1 of 1

Drawdown Behavior of Gravity Drainage Wells

SUPRI TR 97

By
Jan A. Aasen
Henry J. Ramey, Jr.

October 1993

Work Performed Under Contract No. DE-FG22-93BC14899

Prepared for
U.S. Department of Energy
Assistant Secretary for Fossil Energy

Thomas Reid, Project Manager
Bartlesville Project Office
P.O. Box 1398
Bartlesville, OK 74005

Prepared by
Stanford University
Petroleum Research Institute
Stanford, CA 94305

MASTER

Acknowledgements

The author is grateful for the financial support from the Department of Petroleum Engineering at Stanford University, the SUPRI A Industrial Affiliates Program, and the U.S. Department of Energy through contract DE-FG-22-93BC14689.

Abstract

An analytical solution for drawdown in gravity drainage wells is developed. The free-surface flow is viewed as incompressible, and anisotropy effects are included. The well is a line source well, and the reservoir is infinitely large. The model is valid for small drawdowns.

The uniform wellbore potential inner boundary condition is modelled using the proper Green's function. The discontinuity at the wellbore is solved by introducing a finite skin radius, and the formulation produces a seepage face.

The calculated wellbore flux distribution and wellbore pressures are in fair agreement with results obtained using a numerical gravity drainage simulator. Three distinct flow periods are observed. The wellbore storage period is caused by the moving liquid level, and the duration is short. During the long intermediate flow period, the wellbore pressure is nearly constant. In this period the free surface moves downwards, and the liquid is produced mainly by vertical drainage. At long times the semilog straight line appears. The confined liquid solutions by Theis (1935) and van Everdingen and Hurst (1949) may be used during the pseudoradial flow period if the flowrate is low. New type curves are presented that yield both vertical and horizontal permeabilities.

Table of Contents

Acknowledgements	iii
Abstract	v
List of Figures	ix
1 Introduction	1
2 Literature Review	3
3 Uniform Wellbore Flux	5
3.1 Governing Equations.....	5
3.2 Laplace Domain Solution.....	6
3.3 Pseudoradial Flow.....	7
3.4 Skin and Wellbore Storage.....	10
3.5 Summary.....	15
4 Uniform Wellbore Potential	16
4.1 Green's Function Solution Procedure.....	16
4.2 Flux Distribution.....	18
4.3 Pressure Drawdown Behavior.....	23
4.4 Summary.....	33
5 Discussion and Conclusions	36
Nomenclature	39
Bibