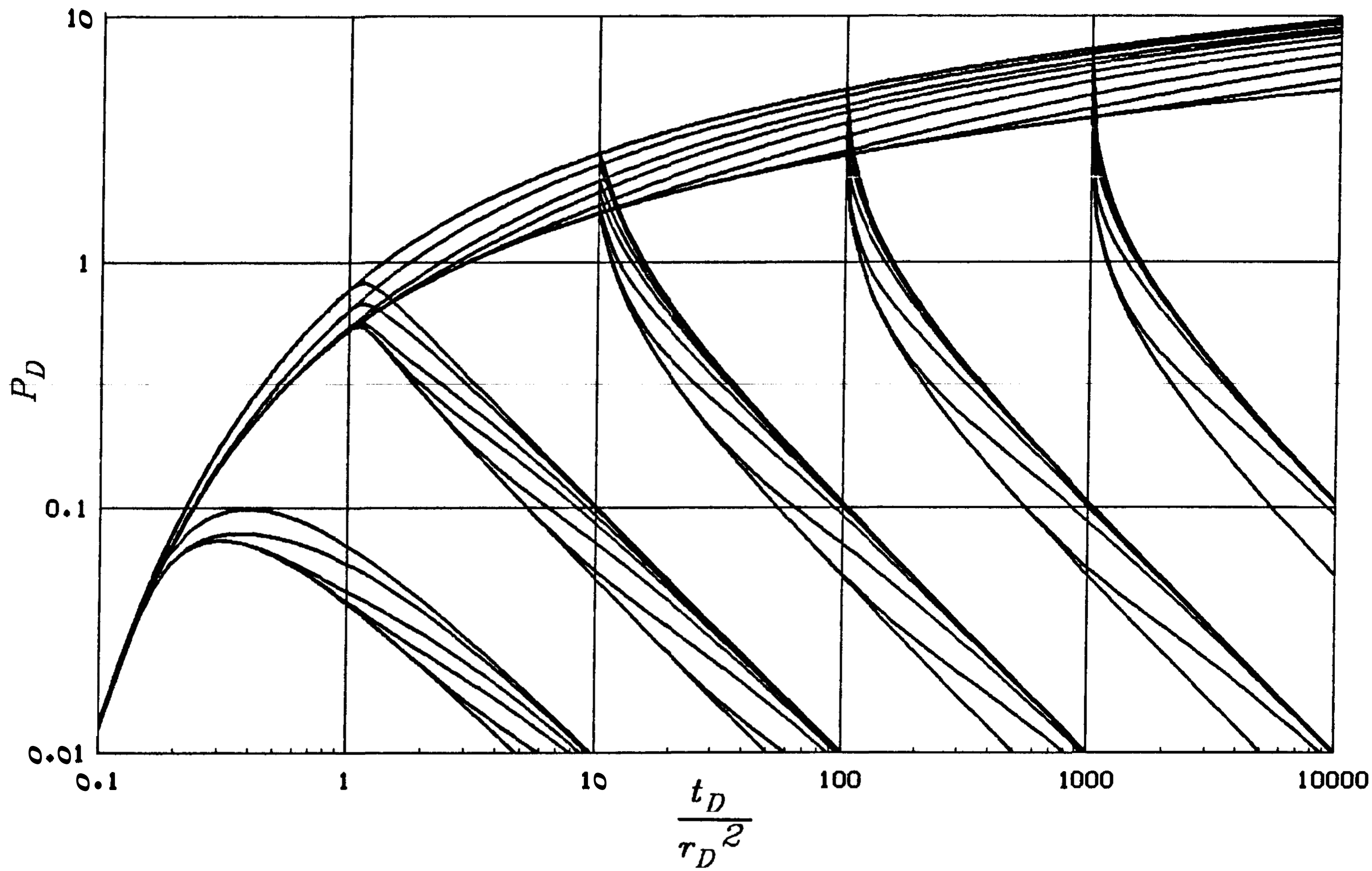

Fig. 4.1 - P_D versus $\frac{t_D}{r_D^2}$ for a drawdown-buildup test in a closed linear fault system.

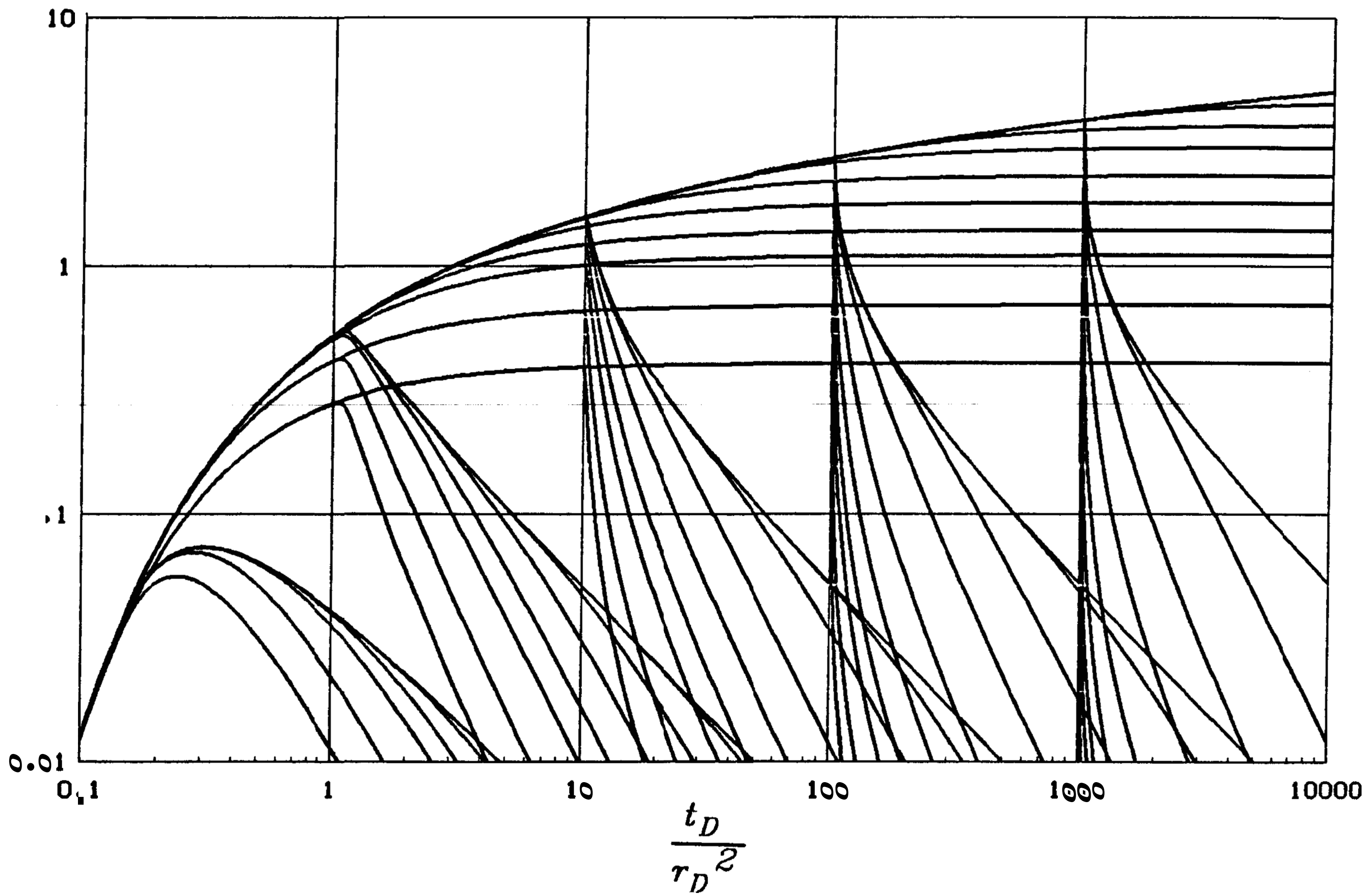

Fig. 4.2 - P_D versus $\frac{t_D}{r_D^2}$ for a drawdown-buildup test in a constant pressure linear boundary system.

5: STORIAGE AND SKIN TYPE CURVE

The storage and skin type curve of Agarwal, Al-Hussainy and Ramey **(1970)** is useful for analysis of short time single well tests during which wellbore storage and skin are present.

Solving the diffusivity equation with the appropriate boundary conditions found in the paper by Agarwal et. al. **(1970)** leads to the Laplace transform of the dimensionless pressure equation governing the curves:

$$\bar{P}_{wD} = \frac{K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s})}{s\left[\sqrt{s}K_1(\sqrt{s}) + C_D s\left[K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s})\right]\right]} \qquad (5.1)$$

The Laplace transform is inverted numerically using the Stehfest algorithm **(1970).** Values of t_D along the desire4 range are first generated **for** which corresponding values of P_{wD} are calculated. Although P_{wD} for positive values of skin effect are inverted directly, this is not **possible for** negative skin values. P_{wD} values **for** negative skin effect **are** inverted indirectly by representing negative skin by a larger effective wellbore radius. To accomplish this, the Laplace equation **is** inverted using a zero skin effect and the dimensionless time and wellbore storage constant are redefined **as** follows:

$$t''_D = t_D e^{2s} \qquad (5.2)$$

$$C'_D = C_D e^{2s} \qquad (5.3)$$

Figure **5.1** shows the plotted values.

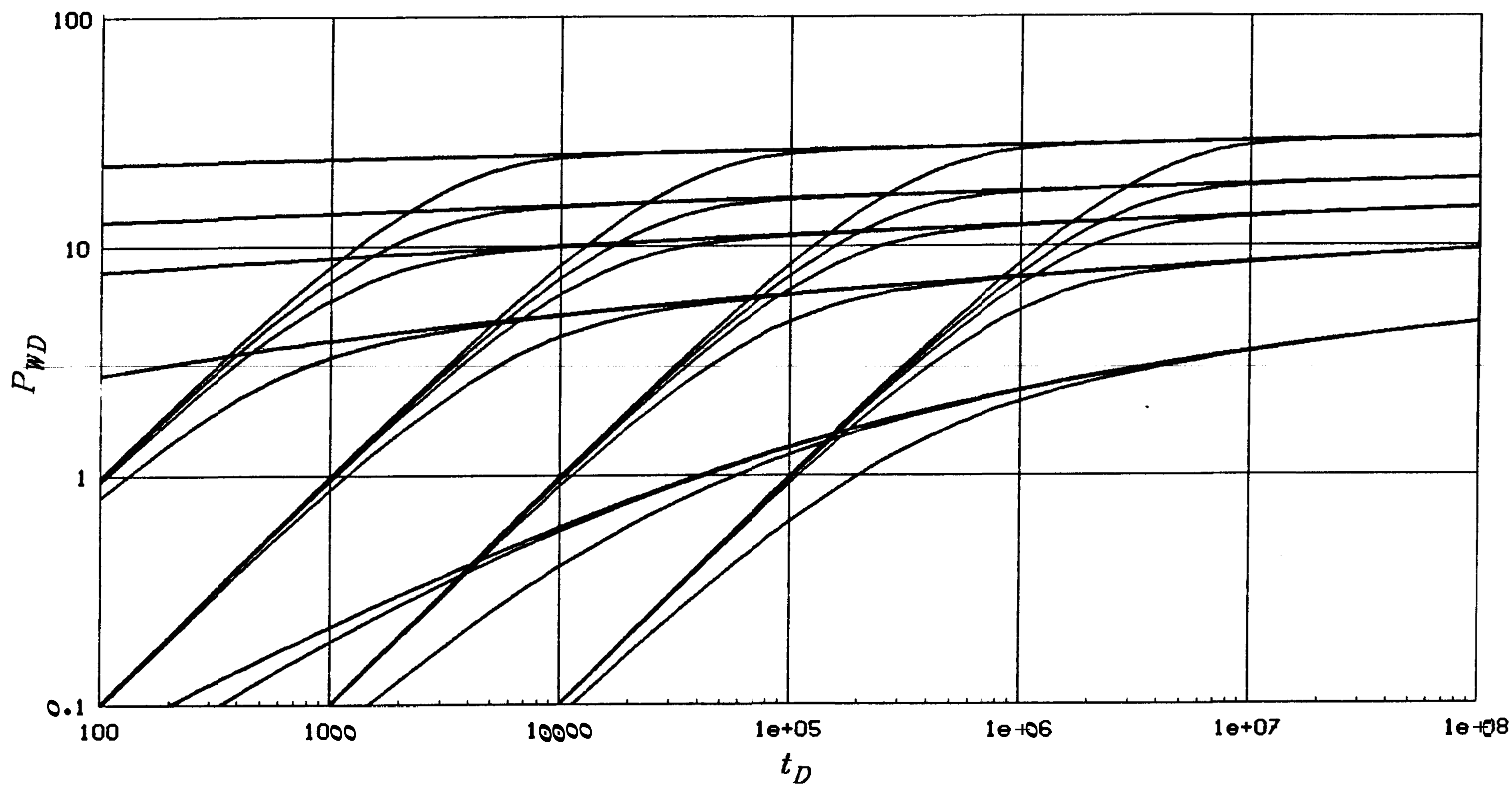

Fig. 5.1 - P_{WD} versus t_D for a well with storage and skin effect. (after Ramey, 1970)

6: SLUG TEST TYPE CURVES

The slug test or drill stem test type curves of Ramey, Agarwal, and Martin (1975), are used to analyze tests during which flow never reaches the surface. These curves include the skin effect and the wellbore storage coefficient. The governing equation for the type curves is the same as Eq. 5.1 for the storage and skin type curve with the exception of a different definition of P_D.

$$\bar{P}_{DR} = \frac{K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s})}{s\left[\sqrt{s}K_1(\sqrt{s}) + C_D s\left(K_0(\sqrt{s}) + S\sqrt{s}K_1(\sqrt{s})\right)\right]} \tag{6.1}$$

P_{DR} Is defined in Eq. 6.2.

$$P_{DR} = \frac{P_i - P_{wf}(t)}{P_i - P_o} \tag{6.2}$$

The equation is solved usihg the Stehfest (1970) algorithm for various values of C_De^{2S} and plotted in three formats, each yielding a greater sensitivity to a particular portion of the data.

1) Semi-log plot of P_{DR} versus $\frac{t_D}{C_D}$ for early and late time flow data.

2) Log-log plot of P_{DR} versus $\frac{t_D}{C_D}$ for late time flow data.

3) Log-log plot of $(1 - P_{DR})$ versus $\frac{t_D}{C_D}$ for early time flow data.

For values of C_De^{2S} less than 10^{20}, S is set equal to zero. This simplified the calculations because then C_De^{2S} equals C_D. For values of C_D greater than or equal to 10^{20}, large values of C_D create numbers that cannot be represented by 64 bit arithmetic. This problem is solved by choosing arbirary positive values of S and back calculating C_D from the desired value of $C_D e^{2S}$.

Figures 6.1, 6.2, and 6.3 show the three slug test type curves.

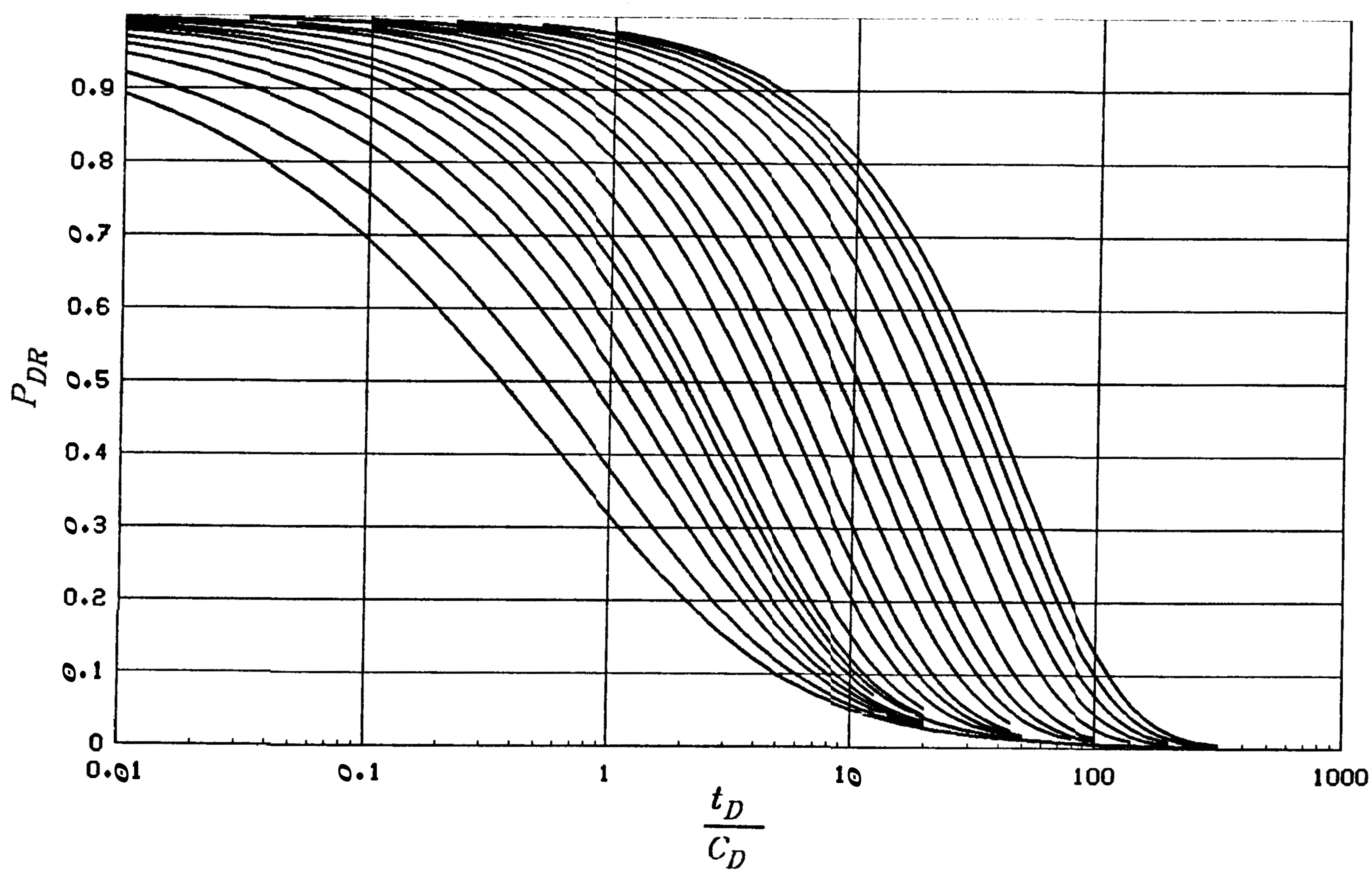

Fig. 6.1 - P_{DR} versus $\frac{t_D}{C_D}$ for slug test early or late time flow period data. (after Ramey, et. al., 1975)

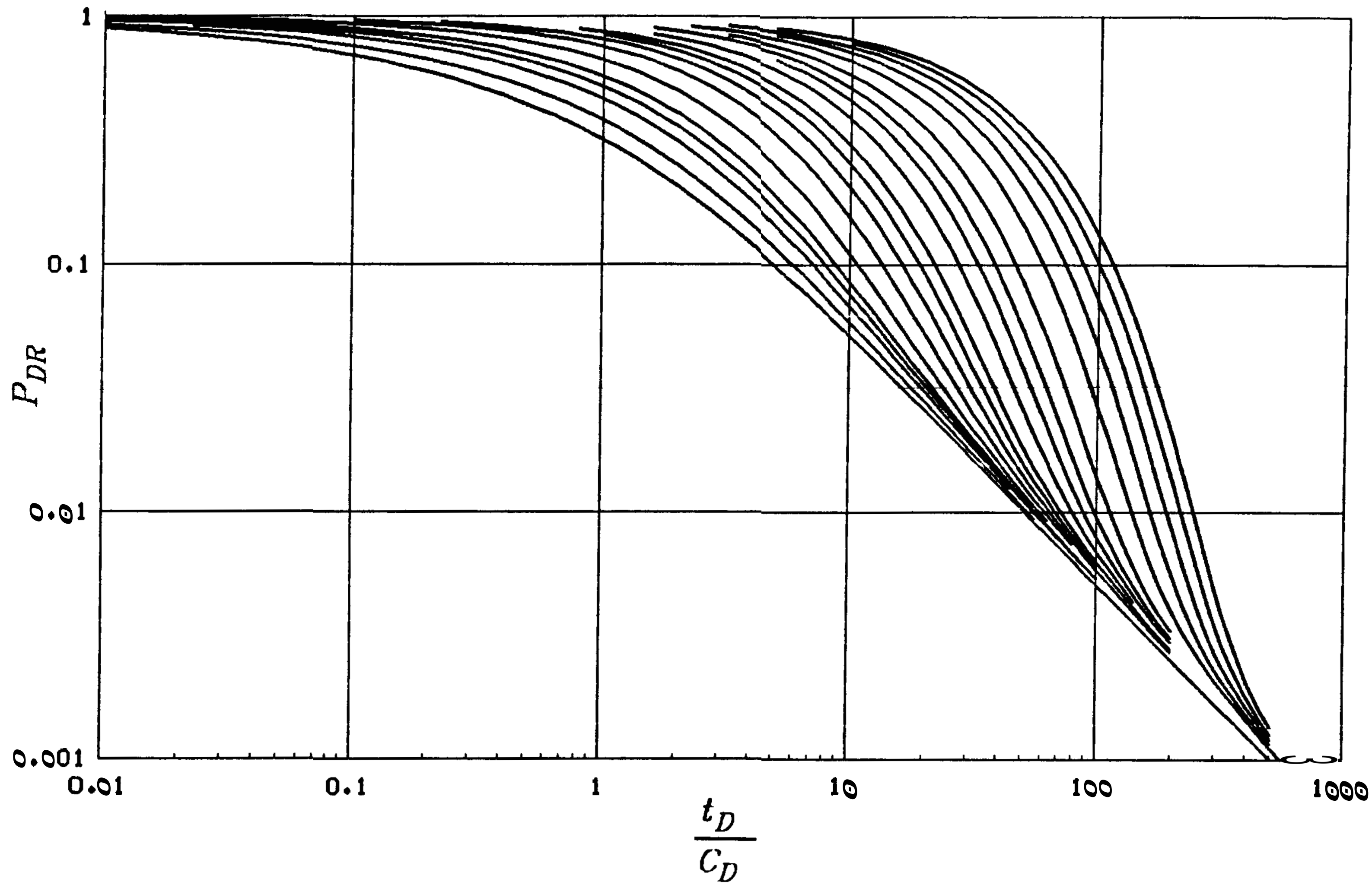

Fig. 6.2 - P_{DR} versus $\frac{t_D}{C_D}$ for slug test late time flow period data. (after Ramey, et. al., 1975)

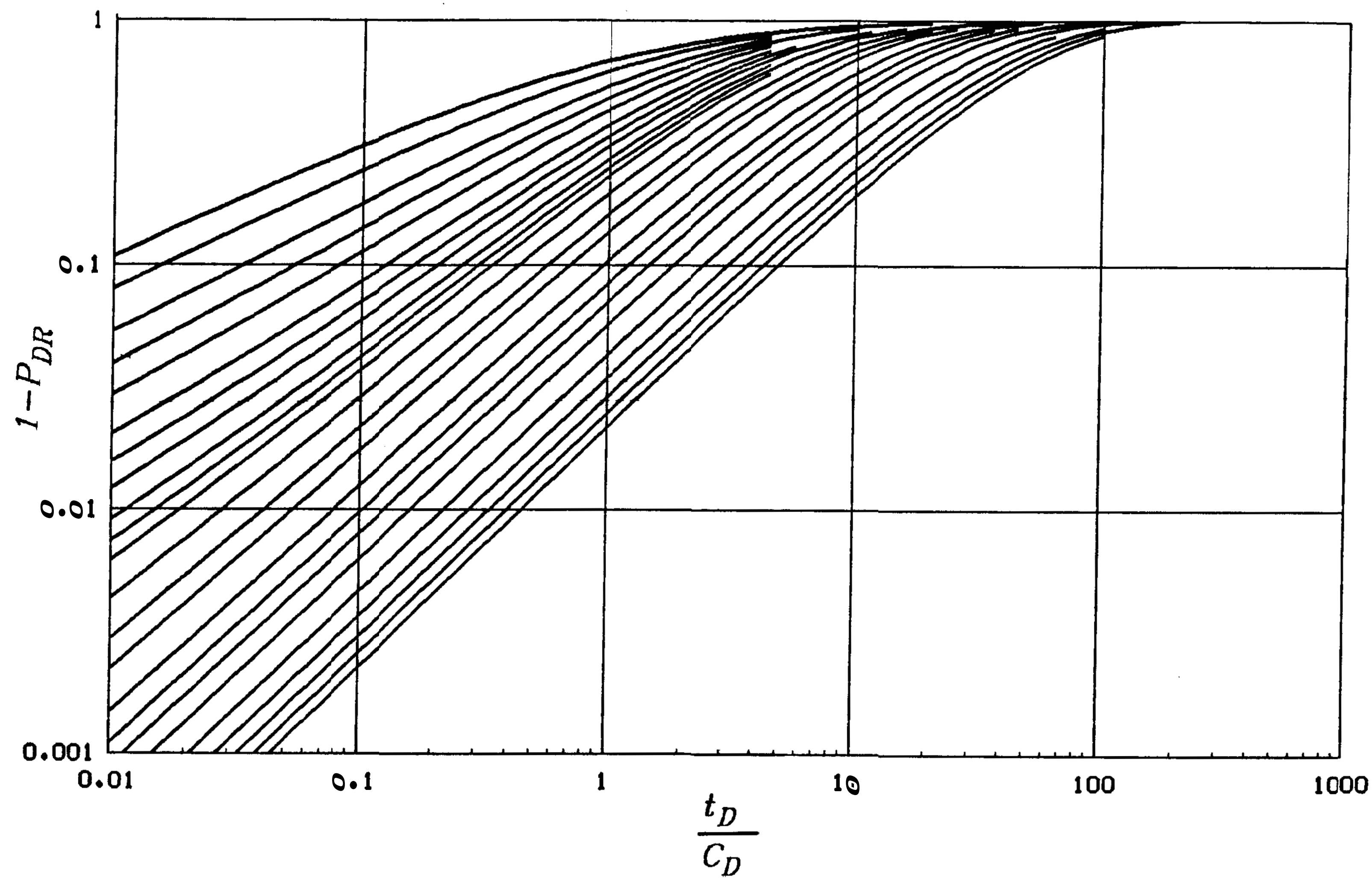

Fig. 6.3 - $(1 - P_{DR})$ versus $\frac{t_D}{C_D}$ for slug test early time flow period flow data. (after Ramey, et. al., 1975)

7: BOUNDED RADIAL FLOW TYPE CURVE

Flock and Aziz (1963) presented a type curve for radial flow at a constant terminal rate. This type curve is used to determine the type of outer boundary on hand: infinite, constant pressure, or no flow. The actual size of the reservoir may be estimated.

Van Everdingen and Hurst (1949) describe the governing equations for the finite radius constant terminal rate case. The Laplace transform of the governing equation for the pressure drop at the well in an infinite reservoir is Eq. 7.1.

$$\overline{P}_D = \frac{K_0(\sqrt{s})}{s^{\frac{3}{2}} K_1(\sqrt{s})} \quad (7.1)$$

For the case of a finite reservoir with no flow across the exterior boundary, the Laplace transform of the governing equation is Eq. 7.2.

$$\overline{P}_D = \frac{K_1(\sqrt{s})I_0(\sqrt{s}) + I_1(\sqrt{s})K_0(\sqrt{s})}{s^{\frac{3}{2}}\left[I_1(r_{e_D}\sqrt{s})K_1(\sqrt{s}) - K_1(r_{e_D}\sqrt{s})I_1(\sqrt{s})\right]} \quad (7.2)$$

Similarly the equation for a constant pressure exterior boundary is Eq. 7.3.

$$\overline{P}_D = \frac{-K_0(r_{e_D}\sqrt{s}) + K_0(\sqrt{s})I_0(r_{e_D}\sqrt{s})}{s^{\frac{3}{2}}\left[K_0(r_{e_D}\sqrt{s})I_1(\sqrt{s}) + I_0(r_{e_D}\sqrt{s})K_1(\sqrt{s})\right]} \quad (7.3)$$

The Stehfest algorithm (1970) is used to invert the Laplace equations for the Infinite case and for various values of r_{e_D}.

Figure 7.1 shows the plotted result.

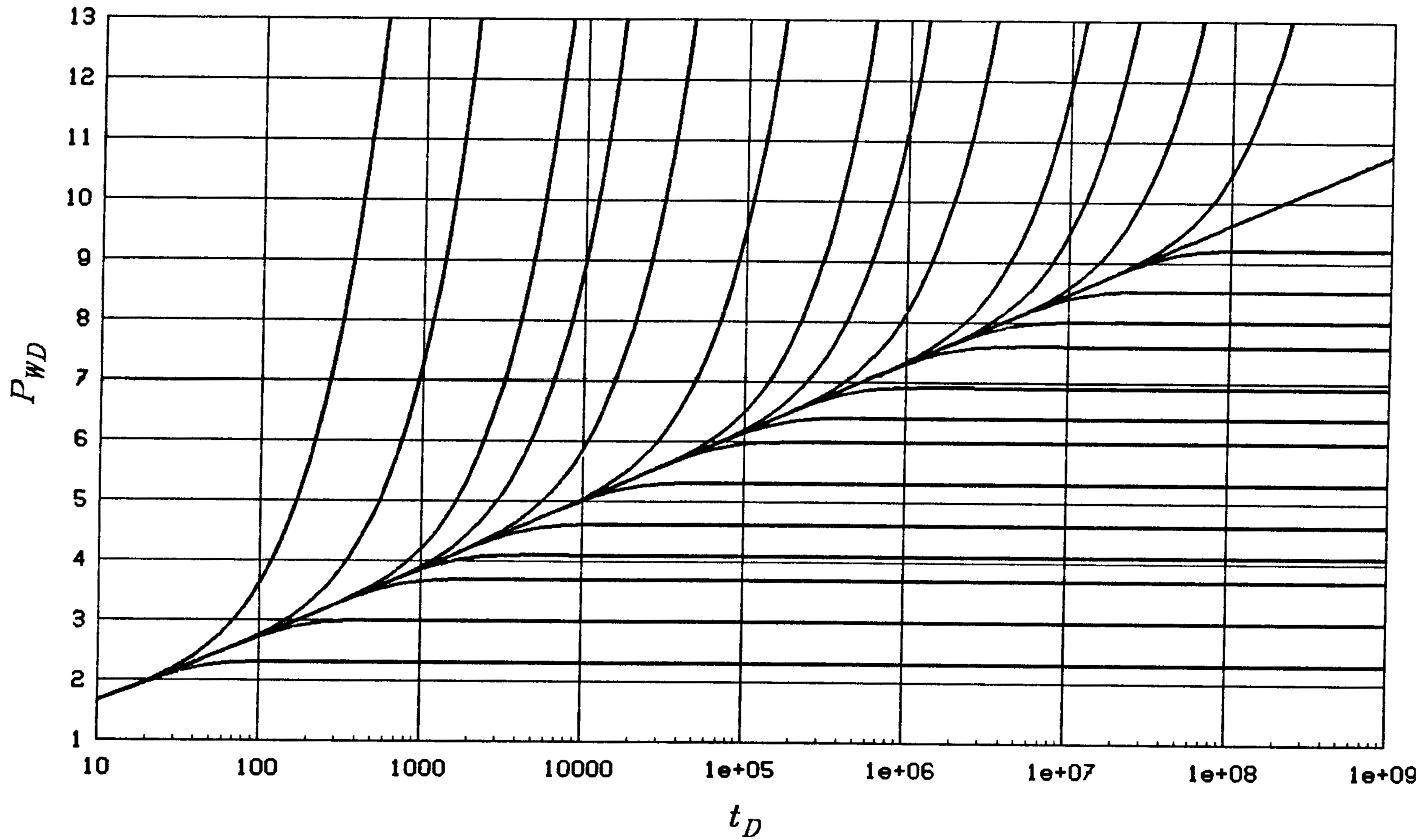

Fig. 7.1 - P_{WD} versus t_D for a well with constant rate production in a finite circular reservoir. (after Flock and Aziz, 1963)

8: GENERALIZED RADIAL FLOW TYPE CURVE

By redefining P_D and t_D it is possible to mathematically collapse to one curve, the various curves presented by Flock and Aziz (1963), see Fig. 7.1. Any value of r_{e_D} may be matched on this generalized curve making the curve more compact versatile. It is arbitrarily chosen to collapse the curves of Fig. 7.1 to the curve corresponding to r_{e_d} = 100. The collapsing is done in two steps. First the curves are moved vertically so that all the curves match at the steady state portion of the Flock and Aziz curve. This may be easily accomplished by realizing that at steady state:

$$P_D = \ln(r_{e_D}) \tag{8.1}$$

The new value of P_D is defined by subtracting the effect of ln (r_{e_D}) and adding the effect of ln (100).

$$P_D^* = P_D - \ln\left(\frac{r_{e_D}}{100}\right) \tag{8.2}$$

To shift the curves horizontally, the infinite acting log approximation of the line source is considered:

$$P_D = \frac{1}{2}\left[\ln\left(\frac{t_D}{r_D{}^2}\right) + 0.80907\right] \tag{8.3}$$

For the curves to collapse horizontally to r_{e_D} = 100:

$$P_D = \frac{1}{2}\left[\ln\left(\frac{t_D^*}{100^2}\right) + 0.80907\right] \tag{8.4}$$

Therefore:

$$t_D^* = t_D\left(\frac{100^2}{r_D{}^2}\right) \tag{8.5}$$

This method of mathematically collapsing the curves was first suggested by William E. Brigham (1970) of Stanford University.

Figure 8.1 shows the Flock and Aziz (1963) curve collapsed to $r_{e_D} = 100$. At early times, where the log approximation of the line source is not valid, the curve flattens.

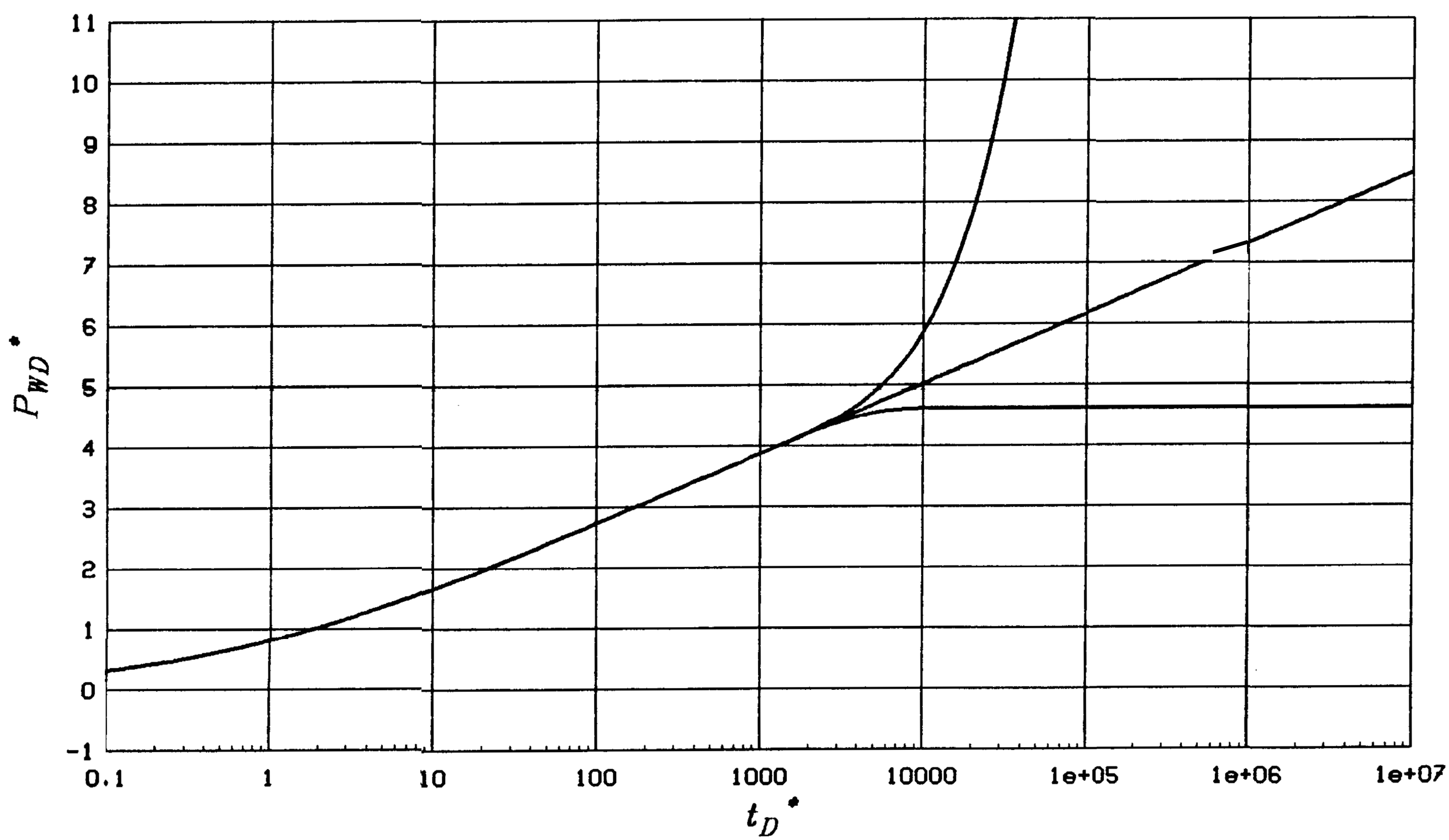

Fig. 8.1 - P_{WD}^* versus t_D^* for a well with constant rate production in a generalized finite circular reservoir. (Curves collapsed to r_{e_D} = 100.)

9: VERTICAL FRACTURE TYPE CURVES

The vertical fracture type curves of Gringarten, Ramey, and Raghavan (1974) are useful for short time type curve analysis. Information may be obtained concerning permeabilities, fracture length, and drainage area.

Two types of vertical fractures are considered: 1) Fluid enters along a uniform flux vertical fracture at a constant rate. 2) Fluid enters the infinite conductivity fracture in such a way that a constant pressure is maintained along the fracture. Figure 9.1 illustrates the system.

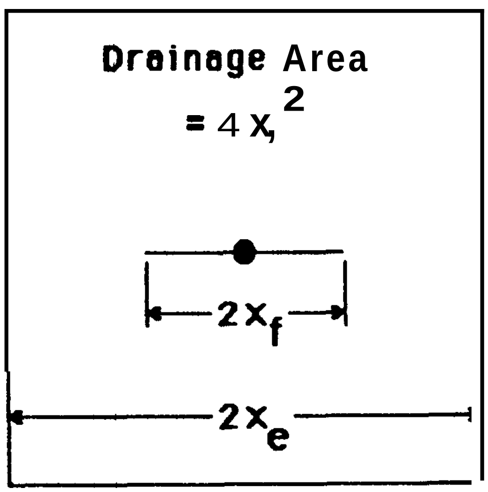

Fig. 9.1 - Definitions of drainage area, X, and X_f for a verticle fracture system.

The case of a vertical fracture in an infinite system is now considered. Equation 9.1 governs the pressure drop for the uniform flux fracture in an infinite system.

$$P_{wD} = \sqrt{\pi t_{Df}}\, erf\left(\frac{1}{2\sqrt{t_{Df}}}\right) - \frac{1}{2} Ei\left(-\frac{1}{4t_{Df}}\right) \tag{9.1}$$

Equation **9.2** is for the infinite conductivity fracture in an infinite system.

$$P_{wD} = \frac{1}{2}\sqrt{\pi t_{Df}}\left[erf\left(\frac{0.134}{\sqrt{t_{Df}}}\right) + erf\left(\frac{0.866}{\sqrt{t_{Df}}}\right)\right] - $$
$$- 0.067\, Ei\left(\frac{-0.018}{t_{Df}}\right) - 0.433\, Ei\left(-\frac{0.750}{t_{Df}}\right) \tag{9.2}$$

Dimensionless time, t_{Df}, **is** based on the fracture length and is defined in Eq. **9.3.**

$$t_{Df} = \frac{0.000264kt}{\varphi \mu C_t X_f^{\,2}} \tag{9.3}$$

P_{wD} in these equations **is** calculated directly **for** various values of t_{Df}. The equation governing finite reservoirs is based on t_{DA}, the dimensionless time based on a square drainage area defined as in Eq. **9.4.**

$$t_{DA} = \frac{0.000264kt}{\varphi \mu C_t 4X_e^{\,2}} \tag{9.4}$$

Substituting Eq. **9.4** into **9.3** yields Eq. **9.5,** the equation relating the finite and infinite drainage area equations.

$$t_{Df} = \frac{4X_e^{\,2}}{X_f^{\,2}} t_{DA} \tag{9.5}$$

The pressure drop for a vertical fracture in the center of a square **is** given by **Eq. 9.6.** For a uniform flux vertical fracture X_D = 0.0 and for an infinite conductivity vertical fracture X_D = **0.732.**

$$P_{wD} = 2n \int_0^{2t_{DA}} \left[1 + 2 \sum_{n=1}^{\infty} \exp(-4n^2\pi^2 t'_{DA}) \right] * \left[1 + 2 \sum_{n=1}^{\infty} \exp(-4n^2\pi^2 t'_{DA}) \frac{\sin n\pi \frac{X_f}{X_e}}{n\pi \frac{X_f}{X_e}} \cos n\pi X_D \frac{X_f}{X_e} \right] dt'_A \qquad (9.6)$$

Eq. 9.6 is solved by numerical integration. Since at short time the finite drainage area cases are infinite acting, values of P_{wD} from the infinite area cases are chosen as starting points of the numerical integration for the finite cases. This reduces the CPU time. **Also,** more accurate results may be obtained then by starting the integration at t_{DA} and P_{wD} values equal to zero.

Perhaps, a better way to approach this problem would be to transform **Eq. 9.6** into Laplace space and then use the Stehfest **(1970)** numerical inversion **algo**rithm **to** solve for P_{wD}. This approach will be left for future study.

Figs. 9.3 and **9.4** show the type curves for the uniform flux and infinite conductivity vertical fracture cases, respectively.

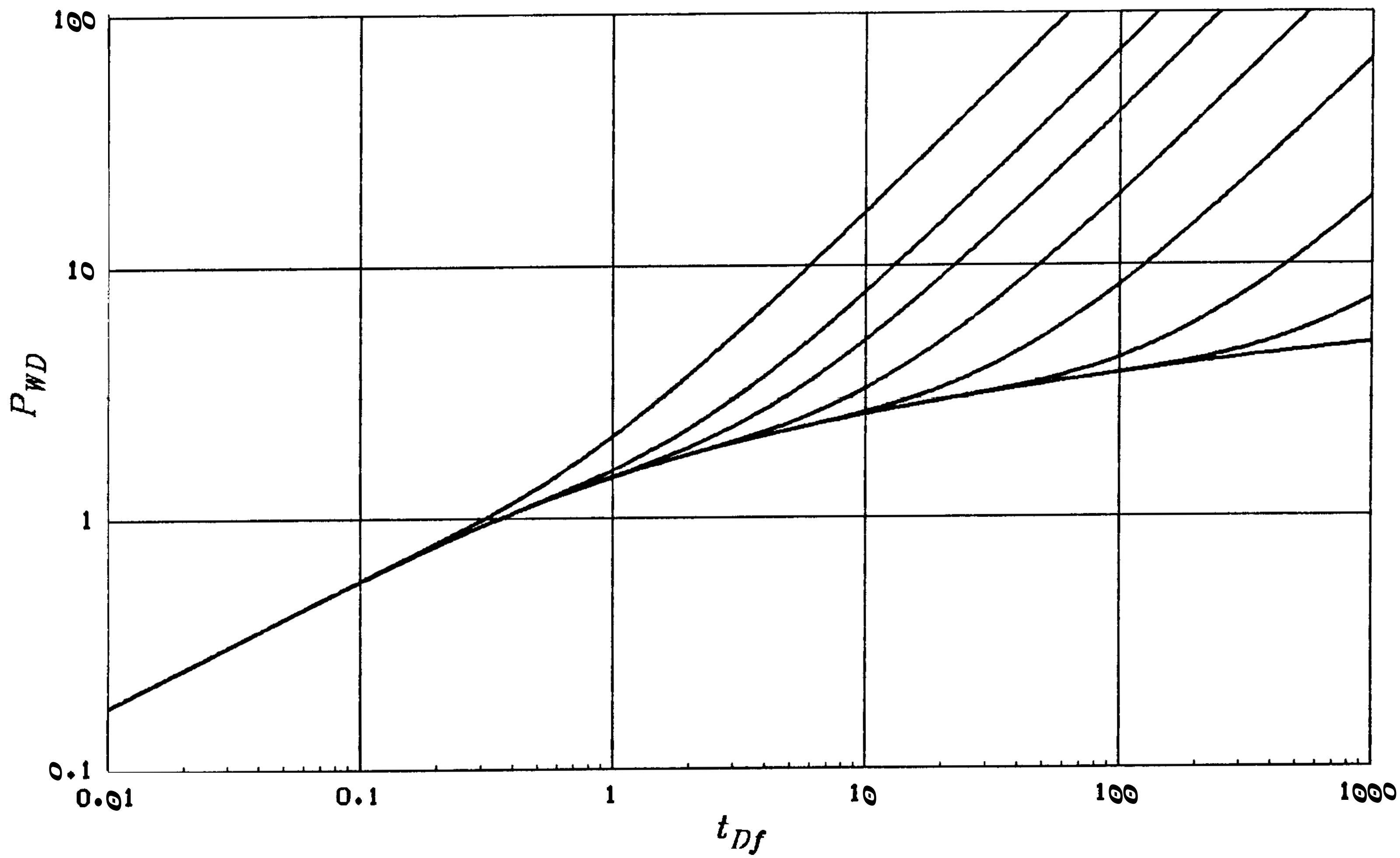

Fig. 9.2 - P_{WD} versus t_{Df} for a uniform flux vertical fracture. (from Gringarten, et. al.)

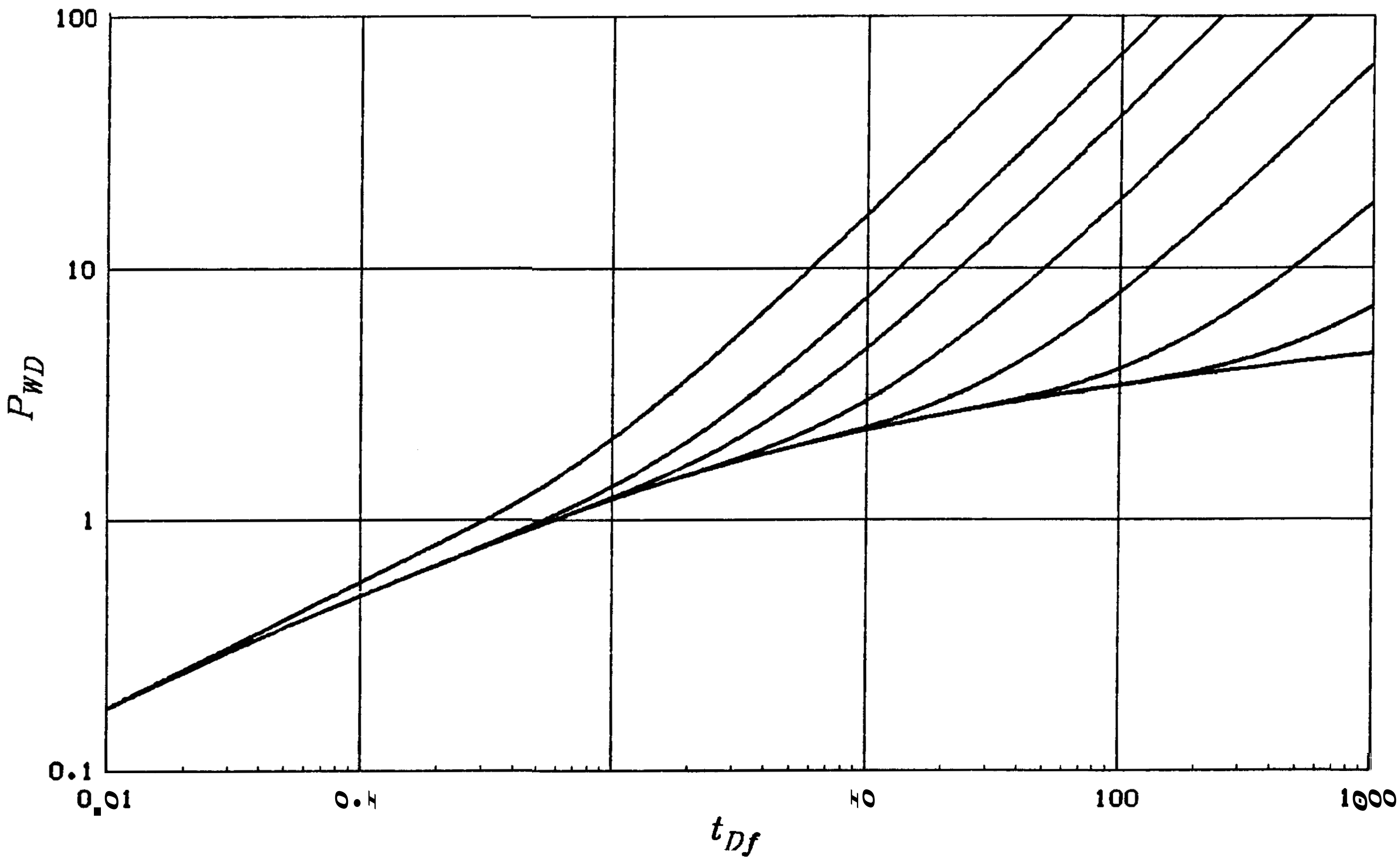

Fig. 9.3 - P_{WD} versus t_{Df} for an infinite conductivity vertical fracture. (from Gringarten, et. al.)

10: CONCLUSIONS AND RECOMMENDATIONS

Twelve type curves were generated by means of a computer. The line source solution type curve was generated using the Exponential Integral. The governing equations of the linear fault curves were variations on the line source solution equation. The drawdown test linear fault type curve used superposition in space of the Exponential Integral to obtain the linear fault. The drawdown and buildup test linear fault type curve went one step further by using superposition in time to achieve the buildup portion or the curve.

The storage and skin type curve, slug test type curves, and radial flow type curve were generated by numerical inversion of their governing Laplace equations with the Stehfest algorithm (1970). This method proved efficient in both time spent writing the programs and the required CPU time.

The infinite conductivity and uniform flux vertical fracture type curves were generated by numerical integration of the published integral equation. Problems were encountered with this method in obtaining small enough time steps at the beginning of the integration. A alternative way to generate the curves would be to transform the integral equation into Laplace space and invert the equation numerically.

For future type curve gsneration, a general interactive program should be written so that the desired range and number of t_D values may be chosen. Then, the only modification necessary for generating new type curves would involve the subroutines for computing the Laplace or the real time equations.

NOMENCLATURE

C_D = dimensionless wellbore storage

C_t = total formation compressibility (psi^{-1})

I_0 = modified Bessel function of the first kind of order zero

I_1 = modified Bessel function of the first kind of order one

h = formation sand thickness (ft)

k = reservoir permeability (md)

K_0 = modified Bessel function of the second kind of order zero

K_1 = modified Bessel function of the second kind of order one

P_D = dimensionless pressure

$\bar{P}_D$ = Laplace transform of dimensionless pressure

P^*_D = redefined dimensionless pressure to collapse radial flow type curve

P_{DR} = dimensionless pressure ratio used in type curve matching slug test data

$\bar{P}_{DR}$ = Laplace transform of dimensionless pressure ratio used in type curve matching slug test data

P_{wD} = dimensionless pressure at well

$\bar{P}_{wD}$ = Laplace transform of dimensionless pressure at well

P_{wf} = flowing pressure in the wellbore (psia)

P_i = static initial reservoir pressure (psia)

P_o = minimum wellbore pressure achieved during slug test (psia)

r_D = dimensionless radial distance

r_{eD} = dimensionless radial distance defined at the exterior boundary

r_w = wellbore radius (ft)

s = Laplace variable

S = dimensionless wellbore skin factor

t = time (seconds)

t_D = dimensionless time

t^*_D = redefined dimensionless time to collapse radial flow type curve

t_{DA} = dimensionless time **based** on a square drainage area

t_{Df} = dimensionless time based on vertical fracture half length

t_{pD} = dimensionless producing time

μ = fluid viscosity (centipoise)

φ = formation porosity (fraction)

X_D = dimensionless coordinate based on the fracture half length

X_e = square reservoir **half** dimension

X_f = fracture half length

REFERENCES

Agarwal, Ram G., Al-Hussainy, Rafi and Ramey, H. J., Jr.: "An Investigation of Wellbore Storage and Skin Effect in Unsteady Liquid Flow: I. Analytical Treatment," *Soc. Pel. Eng. J.* (September, 1970) 279-290; *Trans.*, AIME, 249.

Brigham, W. E.: *Advanced Reservoir Engineering Class*, Stanford University (1979).

Earlougher, Robert C., Jr.: *Advances in Well Test Analysis*, Monograph Series, Society of Petroleum Engineers, Dallas (1967) Vol. 5.

Flock, D. L. and Aziz, K.: "Unsteady State Gas Flow - Use of Drawdown Data in the Prediction of Gas Well Behavior," *J. Can. Pet. Tech.* (Spring, 1963) 9-15.

Gringarten, Alain C., Ramey, Henry J., Jr., and Raghavan, R.: "Unsteady-State Distributions Created by a Well With a Single Infinite-Conductivity Vertical Fracture," *Soc. Pet. Eng. J.* (August, 1974) 347-360; *Trans.* AIME, 257.

Ramey, Henry J., Jr., Agarwal, Ram G., and Martin, Ian: "Analysis of 'Slug Test' of DST Flow Period Data," *J. Can. Pet. Tech.* (July-September, 1975) 37-42.

Van Everdingen, A. F. and Hurst, W.: "The Application of the Laplace Transformation to Flow Problems in Reservoirs," *Trans.*, AIME (December, 1949) 186, 305-324.

Mueller, Thomas D., and Witherspoon, Paul A.: "Pressure Interference Effects Within Reservoirs and Aquifers," *J. Pet. Tech.* (April, 1965) 471-474.

Stehfest, H.: "Algorithm 368, Numerical Inversion of Laplace Transforms," *Communications of the ACM*, D-6 (January 1970) 13, No. 1, 47-48.

Theis, Charles V.: "The Relationship Between the Lowering of the Piezometric Surface and the Rate and Duration of Discharge Using Ground-Water Storage," *Trans.*, AGU (1936) 519-523.

Witherspoon, P. A., and Javandel, I., Neuman, S. P., and Freeze, R. A.: *Interpretation of Aquifer Gas Storage Conditions From Water Pumping Tests*, Monograph of Project NS-38, American Gas Association, Inc., New York (1967) 93-128

APPENDIX A

COMPUTER PROGRAMS

LINE SOURCE SOLUTION TYPE CURVE

```
c
c This program calculates PD vs tD/rD**2 for the line source solution curve.
c
c Variables used:
c mmdei=imsl routine for exponential integral solution
c pd=dimensionless pressure
c td=dimensionless time divided by dimensionless radius squared
c
c
c These loops generate tD/rD**2 values between 0.1 and 10000 and calculate
c corresponding PD values.
c
      Implicit real*8(a-h,o-z)
      dimension td(1000),pd(1000)
      double precision mmdei
      iopt=1
      n=0
      do 10 i=1,6
         do 20 j=1,20
            tdlog=-2.+i+(j-1)/20.
            tdd=10.**tdlog
            if(tdd.gt.10000.) go to 10
            n=n+1
            td(n)=tdd
            arg=-1./(4.*td(n))
            pd(n)=-0.5*mmdei(iopt,arg,ier)
   20    continue
   10 continue
c
c This loop outputs the values for plotting.
c
      write(6,1000)n
 1000 format(i3)
      do 30 i=1,n
         write(6,2000)td(i),pd(i)
 2000    format(e10.5,5x,e15.7)
   30 continue
      stop
      end
c
c
```

SEALED LINEAR FAULT PORTION

OF DRAWDOWN TEST LINEAR FAULT TYPE CURVE

```
c
c This program calculates PD vs tD for the sealed ltnear fault
c portion of the drawdown test linear fault type curve.
c
c Varlables used are:
c mmdei=imsl routlne to calculate the exponential integral
c nrd=number of values of rd desired
c pd=dimensionless pressure
c rd-distance from observation well to image well divided by distance from
c observation well to producing well
c td=dimensionless time
c
      Implicit real*8(a-h,o-z)
c
c This loop reads in the values of rd and calls the subroutine for each value.
c
      read(5,*)nrd
      do 10 i=1,nrd
         read(5,*)rd
         call dd(rd)
   10 continue
      stop
      end
c
c Subroutine dd (drawdown) generates values of tD between 0.1 and 10000
c and calculates the corresponding values of PD.
c
      Subroutine dd(rd)
      implicit real*8(a-h,o-z)
      dimension td(1000),pd(1000)
      double precision mmdei
      iopt=1
      n=0
      do 10 i=1,6
         do 20 j=1,20
            tdlog=-2+i+(j-1)/20.
            tdd=10.**tdlog
            if(tdd.gt.10000)go to 10
            n=n+1
            td(n)=tdd
            arg1=-1./(4*td(n))
            arg2=-(rd*rd)/(4*td(n))
            if(arg2.lt.-50.)go to 40
            pd(n)=-0.5*(mmdei(iopt,arg1,ier)+mmdei(iopt,arg2,ier))
            go to 20
   40       pd(n)=-0.5*(mmdei(iopt,arg1,ier))
   20    continue
   10 continue
c
c This loop outputs the values for plotting.
c
      write(6,1000)n
      do 30 i=1,n
         write(6,2000)td(i),pd(i)
   30 continue
c
 1000 format(i3)
 2000 format(e10.5,5x,e15.7)
      return
      end
c
```

CONSTANT PRESSURE LINEAR BOUNDARY PORTION OF DRAWDOWN TEST LINEAR FAULT TYPE CURVE

```
c
c This program calculates PD vs tD for the constant pressure ltnear boundary
c portion of the drawdown test linear fault type curve.
c
c Variables used are:
c mmdei=imsl routine to crlculate the exponential integral
c nrd=number of values of rd desired
c pd=dimensionless pressure
c rd=distance from observation well to image well divided by distance from
c observation well to producing well
c td=dimensionless time
c
      implicit real*8(a-h,o-z)
c
c This loop reads in the values of rd and calls the subroutine for each value.
c
      read(5,*)nrd
      do 10 i=1,nrd
         read(5,*)rd
         call dd(rd)
   10 continue
      stop
      end
c
c Subroutine dd (drawdown) generates values of tD between 0.1 and 10000
c and calculates the corresponding values of PD.
c
      Subroutine dd(rd)
      implicit real*8(a-h,o-z)
      dimension td(1ØØØ),pd(1ØØØ)
      double precision mmdei
      iopt=1
      n=Ø
      do 10 i=1,6
         do 20 j=1,2Ø
            tdlog=-2+i+(j-1)/2Ø.
            tdd=1Ø.**tdlog
            if(tdd.gt.1ØØØØ)go to 10
            n=n+1
            td(n)=tdd
            arg1=-1./(4*td(n))
            arg2=-(rd*rd)/(4*td(n))
            if(arg2.lt.-5Ø.)go to 4Ø
            pd(n)=-Ø.5*(mmdei(iopt,arg1,ier)-mmdei(iopt,arg2,ier))
            go to 20
   40       pd(n)=-Ø.5*(mmdei(iopt,arg1,ier))
   28    continue
   10 continue
c
c This loop outputs the values for plotting.
c
      write(6,1ØØØ)n
      do 30 i=1,n
         write(6,2ØØØ)td(i),pd(i)
   30 continue
c
 1000 format(i3)
 2008 format(e1Ø.5,5x,e15.7)
      return
      end
c
```

BUILDUP PORTION OF THE DRAWDOWN AND BUILDUP TEST

SEALEDLINEARFAULTTYPECURVE

```
c
c This program calculates PD vs tD during the buildup
c portion of the drawdown bulldup test SEALED ltnear fault type curve.
c Dimensionless times produced considered are 8.1, 1, 10, 1ØØ, and 1000.
c PD vs tD is calculated for the desired values of rd in the input.
c Although rd is infinite for the line source solution, rd=Ø.Ø
c is the flag for the line source.
c
c Variables used are:
c mmdei=imsl routine to calculate the exponential integral
c nrd=number of values of rd desired
c pd=dimensionless pressure
c rd=distance from observation well to image we11 divided by distance from
c observation well to producing well
c td=dimensionless time
c tpd=dimensionless producing time
c
c
      implicit real*8(a-h,o-z)
c
c This loop reads in the values of rd and calls the subroutine.
c
      read(5,*)nrd
      do 10 i=1,nrd
         read(5,*)rd
         call bu(rd)
   10 continue
c
      stop
      end
c
c
c Subroutine bu (buildup) calculates PD vs tD for tpd values of 0.1, 1, 10, 1ØØ,
c and 1000 (i=1,2,3,4,5 respectively) for the passed value of rd.
c
      subroutlne bu(rd)
      implicit real*8(a-h,o-z)
      dimension tdd(1ØØØ),pdd(1ØØØ)
c
      do 10 i=1,5
         tpd=1Ø.**(-2.+i)
         n=Ø
c
         do 20 j=1,6
            do 30 k=1,5Ø
               tdlog=-3.+i+j+(k-1.)/5Ø.
               td=1Ø.**tdlog
               if(td.gt.1ØØØØ)go to 20
               if(rd.eq.Ø.Ø)pd=pduls(td,tpd)
               if(rd.gt.Ø.Ø)pd=pdu(td,tpd,rd)
               if(pd.lt.Ø.ØØ9)go to 20
               n=n+1
               tdd(n)=td
               pdd(n)=pd
 30         continue
 20      continue
c
c This loop outputs the values for plotting.
c
         write(6,1ØØØ)n
         do 40 l=1,n
            write(6,2ØØØ)tdd(l),pdd(l)
 40      continue
c
```

BUILDUP PORTION OF THE DRAWDOWN AND BUILDUP TEST

SEALED LINEAR FAULT TYPE CURVE (CONTINUED)

```
 10   continue
c
1300  format(5x,i3)
2000  format(5x,e1Ø.5,5x,e15.7)
      return
      end
c
c
c This function calculates buildup pd for finite values of rd.
c
      function pdu(td,tpd,rd)
      implicit real*8(a-h,o-z)
      double precision mmdei
      iopt=1
      arg1=-1./(4.*td)
      ei1=mmdei(iopt,arg1,ier)
      arg2=-(rd*rd)/(4.*td)
      if(arg2.lt.-5Ø.)ei2=Ø.Ø
      if(arg2.ge.-5Ø.)ei2=mmdei(iopt,arg2,ier)
      if(td.eq.tpd)go to 10
      arg3=-1./(4.*(td-tpd))
      if(arg3.lt.-5Ø.)ei3=Ø.Ø
      if(arg3.ge.-5Ø.)ei3=mmdei(iopt,arg3,ier)
      arg4=-(rd*rd)/(4.*(td-tpd))
      if(arg4.lt.-5Ø.)ei4=Ø.Ø
      if(arg4.ge.-5Ø.)ei4=mmdei(iopt,arg4,ier)
      pdu=-Ø.5*(ei1+ei2)+Ø.5*(ei3+ei4)
      go to 20
10    pdu=-Ø.5*(ei1+ei2)
2Ø    return
      end
c
c
c This function calculates buildup pd for the linesource.
c
      function pduls(td,tpd)
      implicit real*8(a-h,o-z)
      double precision mmdei
      iopt=1
      arg1=-1./(4.*td)
      ei1=mmdei(iopt,arg1,ier)
      if(td.eq.tpd)go to 10
      arg2=-1./(4.*(td-tpd))
      if(arg2.lt.-5Ø.)ei2=Ø.Ø
      if(arg2.ge.-5Ø.)ei2=mmdei(iopt,arg2,ier)
      pduls=-Ø.5*(ei1)+Ø.5*(ei2)
      go to 20
 10   pduls=-Ø.5*ei1
 20   return
      end
```

BUILDUP PORTION OF THE DRAWDOWN AND BUILDUP TEST

CONSTANT PRESSURE LINEAR BOUNDARY TYPE CURVE

```
c
c This program calculates PD vs tD during the buildup portion of the
c drawdown buildup test CONSTANT PRESSURE linear toundary type curve.
c Dimensionless times produced considered are 0.1, 1, 10, 1ØØ, and 1000.
c PD vs tD is calculated for the desired values of rd in the input.
c Although rd is infinite for the line source solution, rd=Ø.Ø
c is the flag for the line source.
c
c Variables used are:
c mmdei=imsl routlne to calculate the exponential integral
c nrd=number of values of rd desired
c pd=dimensionless pressure
c rd=distance from observation well to image well divided by distance from
c observation well to producing well
c td=dimensionless time
c tpd=dimensionless producing tlme
c
c
      implicit real*8(a-h,o-z)
c
c Thls loop reads In the values of rd and calls the subroutine.
c
      read(5,*)nrd
      do 10 i=1,nrd
         read(5,*)rd
         call bu(rd)
   10 continue
c
      stop
      end
c
c
c Subroutine bu (buildup) calculates PD vs tD for tpd values of 0.1, 1, 10, 100
c and 1ØØØ (i=1,2,3,4,5 respectively) for the passed value of rd.
c
      subroutine bu(rd)
      implicit real*8(a-h,o-z)
      dimension tdd(1ØØØ),pdd(1ØØØ)
c
      do 10 i=1,5
         tpd=1Ø.**(-2.+i)
         n=Ø
c
         do 20 j=1,6
            do 30 k=1,1ØØ
               tdlog=-3.+i+j+(k-1.)/1ØØ.
               td=1Ø.**tdlog
               if(td.gt.1ØØØØ)go to 20
               if(rd.eq.Ø.Ø)pd=pduls(td,tpd)
               if(rd.gt.Ø.Ø)pd=pdu(td,tpd,rd)
               if(pd.lt.Ø.ØØ6)go to 20
               n=n+1
               tdd(n)=td
               pdd(n)=pd
 30         continue
 20      continue
c
c This loop outputs the values for plotting.
c
         write(6,1ØØØ)n
         do 40 l=1,n
            write(6,2ØØØ)tdd(l),pdd(l)
 48      continue
```

```
c
```

BUILDUP PORTION OF THE DRAWDOWN AND BUILDUP TEST

CONSTANT PRESSURE LINEAR BOUNDARY TYPE CURVE (CONTINUED)

```
 10    continue
c
1000   format(5x,i3)
2000   format(5x,e1Ø.5,5x,e15.7)
       return
       end
c
c
c This function calculates buildup pd for finite values of rd.
c
       function pdu(td,tpd,rd)
       implicit real*8(a-h,o-z)
       double precision mmdei
       iopt=1
       arg1=-1./(4.*td)
       ei1=mmdei(iopt,arg1,ier)
       arg2=-(rd*rd)/(4.*td)
       if(arg2.lt.-5Ø.)ei2=Ø.Ø
       if(arg2.ge.-5Ø.)ei2=mmdei(iopt,arg2,ier)
       if(td.eq.tpd)go to 10
       arg3=-1./(4.*(td-tpd))
       if(arg3.lt.-5Ø.)ei3=Ø.Ø
       if(arg3.ge.-5Ø.)ei3=mmdei(iopt,arg3,ier)
       arg4=-(rd*rd)/(4.*(td-tpd))
       if(arg4.lt.-5Ø.)ei4=Ø.Ø
       if(arg4.ge.-5Ø.)ei4=mmdei(iopt,arg4,ier)
       pdu=-Ø.5*(ei1-ei2)+Ø.5*(ei3-ei4)
       go to 20
 10    pdu=-Ø.5*(ei1-ei2)
 20    return
       end
c
c
c This function calculates buildup pd for the linesource.
c
       function pduls(td,tpd)
       implicit real*8(a-h,o-z)
       double precision mmdei
       iopt=1
       arg1=-1./(4.*td)
       ei1=mmdei(iopt,arg1,ier)
       if(td.eq.tpd)go to 10
       arg2=-1./(4.*(td-tpd))
       if(arg2.lt.-5Ø.)ei2=Ø.Ø
       if(arg2.ge.-5Ø.)ei2=mmdei(iopt,arg2,ier)
       pduls=-Ø.5*(ei1)+Ø.5*(ei2)
       go to 2Ø
 10    pduls=-Ø.5*ei1
 20    return
       end
```

STORAGE AND SKIN TYPE CURVE INCLUDING STEHFEST SUBROUTINE

```
c
c This program generates td versus Pd for the Storage and Skin Type Curve of
c Agarwal, Al-Hussainy and Ramey.
c
c Variables used:
c cd=dimesionless storage constant
c mmbskØ=imsl routine which calculates Bessel function KØ
c mmbsk1=imsld routine which calculates Bessel funcion K1
c n=number of iterations in the Stehfest routine (recommend n=1Ø)
c nskin-number of skin values desired
c pd=dimensionless pressure
c pwdl=Stehfest function that inverts Laplace equation
c pwd=function called within Stehfest that contains Laplace equation to
c be inverted
c skin=skin effect value
c td=dimensionless time
c
      Implicit real*8 (a-h,o-z)
      common cd,skin
      double precision pwdl
c
c
      OPEN (UNIT=3,FILE='agar.out')
      REWIND (UNIT=3)
      OPEN(UNIT=4, FILE='agar.in', STATUS='old', ACCESS='sequential')
      REWIND(UNIT=4)
c
c Here the desired storage constant, the number of iterations desired in
c the Stehfest algorithm subroutine, and the number of skin values are entered.
c
      read (4,*)   cd,n,nskin
c
c Skin is read, td values are generated, and Stehfest function called to
c calculate pd for each value of td.
c
      do 1650 jjj=1,nskin
          read(4,*)skin
c
          nump=141
          write (3,199)nump
c
      do 20 i=1,2Ø
   do 30 j=1,18
          a=i
          b=j
          td=Ø.ØØØØØ1*(1+(b-1.Ø)*Ø.5)*(1Ø.Ø**a)
          if (td.lt.25.Ø.or.td.gt.1ØØØØØØØØØ) go to 30
          if(skin.lt.Ø.Ø)td=td*(dexp(2*skin))
c         if (td.lt.Ø.ØØ1.or.td.gt.1ØØØØØØ) go to 30
      pd=pwd(td,n)
          if(skin.lt.Ø.Ø)td=td/(dexp(2.*skin))
          write (3,777) td,pd
30        continue
20       continue
777      format (5x,e12.5,5x,e12.5)
c
1650  continue
c
199   format(i3)
      stop
      end
c
c
      function pwdl(arg,i)
```

STORAGE AND SKIN TYPE CURVE INCLUDING STEHFEST SUBROUTINE (CONTINUED)

```
        implicit real*8 (a-h,o-z)
        common cd,skin
        double precision mmbskØ, mmbskl
        integer iopt,ier
        if(skin.lt.Ø.Ø)go to 10
        go to 28
10      s=Ø.Ø
        c=cd*(dexp(2.*skin))
        go to 30
20      s=skin
        c=cd
30      iopt=1
         aarg=dsqrt(arg)
          akØ=mmbskØ(iopt,aarg,ier)
          ak1=mmbsk1(iopt,aarg,ier)
          anum=akØ+s*ak1*aarg
      denom=(c*arg*(akØ+s*aarg*ak1)+aarg*ak1)*arg
          pwd1=anum/denom
        return
        end
C                THE STEHFEST ALGORITHM
C            ******************************
C
       FUNCTION PWD(TD,N)
C           THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
C           INVERSE OF F(S).
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION G(5Ø),V(5Ø),H(25)
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).
       IF (N.EQ.M) GO TO 17
       M=N
       DLOGTW=Ø.6931471805599
       NH=N/2
C
C           THE FACTORIALS OF 1 TO N ARE CALCULATED INTO ARRAY G.
       G(1)=1
       DO 1 I=2,N
       G(I)=G(I-1)*I
1      CONTINUE
C
C           TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H.
       H(1)=2./G(NH-1)
       DO 6 I=2,NH
       FI=I
       IF(I-NH) 4,5,6
4      H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
       GO TO 6
5      H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
6      CONTINUE
C
C           THE TERMS (-1)**NH+1 ARE CALCULATED.
C           FIRST THE TERM FOR I=1
       SN=2*(NH-NH/2*2)-1
C
C           THE REST OF THE SN'S ARECALCULATED IN THE MAIN RUTINE.
C
C
C           THE ARRAY V(I) IS CALCULATED.
       DO 7 I=1,N
C
C           FIRST SET V(I)=Ø
```

STORAGE AND SKIN TYPE CURVE INCLUDING STEHFEST SUBROUTINE (CONTINUED)

```
      V(I)=Ø.
C
C          THE LIMITS FOR K ARE ESTABLISHED.
C          THE LOWER LIMIT IS K1=INTEG((I+1/2))
      K1=(I+1)/2
C
C          THE UPPER LIMIT IS K2=MIN(I,N/2)
      K2=I
      IF (K2-NH) 8,8,9
9     K2=NH
C
C          THE SUMMATION TERM IN V(I) IS CALCULATED.
8     DO 10 K=K1,K2
      IF (2*K-I) 12.13.12
12    IF (I-K) 11,14,11
11    V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GO TO 10
13    V(I)=V(I)+H(K)/G(I-K)
      GO TO 10
14    V(I)=V(I)+H(K)/G(2*K-I)
10    CONTINUE
C
C          THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C          ACCORDING TO SN.
      V(I)=SN*V(I)
C      WRITE (6,21) I,V(I)
c21    FORMAT (5X,'I=',I5,5X,'V(I)=',D2Ø.9)
C
C          THE TERM SN CHANGES ITS SIGN EACH ITERATION.
      SN=-SN
7     CONTINUE
C
C          THE NUMERICAL APPROXIMATION IS CALCULATED.
17    PWD=Ø.
      A=DLOGTW/TD
      DO 15 I=1,N
      ARG=A*I
      PWD=PWD+V(I)*PWDL(ARG,I)
15    CONTINUE
      PWD=PWD*A
18    RETURN
      END
```

SLUG TEST TYPE CURVE

```
c
c This program generates PDR versus tD/CD or (1-PDR) versus tD/CD for
c for the slug test type curves of Ramcy, Agarwal, and Martin.  The
c Stehfest subroutine is used to invert the Laplace equation.
c
c Variables used:
c
c cde2s=storage constant*(e**2*SKIN)
c cdstart=cde2s value where calculations begin
c cdstop=cde2s value where calulation end
c n=number of iterations in the Stehfest routine
c ncurves=number of curves
c pd1=dimensionless pressure for a slug test
c pd2=1-dimensionless pressure for a slug test
c skin=skin effect value
c tdcd=dimensionless tine/storage constant
c

c
      implicit real*8(a-h,o-z)
      common skin
      dimension cde2s(100),cdstrt(100),cdstop(100),tdcd(1000),
     &pd1(1000),pd2(1000),ski(100)
      character nin*10
      character nout1*10
      character nout2*10
c
      write(6,1000)
 1000 format(/,'SLUG TEST TYPE CURVE GENERATOR',/,80('*'),//,
     &'INPUT FILE NAME FOR VALUES OF Cd*(e**2s) ?')
      read(5,*)nin
c
      write(6,1100)
 1100 format(/,'Pd OUTPUT FILE NAME ?')
      read(5,*)nout1
c
      write(6,1200)
 1200 format(/,'(1-PD) OUTPUT FILE NAME ?')
      read(5,*)nout2
c
      open(1,file=nin)
      open(2,file=nout1)
      open(3,file=nout2)
c
      read(1,*)ncurves,n
c
      do 10 i=1, ncurves
        read(1,*)cde2s(i),cdstrt(i),cdstop(i),ski(i)
   10 continue
c
c
      npts=101
      m=777
c
      do 20 i=1,ncurves
        cd=cde2s(i)
        skin=ski(i)
        ncount=0
          do 30 j=1,npts
          tdcdlog = -2. + (j-1)/((npts-1)/5.)
          tdcd(j) = 10**tdcdlog
          td = tdcd(j) * cd
          pd1(j) = pwd(td,n,m,cd)
          pd2(j) = 1. - pd1(j)
```

SLUG TEST TYPE CURVE (CONTINUED)

```
          if(tdcd(j).lt.cdstrt(i))go to 30
          if(tdcd(j).gt.cdstop(i))go to 30
          ncount = ncount + 1
   30   continus
c
        write(2,2000)ncount
        wrtie(3,2000)ncount
 2000   format(i5)
c
        do 35 j=1,npts
          if(tdcd(j).lt.cdstrt(i))go to 35
          if(tdcd(j).gt.cdstop(i))go to 35
          write(2,2010)tdcd(j),pd1(j)
          write(3,2010)tdcd(j),pd2(j)
 2010     format(e15.8,10x,e15.8)
   35   continue
c
c
   20 continue
c
      write(6,1500)
 1500 format(//,20('*'),'RUN FINISHED',20('*'))
c
      stop
      end
c
c
c
      function pwd1(s,i,cd)
      implicit read*8(a-h,o-z)
      common skin
      double precision mmbsk0, mmbskl
      iopt=1
      rs = sqrt(S)
      a0=mmbsk0(iopt,rs,ier)
      a1=mmbsk1(iopt,rs,ier)
c
      anum=(a0+skin*rs*a1)*cd
      denom=rs*a1+cd*s*(a0+skin*rs*a1)
c
      pwd1=anum/denom
c
      return
      end
c
```

INFINITE SYSTEM PORTION OF RADIAL FLOW,

CONSTANT TERMINAL RATE CASE TYPE CURVE

```
c This program calculates Pd versus td at the well for
c radial flow in an infinite reservoir.
c
c Variables used:
c
c n=number of iteration in the Stehfest routine
c ppd=dimensionless pressure
c pwd1=Stehfest functton that Inverts Laplace equation
c pwd=function called within Stehfest that contairs Laplace equation to
c be inverted
c td=dimensionless time
c ttd=dimensionless time
c
      Implicit real*8 (a-h,o-z)
c
      dimension ttd(5ØØ),ppd(5ØØ)
c
      n=1Ø
      nn-0
      npts=161
      do 20 i=1,npts
          tdlog=-3.+(i-1.)/((npts-1.)/8.)
          td=1Ø.Ø**tdlog
          if (td.1t.Ø.ØØ1.or.td.gt.1ØØØØØØØ) go to 20
          nn=nn+1
          ttd(nn)=td
      ppd(nn)=pwd(td,n)
28      continue
c
        write(6,776)nn
776     format(5x,i5)
        do 124 iii=1,nn
           write(6,777) ttd(iii),ppd(iii)
124     continue
777     format (5x,e12.5,5x,e12.5)
c
      stop
      end
c
c
      function pwd1(arg,i)
      implicit real*8 (a-h,o-z)
      double precision mmbskØ, mmbskl
      integer iopt,ier
      iopt=1
        aarg=dsqrt(arg)
        akØ=mmbskØ(iopt,aarg,ier)
        ak1=mmbsk1(iopt,aarg,ier)
        pwd1=akØ/((arg**1.5)*ak1)
      return
      end
```

NO FLOW CIRCULAR BOUNDARY PORTION OF RADIAL FLOW,

CONSTANT TERMINAL RATE CASE TYPE CURVE

```
c
c This program calculates Pd versus td at the well for a finite circular
c reservoir with no flow at the external boundary.
c
c Variables used:
c n=number 3f iterstlons in the Stehfett routine
c ppd=dimensionless pressure
c pwdl=Stehfest function that inverts Laplace equation
c pwd-function called within Stehfest that contaires Laplace equation to
c be inverted
c rd=radius of the well divided by the radius of the reservoir
c td=dimensionless time
c ttd=dimensionless time
c
        Implicit real*8 (a-h,o-z)
        common rdn
        dimension rd(5Ø),ttd(5ØØ),ppd(5ØØ)
c
         read(5,*)nrd
         do 1Ø i=1,nrd
            read(5,*)rd(i)
1Ø       continue

c
       do 15 k=1,nrd
           rdn=rd(k)
c
        n=8
        if(rd(k).le.1ØØ.Ø)tdmin=1Ø.Ø
        if(rd(k).gt.1ØØ.Ø)tdmin=1Ø.Ø*rd(k)
        do 123 ii=1,5ØØ
           ttd(ii)=Ø.Ø
           ppd(ii)=Ø.Ø
123     continue
        nn=Ø
        npts=161
        do 2Ø i=1,npts
            tdlog=-1+(i-1)/((npts-1)/8.)
            td=1Ø.**tdlog
            if (td.lt.tdmin.or.td.gt.1ØØØØØØØØØ) go to 28
            nn=nn+1
            ttd(nn)=td
        ppd(nn)=pwd(ttd(nn),n)
28        continue
c
          write (6,776) nn
776       format (5x,i5)
          do 124 iii=1,nn
             write (6,777) ttd(iii),ppd(iii)
124       continue
777       format (5x,e12.5,5x,e12.5)
13      continue
c
c
         stop
         end
c
c
         functton pwdl(arg,i)
         implicit real*8 (a-h,o-z)
         common rdn
         double precision mmbskØ, mmbsk1,mmbsiØ,mmbsi1
         integer iopt,ier
         iopt=1
```

NO FLOW CIRCULAR BOUNDARY PORTION OF RADIAL FLOW,

CONSTANT TERMINAL RATE CASE TYPE CURVE (CONTINUED)

```
  aarg=dsqrt(arg)
  rarg=rdn*aarg
  rk1=mmbsk1(iopt,rarg,ier)
  akØ=mmbskØ(iopt,aarg,ier)
  ak1=mmbsk1(iopt,aarg,ier)
  ri1=mmbsi1(iopt,rarg,ier)
  aiØ=mmbsiØ(iopt,aarg,ier)
  ai1=mmbsi1(iopt,aarg,ier)
  anum=rk1*aiØ+ri1*akØ
  denom=(arg**1.5)*(ri1*ak1-rk1*ai1)
  pwd1=anum/denom
return
end
```

CONSTANT PRESSURE CIRCULAR BOUNDARY PORTION OF RADIAL FLOW, CONSTANT TERMINAL RATE CASE TYPE CURVE

```
c
c This program calculates Pd versus td at the well for a finite circular
c reservoir with constant pressure at the external boundary.
c
c Variables used:
c n=number of iterations in the Stehfcst routine
c ppd-dimensionless pressure
c pwdl=Stehfest function that inverts Laplace equation
c pwd=function called withn Stehfest that contains Laplace equation to
c be inverted
c rd=radius of the well divided by radius of the reservoir
c td=dimensionless time
c ttd=dimensionless time
c
       Implicit real*8 (a-h,o-z)
       common rdn
       dimension rd(5Ø).ttd(5ØØ),ppd(5ØØ)
c
        read(5,*)nrd
        do 10 i=1,nrd
           read(5,*)rd(i)
10      continue
c
      do 15 k=1,nrd
          rdn=rd(k)
c
       n=8
       if(rd(k).le.1ØØ.Ø)tdmin=1Ø.Ø
       if(rd(k).gt.1ØØ.Ø)tdmin=1Ø.Ø*rd(k)
       do 123 ii=1,5ØØ
          ttd(ii)=Ø.Ø
          ppd(ii)=Ø.Ø
123    continue
       nn=Ø
       npts=161
       do 20 i=1,npts
           tdlog=-1+(i-1)/((npts-1)/8.)
           td=1Ø.Ø**tdlog
           if (td.lt.tdmin.or.td.gt.1ØØØØØØØØØ) go to 20
           nn=nn+1
           ttd(nn)=td
           ppd(nn)=pwd(ttd(nn),n)
26       continue
c
         write(6,776)nn
776      format(5x,i5)
         do 124 iii=1,nn
            write(6,777) ttd(iii),ppd(iii)
124      continue
777      format (5x,e12.5,5x,e12.5)
15    continue
c
c
        stop
        end
c
c
        function pwdl(arg,i)
        implicit real*8 (a-h,o-z)
        common rdn
        double precision mmbskØ, mmbsk1,mmbsiØ,mmbsi1
        integer iopt,ier
        iopt=1
          aarg=dsqrt(arg)
```

CONSTANT PRESSURE CIRCULAR BOUNDARY PORTION OF RADIAL FLOW,

CONSTANT TERMINAL RATE CASE TYPE CURVE (CONTINUED)

```
    rarg=rdn*aarg
    rkØ=mmbskØ(iopt,rarg,ier)
    akØ=mmbskØ(iopt,aarg,ier)
    ak1=mmbsk1(iopt,aarg,ier)
    riØ=mmbsiØ(iopt,rarg,ier)
    aiØ=mmbsiØ(iopt,aarg,ier)
    ai1=mmbsi1(iopt,aarg,ier)
    anum=(-rkØ*aiØ)+(akØ*riØ)
    denom=(arg**1.5)*(rkØ*ai1+riØ*ak1)
    pwd1=anum/denom
  return
  end
```

INFINITE PORTION OF UNIFORM FLUX VERTICAL FRACTURE TYPE CURVE

```
c  This program generates Pd versus td for the infinite line
c  of the uniform flux vertical fracture type curve.
c
c Variables used:
c
c pd=dimensionless pressure
c ppd=dimensionless pressure
c td=dimensionless time
c ttd=dimensionless time
       Implicit real*8 (a-h,o-z)
       Dimension ttd(500),ppd(500)
c
c
       OPEN (UNIT=3,FILE='infuf.out')
       REWIND (UNIT=3)
c
c
c
       api=3.1415927
       nn=0
c
c
          do 10 i=1,6
             do 20 j=1,20
                tdlog=-3.+i+(j-1)/20.
                td=10.**tdlog
                if(td.gt.1000.)go to 10
                pd=pwd(td)
21              nn=nn+1
                ttd(nn)=td
                ppd(nn)=pd
20           continue
10        continue
          write(3,200)nn
          do 30 i=1,nn
             write(3,100)ttd(i),ppd(i)
30        continue
c
100     format (5x,e12.5,5x,e12.5)
200     format (5x,i5)
c
       stop
       end
c
c
       function pwd(td)
       implicit real*8 (a-h,o-z)
       double precision derf,mmdei
       api=3.1415927
       iopt=1
       arg1=1./(2.*dsqrt(td))
       arg2=-1./(4.*td)
       pwd=dsqrt(api*td)*derf(arg1)-0.5*mmdei(iopt,arg2,ier)
        return
        end
c
```

INFINITE PORTION OF INFINITE CONDUCTIVITY VERTICAL FRACTURE TYPE CURVE

```
c  This program generates Pd versus td for the infinite line
c  of the infinite conductivity vertical fracture type curve.
c
c Varlables used:
c pd=dimensionless pressure
c ppd=dimensionless pressure
c td=cimensionless time
c ttd=dimensionless time

       Implicit real*8 (a-h,o-z)
       Dimension ttd(5ØØ),ppd(5ØØ)
c
c
       OPEN (UNIT=3,FILE='infic.out')
       REWIND (UNIT=3)
c
c
c
       api=3.1415927
       nn=Ø
c
c
          do 10 i=1,6
             do 20 j=1,2Ø
                tdlog=-3.+i+(j-1)/2Ø.
                td=1Ø.**tdlog
                if(td.gt.1ØØØ.)go to 10
                pd=pwd(td)
21              nn=nn+1
                ttd(nn)=td
                ppd(nn)=pd
2Ø           continue
1Ø        continue
          write(3,2ØØ)nn
          do 30 i=1,nn
             write(3,1ØØ)ttd(i),ppd(i)
38        continue
c
1ØØ     format (5x,e12.5,5x,e12.5)
2a0     format (5x,i5)
c
       stop
       end
c
c
       functton pwd(td)
        implicit real*8 (a-h,o-z)
       double precision derf,mmdei
       api=3.1415927
       fopt=1
       arg1=Ø.134/(dsqrt(td))
       arg2=Ø.866/(dsqrt(td))
       arg3=-Ø.Ø18/td
       arg4=-Ø.75Ø/td
       term1=derf(arg1)+derf(arg2)
       term2=mmdei(iopt,arg3,ier)
       term3=mmdei(iopt,arg4,ier)
       pwd=Ø.5*dsqrt(api*td)*term1-Ø.Ø67*term2-Ø.433*term3
         return
         end
 c
```

FINITE SYSTEM PORTION FOR EITHER UNIFORM FLUX OR INFINITE CONDUCTIVITY VERTICAL FRACTURE TYPE CURVES

```
c This program generates PD versus tD using any value of Xe/Xf
c for either the uniform flux or infinite conductivfty vertical
c fracture type curve.
c
c Variables used:
c pd=dimensionless pressure
c tda=dimensionless time bases on a square drainage area
c xexf=fracture half length/square reservoir half dimension
c xd=dimensionless coordinate based on fracture length, xd=Ø.Ø for
c uniform flux vertical fracture, xd=Ø.732 for infinite conductivity
c vertical fracture
c
      Implicit real*8 (a-h,o-z)
c
c
      OPEN (UNIT=3,FILE='vf.out')
      REWIND (UNIT=3)
      OPEN(UNIT=4, FILE='vf.in', STATUS='old', ACCESS='sequential')
      REWIND(UNIT=4)
c
c                 reading the input parameters
      read (4,*)    xexf,xd
c
c
      api=3.1415927
c
c
      algo1=Ø.Ø
      algo2=Ø.Ø
      pd=Ø.0
      tdao=Ø.Ø
      do 10 i=1,1Ø
      do 20 j=1,1ØØ
          tdalog=-1Ø.+i+(j-1)/1ØØ.
          tda=1Ø.**tdalog
          dtda=tda-tdao
          sub1=fun1(tda)
          sub2=fun2 (tda,xexf,xd)
          algo2=(1.Ø+sub1)*(1.Ø+sub2)
          pd=pd+2*api*dtda*(algo2+algo1)/2.Ø
          write (3,1ØØ) tda,pd
          algo1=algo2
          tdao=tda
 25     continue
 10     continue
1ØØ     format (5x,e12.5,5x,e12.5)
c
      stop
      end
c
c
      function fun1(tda)
       implicit real*8 (a-h,o-z)
      api=3.1415927
      totsum=Ø.Ø
         do 200 i=1,1ØØØØ
          x=4.Ø*i*i*api*api*tda
          if (x.gt.5Ø) go to 202
          sum1=dexp(-x)
          go to 203
 202      sum1=Ø.Ø
 203      totsum1=totsum+sum1
          delta=(totsum1-totsum)
          if (delta.eq.Ø.Ø) go to 200
```

FINITE SYSTEM PORTION FOR EITHER UNIFORM FLUX OR INFINITE CONDUCTIVITY VERTICAL FRACTURE TYPE CURVES (CONTINUED)

```
            delta=delta/totsum1
            delt=abs(delta)
            if (delt.lt.0.80000001) go to 201
            totsum=totsum1
210     continue
281     fun1=2.Ø*totsum1
        return
        end
c
c
c
      function fun2(tda,xexf,xd)
      implicit real*8 (a-h,o-z)
      api=3.1415927
      totsum=Ø.Ø
         do 3ØØ i=1,1ØØØØ
            x=4.Ø*i*i*api*api*tda
            y=api*i*xexf
            z=y*xd
            yy=dsin(y)/y
            zz=dcos(z)
c
            if (x.gt.5Ø) go to 302
            sum1=dexp(-x) *yy*zz
            go to 383
3Ø2         sum1=Ø.Ø
313         totsum1=totsum+sum1
            delta=(totsum1-totsum)
            if (delta.eq.Ø.Ø) go to 300
            delta=delta/totsum1
            delt=abs(delta)
            if (delt.lt.0.00000001)  go to 381
            totsum=totsum1
300   continue
311    fun2=2.Ø*totsum1
      return
      end
```

APPENDIX B

SELECTED DATA FROM THAT GENERATED BY THE COMPUTER PROGRAMS FOR PRODUCTION OF THE TYPE CURVES

SELECTED LINE SOURCE SOLUTION VALUES

$\frac{t_D}{r_D^2}$	P_D
0.1000	0.0125
0.1413	0.0338
0.1995	0.0729
0.2818	0.1331
0.3981	0.2149
0.5623	0.3165
0.7943	0.4352
1.0000	0.5221
1.4125	0.6620
1.9953	0.8107
2.8184	0.9660
3.9811	1.1262
5.6234	1.2900
7.9433	1.4563
10.0000	1.5683
14.1250	1.7373
19.9530	1.9075
28.1840	2.0783
39.8110	2.2497
66.2340	2.4215
79.4330	2.5936
100.0000	2.7084
125.8900	2.8232
177.8300	2.9956
316.2300	3.2832
446.6800	3.4557
830.9600	3.6284
891.2500	3.8010
1000.0000	3.8585
1412.5000	4.0312
1995.3000	4.2039
2818.4000	4.3765
3981.1000	4.5492
5823.4000	4.7219
7943.3000	4.8946
10000.0000	5.0097

SELECTED DRAWDOWN TEST SEALED LINEAR FAULT VALUES

$\frac{r_2}{r_1}$	$\frac{t_D}{r_D^2}$	P_D
1.6	0.1000	0.0127
	1.0000	0.7674
	10.0000	2.7464
	100.0000	5.0128
	1000.0000	7.3118
	10000.0000	9.6140
2.0	0.1000	0.0125
	1.0000	0.6318
	10.0000	2.4797
	100.0000	4.7273
	1000.0000	7.0243
	10000.0000	9.3263
3.0	0.1000	0.0125
	1.0000	0.6395
	10.0000	2.1319
	100.0000	4.3281
	1000.0000	6.6195
	10000.0000	8.9209
4.0	0.1000	0.0125
	1.0000	0.5240
	10.0000	1.9194
	100.0000	4.0490
	1000.0000	6.3327
	10000.0000	8.6333
6.0	0.1000	0.0125
	1.0000	0.6221
	10.0000	1.6983
	100.0000	3.6677
	1000.0000	6.9297
	10000.0000	8.2281
10.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.5807
	100.0000	3.2306
	1000.0000	6.4268
	10000.0000	7.7181

SELECTED DRAWDOWN TEST SEALED LINEAR FAULT VALUES (CONTINUED)

$\frac{r_2}{r_1}$	$\frac{t_D}{r_D^2}$	P_D
20.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.6683
	100.0000	2.8181
	1000.0000	4.7700
	10000.0000	7.0287
40.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.5683
	100.0000	2.7103
	1000.0000	4.2097
	10000.0000	6.3504
100.0	0.1000	0.0125
	1.0000	0.6221
	10.0000	1.5683
	100.0000	2.7084
	1000.0000	3.8710
	10000.0000	5.5319

SELECTED DRAWDOWN TEST CONSTANT PRESSURE LINEAR BOUNDARY VALUES

$\frac{r_2}{r_1}$	$\frac{t_D}{r_D^2}$	p_D
1.6	0.1000	0.0122
	1.0000	0.2769
	10.0000	0.3902
	100.0000	0.4039
	1000.0000	0.4053
	10000.0000	0.4054
2.0	0.1000	0.0125
	1.0000	0.4124
	10.0000	0.6568
	100.0000	0.6894
	1000.0000	0.6928
	10000.0000	0.6931
3.0	0.1000	0.0125
	1.0000	0.5048
	10.0000	1.0046
	100.0000	1.0887
	1000.0000	1.0976
	10000.0000	1.0985
4.0	0.1000	0.0126
	1.0000	0.6203
	10.0000	1.2171
	100.0000	1.3677
	1000.0000	1.3844
	10000.0000	1.3861
6.0	0.1000	0.0125
	1.0000	0.6221
	10.0000	1.4382
	100.0000	1.7490
	1000.0000	1.7074
	10000.0000	1.7913
10.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.6558
	100.0000	2.1862
	1000.0000	2.2903
	10000.0000	2.3013

SELECTED DRAWDOWN TEST CONSTANT PRESSURE LINEAR BOUNDARY VALUES
(CONTINUED)

$\frac{r_2}{r_1}$	$\frac{t_D}{r_D^2}$	p_D
20.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.6683
	100.0000	2.5987
	1000.0000	2.8471
	10000.0000	2.9908
40.0	0.1000	0.0125
	1.0000	0.5221
	10.0000	1.5683
	100.0000	2.7065
	1000.0000	3.5074
	10000.0000	3.6691
100.0	0.1000	0.0126
	1.0000	0.6221
	10.0000	1.6683
	100.0000	2.7084
	1000.0000	3.8461
	10000.0000	4.4876

SELECTED DRAWDOWN AND BUILDUP TEST SEALED LINEAR FAULT VALUES FOR $r_2 / r_1 = 2.0$

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	p_D
0.1	0.1000	0.0127
	0.1585	0.0465
	0.2512	0.0856
	0.3981	0.0983
	0.6310	0.0886
	1.0000	0.0696
	1.5849	0.0503
	2.5119	0.0345
	3.9811	0.0230
	6.3096	0.0150
	10.0000	0.0096
1.0	1.0000	0.7674
	1.6849	0.6566
	2.5119	0.4128
	3.0811	0.2573
	6.3096	0.1609
	10.0000	0.1010
	16.8490	0.0635
	25.1100	0.0400
	39.8110	0.0252
	63.0060	0.0159
	100.0000	0.0100
10.0	10.0000	2.7464
	15.8490	0.9642
	25.1190	0.4971
	39.8110	0.2859
	63.0060	0.1713
	100.0000	0.1049
	158.4900	0.0650
	261.1000	0.0406
	398.1100	0.0254
	630.9600	0.0160
	1000.0000	0.0100

SELECTED DRAWDOWN AND BUILDUP TEST SEALED LINEAR FAULT VALUES
FOR r_2/r_1 = 2.0(CONTINUED)

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	p_D
100.0	100.0000	5.0128
	168.4900	0.9925
	251.1900	0.5066
	398.1100	0.2889
	630.0600	0.1724
	1000.0000	0.1053
	1584.9000	0.0652
	2511.9000	0.0406
	3981.1000	0.0254
	6309.6000	0.0160
	10000.0000	0.0100
1000.0	1000.0000	7.3118
	1584.9000	0.9964
	2511.9000	0.6076
	3981.1000	0.2892
	6309.6000	0.1725
	10000.0000	0.1054

SELECTED DRAWDOWN AND BUILDUP TEST SEALED LINEAR FAULT VALUES FOR $r_2/r_1 = 10.0$

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	p_D
0.1	0.1000	0.0125
	0.1585	0.0432
	0.2512	0.0706
	0.3981	0.0700
	0.6310	0.0560
	1.0000	0.0405
	1.5849	0.0278
	2.5119	0.0189
	3.9811	0.0132
	6.3096	0.0096
1.0	1.0000	0.5221
	1.5849	0.3817
	2.6119	0.2260
	3.9811	0.1452
	6.3096	0.1009
	10.0000	0.0717
	16.8490	0.0502
	26.1190	0.0342
	39.8110	0.0228
	63.0960	0.0149
	100.0000	0.0096
10.0	10.0000	1.6983
	16.8490	0.6805
	25.1190	0.4101
	39.8110	0.2549
	63.0960	0.1597
	100.0000	0.1005
	168.4900	0.0633
	261.1900	0.0399
	3g8.1100	0.0251
	630.9600	0.0159
	1000.0000	0.0100

SELECTED DRAWDOWN AND BUILDUP TEST SEALED LINEAR FAULT VALUES FOR r_2/r_1 = 10.0 (CONTINUED)

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	P_D
100.0	100.0000	3.6677
	158.4900	0.9494
	251.1900	0.4958
	398.1100	0.2854
	630.9600	0.1712
	1000.0000	0.1048
	1684.9000	0.0650
	2511.9000	0.0405
	3981.1000	0.0254
	6309.6000	0.0160
	10000.0000	0.0100
1000.0	1000.0000	5.9297
	1684.9000	0.9919
	2511.9000	0.5065
	3981.1000	0.2889
	6308.6000	0.1724
	10000.0000	0.1053

SELECTED DRAWDOWN AND BUILDUP TEST CONSTANT PRESSURE LINEAR BOUNDARY VALUES FOR $r_2 / r_1 = 2.0$

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	p_D
0.1	0.1000	0.0122
	0.1259	0.0239
	0.1585	0.0400
	0.1995	0.0527
	0.2512	0.0557
	0.3162	0.0506
	0.3981	0.0416
	0.5012	0.0319
	0.6310	0.0234
	0.7943	0.0165
	1.0000	0.0114
	1.2589	0.0077
1.0	1.0000	0.2769
	1.2589	0.2023
	1.5849	0.1062
	1.9953	0.0582
	2.5119	0.0332
	3.1623	0.0195
	6.1286	0.0068
10.0	10.0000	0.3902
	12.6890	0.0436
	15.8490	0.0161
	19.9530	0.0076
100.0	100.0000	0.4039
	120.2300	0.0064
1000.0	1000.0000	0.4053
	1023.3000	0.0065

SELECTED DRAWDOWN AND BUILDUP TEST CONSTANT PRESSURE LINEAR BOUNDARY VALUES FOR $r_2 / r_1 = 10.0$

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	P_D
0.1	0.1000	0.0125
	0.1259	0.0249
	0.1585	0.0432
	0.1995	0.0606
	0.2512	0.0706
	0.3162	0.0730
	0.3981	0.0700
	0.5012	0.0637
	0.6310	0.0560
	0.7943	0.0480
	1.0000	0.0405
	1.2589	0.0336
	1.5849	0.0276
	1.9953	0.0224
	2.6119	0.0178
	3.1623	0.0139
	3.9811	0.0106
	6.0119	0.0079
	1.0000	0.5221
	1.2589	0.4979
	1.6849	0.3811
	1.8963	0.2894
	2.6119	0.2201
	2.5704	0.2141
	3.2359	0.1615
	4.0738	0.1200
	5.1286	0.0873
	6.4565	0.0622
	8.3176	0.0418
	10.0000	0.0309
	12.6890	0.0209
	16.8490	0.0139
	19.9530	0.0092
	25.1190	0.0060
10.0	10.0000	1.4382
	12.5890	0.5754
	16.8490	0.2898
	19.9530	0.1587
	25.1190	0.0910
	31.6230	0.0537

SELECTED DRAWDOWN AND BUILDUP TEST CONSTANT PRESSURE LINEAR BOUNDARY VALUES FOR r_2 / r_1 = 10.0 (CONTINUED)

$\frac{t_{pD}}{r_D^2}$	$\frac{t_D}{r_D^2}$	P_D
10.0	39.8110	0.0323
	50.1190	0.0196
	63.0960	0.0121
	79.4330	0.0075
	87.0960	0.0062
100.0	100.0000	1.7490
	125.8900	0.1208
	158.4900	0.0447
	199.5300	0.0213
	251.1900	0.0112
	316.2300	0.0063
1000.0	1000.0000	1.7874
	1258.9000	0.0133
	1479.1000	0.0061

SELECTED STORAGE AND SKIN TYPE CURVE VALUES
FOR C_D = 100.0

S	t_D	P_D
-6.0	60.0	0.0486
	100.0	0.0698
	600.0	0.1557
	1000.0	0.2165
	5000.0	0.4447
	10000.0	0.5914
	60000.0	1.0647
	100000.0	1.3233
	500000.0	2.0171
	1000000.0	2.3421
	6000000.0	3.1248
	10000000.0	3.4679
	50000000.0	4.2693
	100000000.0	4.6154
	600000000.0	5.4197
	1000000000.0	5.7662
0.0	50.0	0.4366
	100.0	0.7976
	500.0	2.4356
	1000.0	3.2676
	5000.0	4.5586
	10000.0	4.9566
	60000.0	5.8027
	100000.0	6.1648
	500000.0	6.9643
	1000000.0	7.3116
	6000000.0	8.1168
	10000000.0	8.4635
	50000000.0	9.2683
	100000000.0	8.6140
	500000000.0	10.4200
	1000000000.0	10.7660
5.0	50.0	0.4819
	100.0	0.9324
	600.0	3.7007
	1000.0	5.8019
	6000.0	8.3852
	10000.0	9.8924
	50000.0	10.7920
	100000.0	11.1500
	600000.0	11.9630
	1000000.0	12.3110

SELECTED STORAGE AND SKIN TYPE CURVE VALUES FOR C_D = 100.0 (CONTINUED)

S	t_D	P_D
6.0	6000000.0	13.1170
	10000000.0	13.4630
	50000000.0	14.2680
	100000000.0	14.6150
	600000000.0	16.4200
	1000000000.0	15.7660
10.0	60.0	0.4895
	100.0	0.9597
	600.0	4.1413
	1000.0	7.0149
	5000.0	13.8500
	10000.0	14.8020
	60000.0	15.7810
	100000.0	16.1450
	500000.0	16.9620
	1000000.0	17.3110
	5000000.0	18.1170
	50000000.0	19.2680
	100000000.0	19.6150
	500000000.0	20.4200
	1000000000.0	20.7660
20.0	60.0	0.4943
	100.0	0.9777
	500.0	4.4901
	1000.0	8.1222
	6000.0	21.0960
	10000.0	24.2400
	50000.0	25.7560
	100000.0	26.1340
	600000.0	26.9600
	1000000.0	27.3100
	5000000.0	28.1160
	10000000.0	28.4630
	50000000.0	29.2680
	100000000.0	29.6150
	600000000.0	30.4200
	1000000000.0	30.7660

SELECTED STORAGE AND SKIN TYPE CURVE VALUES
FOR C_D = 100000.0

S	t_D	P_D
-5.0	60.0	0.0005
	100.0	0.0010
	600.0	0.0049
	1000.0	0.0096
	6000.0	0.0458
	10000.0	0.0879
	60000.0	0.3632
	100000.0	0.6219
	500000.0	1.6087
	1000000.0	2.0895
	5000000.0	3.0598
	10000000.0	3.4324
	60000000.0	4.2607
	100000000.0	4.6108
	600000000.0	6.4186
	1000000000.0	5.7657
0.0	60.0	0.0005
	100.0	0.0010
	600.0	0.0050
	1000.0	0.0100
	6000.0	0.0497
	10000.0	0.0988
	60000.0	0.4763
	100000.0	0.9137
	600000.0	3.4557
	1000000.0	5.2138
	6000000.0	7.8971
	10000000.0	8.3671
	50000000.0	9.2493
	100000000.0	0.6051
	600000000.0	10.4170
	1000000000.0	10.7650
5.0	50.0	0.0005
	100.0	0.0010
	500.0	0.0050
	1000.0	0.0100
	6000.0	0.0499
	10000.0	0.0995
	50000.0	0.4878
	100000.0	0.9536
	600000.0	4.0386
	1000000.0	6.7166

SELECTED STORAGE AND SKIN TYPE CURVE VALUES
FOR C_D = 100000.0 (CONTINUED)

S	t_D	P_D
6.0	1000000.0	6.7166
	6000000.0	12.5150
	10000000.0	13.2890
	60000000.0	14.2380
	100000000.0	14.6000
	500000000.0	15.4160
	1000000000.0	16.7650
10.0	50.0	0.0005
	100.0	0.0010
	600.0	0.0050
	1000.0	0.0100
	6000.0	0.0499
	10000.0	0.0997
	60000.0	0.4918
	100000.0	0.9683
	600000.0	4.3042
	1000000.0	7.5165
	5000000.0	16.6250
	10000000.0	18.1440
	60000000.0	19.2270
	100000000.0	10.6950
	600000000.0	20.4150
	1000000000.0	20.7640
20.0	50.0	0.0005
	100.0	0.0010
	600.0	0.0060
	1000.0	0.0100
	6000.0	0.0499
	10000.0	0.0998
	60000.0	0.4951
	100000.0	0.9806
	600000.0	4.5526
	1000000.0	8.3358
	5000000.0	23.0700
	10000000.0	27.2780
	60000000.0	29.2000
	100000000.0	29.5830
	600000000.0	30.4130
	1000000000.0	30.7630

SELECTED SLUG TEST TYPE CURVE VALUES FOR $C_D e^{2S} = 10^2$

$\frac{t_D}{C_D}$	P_{DR}
0.0100	0.9845
0.0141	0.9807
0.0200	0.9760
0.0282	0.9700
0.0398	0.9623
0.0562	0.9524
0.0794	0.9398
0.1000	0.9295
0.1413	0.9105
0.1995	0.8863
0.2818	0.8555
0.3981	0.8168
0.5623	0.7687
0.7943	0.7098
1.0000	0.6642
1.4125	0.5864
1.9953	0.4986
2.8184	0.4045
3.9811	0.3099
6.6234	0.2223
7.8433	0.1488
10.0000	0.1099
14.1254	0.0672
19.9526	0.0406
28.1838	0.0250
39.8107	0.0160

SELECTED SLUG TEST TYPE CURVE VALUES FOR $C_D e^{2S} = 10^3$

$\frac{t_D}{C_D}$	P_{DR}
0.0100	0.9926
0.0141	0.9905
0.0200	0.9878
0.0282	0.9842
0.0398	0.9795
0.0562	0.9732
0.0794	0.9650
0.1000	0.9581
0.1413	0.9450
0.1995	0.9279
0.2818	0.9054
0.3981	0.8761
0.5623	0.8383
0.7943	0.7902
1.0000	0.7515
1.4125	0.6826
1.9953	0.6004
2.8184	0.5063
3.9811	0.4043
6.6234	0.3018
7.9433	0.2078
10.0000	0.1544
14.1254	0.0925
19.9526	0.0625

SELECTED SLUG TEST TYPE CURVE VALUES FOR $C_D e^{2S} = 10^{10}$

$\frac{t_D}{C_D}$	P_{DR}
0.0316	0.9967
0.0447	0.9955
0.0631	0.9937
0.0891	0.9913
0.1000	0.9903
0.1413	0.9866
0.1995	0.9814
0.2818	0.9743
0.3981	0.9644
0.5623	0.9508
0.7943	0.9323
1.0000	0.9164
1.4125	0.8857
1.9953	0.8447
2.8184	0.7909
3.9811	0.7218
6.6234	0.6359
7.9433	0.5337
10.0000	0.4583
14.1254	0.3401
10.9526	0.2268
28.1838	0.1310
30.8107	0.0657
66.2341	0.0287
70.4328	0.0126

SELECTED SLUG TEST TYPE CURVE VALUES FOR $C_D e^{2S} = 10^{40}$

$\frac{t_D}{C_D}$	P_D
1.0000	0.9786
1.4125	0.9698
1.9953	0.9578
2.8184	0.9412
3.9811	0.9182
6.6234	0.8869
7.9433	0.8446
10.0000	0.8090
14.1254	0.7421
19.9526	0.6574
28.1838	0.5545
39.8107	0.4367
66.2341	0.3124
78.4328	0.1958
100.0000	0.1302
141.2638	0.0686
199.5263	0.0204
281.8382	0.0059

SELECTED INFINITE SYSTEM RADIAL FLOW TYPE CURVE VALUES

t_D	P_D
0.0010	0.0352
0.0016	0.0441
0.0025	0.0553
0.0040	0.0693
0.0063	0.0866
0.0100	0.1081
0.0158	0.1347
0.0251	0.1673
0.0398	0.2072
0.0631	0.2557
0.1000	0.3142
0.1585	0.3842
0.2512	0.4669
0.3981	0.5635
0.6310	0.6751
1.0000	0.8021
1.5849	0.9446
2.5119	1.1019
3.9811	1.2731
6.3096	1.4666
10.0000	1.6509
15.8490	1.8540
26.1190	2.0642
89.8110	2.2800
63.0960	2.4999
100.0000	2.7229
168.4900	2.9481
261.1900	3.1749
308.1100	3.4028
680.9600	3.6314
1000.0000	3.8606
1584.9000	4.0901
2611.9000	4.3199
3981.1000	4.5498
63Q9.6000	4.7799
10000.0000	5.0100
15849.0000	5.2401
25119.0000	5.4703
39811.0000	6.7006
63096.0000	5.9308
100000.0000	6.1610

SELECTED SEALED BOUNDARY RADIAL FLOW TYPE CURVE VALUES FOR r_{eD} = 100.0

t_D	P_D
10.0000	1.6508
14.1250	1.8024
10.9530	1.0583
28.1840	2.1176
39.8110	2.2799
66.2340	2.4445
79.4330	2.6110
100.0000	2.7228
141.2500	2.8915
199.5300	3.0612
281.8400	3.2317
398.1100	3.4027
662,3400	3.5742
794.3300	3.7463
1000.0000	3.8609
1412.5000	4.0328
1995.3000	4.2093
2818.4000	4.4058
3981.1000	4.6505
5623.4000	4.9817
7043.3000	6.4456
10000.0000	5.8567
14125.0000	6.6818
19063.0000	7.8476
28164.0000	0.4942
30811.0000	11.8200
66214.0000	16.1060
79483.0000	19.7470
100000.0000	23.8610
141260.0000	32.1140
199510.0000	43.7710
281840.0000	60.2380
398110.0000	83.4980
562340.0000	116.3500
794390.0000	162.7600
1000000.0000	203.9100
1412600.0000	286.4400
1995300.0000	403.0100
2818400.0000	567.6800
3981100.0000	800.2700
6623400.0000	1128.8000
7943300.0000	1692.9000
10000000.0000	2004.4000

SELECTED CONSTANT PRESSURE BOUNDARY RADIAL FLOW TYPE CURVE VALUES FOR r_{eD} = 100.0

t_D	P_D
10.0000	1.6508
14.1250	1.8024
10.9530	1.9583
28.1840	2.1176
39.8110	2.2799
66.2340	2.4445
79.4330	2.6110
100.0000	2.7228
141.2500	2.8915
199.5300	3.0612
281.8400	3.2317
398.1100	3.4027
662.3400	3.5740
794.3300	3.7456
1000.0000	3.8603
1412.5000	4.0328
1995.3000	4.2016
2818.4000	4.3546
3981.1000	4.4754
6823.4000	4.5532
7843.3000	4.6915
10000.0000	4.6019
14125.0000	4.6063
19863.0000	4.6059
28184.0000	4.6053
36181 1.0000	4.6050
56234.0000	4.6050
79433.0000	4.6051
100000.0000	4.6051
141250.0000	4.6052
199530.0000	4.6052
281840.0000	4.6052
398110.0000	4.6052
562340.0000	4.6052
794330.0000	4.6052
1000000.0000	4.6052
1412600.0000	4.6052
1995300.0000	4.6052
2818400.0000	4.6052
3981100.0000	4.6052
6623400.0000	4.6052
7943300.0000	4.6052
10000000.0000	4.6052

SELECTED INFINITE SYSTEM GENERALIZED RADIAL FLOW TYPE CURVE VALUES

t_D^*	P_D^*
0.0010	0.0352
0.0016	0.0441
0.0025	0.0553
0.0040	0.0693
0.0063	0.0866
0.0100	0.1081
0.0158	0.1347
0.0251	0.1673
0.0398	0.2072
0.0631	0.2557
0.1000	0.3142
0.1585	0.3842
0.2512	0.4669
0.3981	0.5635
0.6310	0.6751
1.0000	0.8021
1.5849	0.9446
2.5119	1.1019
3.9811	1.2731
6.3096	1.4566
10.0000	1.6509
15.8490	1.8540
25.1190	2.0642
39.8110	2.2800
63.0960	2.4999
100.0000	2.7229
158.4900	2.9481
251.1900	3.1749
398.1100	3.4028
630.9600	3.6314
1000.0000	3.8606
1584.9000	4.0901
2511.9000	4.3199
3981.1000	4.5498
6309.6000	4.7799
10000.0000	5.0100
15849.0000	5.2401
25119.0000	5.4703
39811.0000	5.7006
63096.0000	6.9308
100000.0000	6.1610

SELECTED SEALED BOUNDARY GENERALIZED RADIAL FLOW TYPE CURVE VALUES

t_D^*	P_D^*
10.0000	1.6508
14.1250	1.8024
19.9530	1.9583
28.1840	2.1176
39.8110	2.2799
56.2340	2.4445
79.4330	2.6110
100.0000	2.7228
141.2500	2.8915
199.5300	3.0612
281.8400	3.2317
388.1100	3.4027
662.3400	3.5742
784.3300	3.7463
1000.0000	3.8609
1412.5000	4.0328
1985.3000	4.2093
2818.4000	4.4058
3981.1000	4.6505
6623.4000	4.9817
7943.3000	6.4456
10000.0000	6.8567
14125.0000	6.6818
19963.0000	7.8475
28184.0000	9.4942
39811.0000	11.8200
56234.0000	15.1060
79433.0000	19.7470
100000.0000	23.8610
141250.0000	32.1140
199530.0000	43.7710
281840.0000	60.2380
398110.0000	83.4980
562340.0000	116.3500
794330.0000	162.7600
1000000.0000	203.9100
1412500.0000	286.4400
1995300.0000	403.0100
2818400.0000	567.6800
3981100.0000	800.2700
5623400.0000	1128.8000
7943300.0000	1692.9000
10000000.0000	2004.4000

SELECTED CONSTANT PRESSURE BOUNDARY
GENERALIZED RADIAL FLOW TYPE CURVE VALUES

t_D^*	P_D^*
10.0000	1.6508
14.1250	1.8024
19.9530	1.9583
28.1840	2.1176
39.8110	2.2799
66.2340	2.4445
79.4330	2.6110
100.0000	2.7228
141.2500	2.8915
199.5300	3.0612
281.8400	3.2317
398.1100	3.4027
662.3400	3.5740
794.3300	3.7456
1000.0000	3.8603
1412.6000	4.0328
1995.3000	4.2015
2818.4000	4.3546
3981.1000	4.4764
6623.4000	4.5532
7943.3000	4.5915
10000.0000	4.6019
14125.0000	4.6063
19953.0000	4.6059
28184.0000	4.6053
39811.0000	4.6050
66234.0000	4.6050
79433.0000	4.6051
100000.0000	4.6051
141250.0000	4.6052
199530.0000	4.6052
281840.0000	4.6052
398110.0000	4.6052
S62340.0000	4.6052
794330.0000	4.6052
1000000.0000	4.6052
1412500.0000	4.6052
1995300.0000	4.6052
2818400.0000	4.6052
3981100.0000	4.6052
6623400.0000	4.6052
7943300.0000	4.6052
10000000.0000	4.6052

SELECTED UNIFORM FLWX VERTICAL FRACTURE TYPE CURVE VALUES FOR AN INFINITE SYSTEM

t_{Df}	P_D
0.0100	0.1773
0.0141	0.2107
0.0200	0.2504
0.0282	0.2976
0.0398	0.3536
0.0562	0.4202
0.0794	0.4989
0.1000	0.5588
0.1413	0.6600
0.1995	0.7748
0.2818	0.9020
0.3981	1.0397
0.5623	1.1862
0.7943	1.3395
1.0000	1.4447
1.4125	1.6060
1.9953	1.7704
2.8184	1.9372
3.9811	2.1057
5.6234	2.2754
7.9433	2.4459
10.0000	2.5600
14.1250	2.7315
19.9530	2.9033
28.1840	3.0754
39.8110	3.2477
66.2340	3.4200
79.4330	3.5925
100.0000	3.7075
141.2500	3.8801
199.5300	4.0527
281.8400	4.2254
398.1100	4.3980
562.3400	4.5707
794.3300	4.7433
1000.0000	4.8585

SELECTED INFINITE CONDUCTIVITY VERTICAL FRACTURE TYPE CURVE VALUES FOR AN INFINITE SYSTEM

t_{Df}	P_D
0.0100	0.1764
0.0141	0.2084
0.0200	0.2452
0.0282	0.2872
0.0398	0.3348
0.0562	0.3887
0.0794	0.4495
0.1000	0.4944
0.1413	0.5692
0.1995	0.6541
0.2818	0.7504
0.3981	0.8586
0.6623	0.9788
0.7943	1.1100
1.0000	1.2029
1.4125	1.3491
1.9953	1.5021
2.8184	1.6603
3.9811	1.8224
6.6234	1.9875
7.9433	2.1548
10.0000	2.2672
14.1250	2.4367
19.9530	2.6072
28.1840	2.7783
39.8110	2.9499
66.2340	3.1218
78.4330	8.2939
100.0000	3.4088
141.2500	3.5812
199.5300	3.7536
281.8400	3.9262
398.1100	4.0988
562.3400	4.2714
794.3300	4.4440
1000.0000	4.5591

SELECTED UNIFORM FLUX VERTICAL FRACTURE SOLUTION VALUES

$\frac{X_e}{X_f}$	t_{Df}	P_D
1.0	0.0200	0.2506
	0.0400	0.3544
	0.0500	0.3963
	0.1000	0.5605
	0.5000	1.3067
	1.0000	2.0944
	5.0000	8.3776
	10.0000	16.2320
	60.0000	79.0630
	100.0000	157.6000
	600.0000	785.9200
	800.0000	1257.2000
	1000.0000	1571.3000
1.5	0.0504	0.3979
	0.1260	0.6250
	0.5580	1.1979
	1.8000	2.1048
	4.6800	4.1165
	8.0000	7.1315
	52.2000	37.2910
	144.0000	101.3800
	432.0000	302.4400
	676.0000	402.9700
	864.0000	604.0300
2.0	0.1000	0.5587
	0.2000	0.7756
	0.4000	1.0417
	1.0000	1.4562
	4.0000	2.6799
	10.0000	5.0361
	40.0000	16.8170
	100.0000	40.3790
	400.0000	158.1900
	1000.0000	393.8100

SELECTED UNIFORM FLUX VERTICAL FRACTURE SOLUTION VALUES
(CONTINUED)

$\frac{X_e}{X_f}$	t_{Df}	P_D
3.0	0.5040	1.1389
	0.9000	1.3963
	4.3200	2.2448
	115.2000	21.6020
	313.2000	66.1600
	684.0000	120.8800
5.0	2.0000	1.7716
	4.0000	2.1087
	6.0000	2.2202
	10.0000	2.6137
	50.0000	6.1390
	100.0000	8.2806
	600.0000	33.4130
	1000.0000	64.8290
10.0	7.0000	2.3834
	8.0000	2.4495
	10.0000	2.6600
	60.0000	3.4674
	100.0000	4.2573
	600.0000	10.5410
	1000.0000	18.3950
20.0	10.0000	2.6600
	11.1200	2.6127
	66.8000	3.4254
	101.6000	3.7239
	248.0000	4.3513
	684.0000	6.6722
	920.0000	6.9917

SELECTED INFINITE CONDUCTIVITY VERTICAL FRACTURE SOLUTION VALUES

$\frac{X_e}{X_f}$	t_{Df}	P_D
1.0	0.0100	0.1772
	0.0500	0.3963
	0.1000	0.5605
	0.5000	1.3067
	1.0000	2.0944
	6.0000	8.3776
	10.0000	16.2320
	60.0000	79.0630
	100.0000	157.6000
	500.0000	785.9200
	1000.0000	1571.3000
1.5	0.0104	0.1705
	0.0266	0.2797
	0.0589	0.3967
	0.1035	0.5018
	0.5895	1.0436
	1.0350	1.3756
	6.8950	4.7720
	10.3500	7.8821
	68.9500	41.8110
	103.6000	72.9130
	689.5000	412.2100
	751.5000	525.3000
2.0	0.0800	0.4509
	0.1000	0.4945
	0.4000	0.8610
	1.0000	1.2248
	4.0000	2.4375
	10.0000	4.7938
	40.0000	16.5750
	100.0000	40.1370
	400.0000	157.9500
	1000.0000	393.5700

SELECTED INFINITE CONDUCTIVITY VERTICAL FRACTURE SOLUTION VALUES (CONTINUED)

$\frac{X_e}{X_f}$	t_{Df}	p_D
3.0	0.1008	0.4961
	0.1512	0.5852
	0.3528	0.8195
	0.7920	1.1091
	1.2960	1.3131
	5.7600	2.2253
	10.8000	3.1059
	86.4000	16.3010
	136.8000	25.0970
	338.4000	60.2830
	648.0000	114.3200
5.0	0.7000	1.0609
	0.8000	1.1129
	1.0000	1.2030
	6.0000	1.9342
	10.0000	2.3235
	50.0000	4.8482
	100.0000	7.9898
	500.0000	33.1230
	1000.0000	64.5380
10.0	2.0000	1.5033
	4.0000	1.8248
	6.0000	1.9312
	10.0000	2.2674
	60.0000	3.1703
	100.0000	3.9602
	600.0000	10.2430
	1000.0000	18.0970
20.0	8.0000	2.1583
	10.0000	2.2673
	40.0000	2.9525
	80.0000	3.3000
	100.0000	3.4167
	400.0000	4.6508
	800.0000	6.2216
	1000.0000	7.0070

APPENDIX C

WORKING TYPE CURVES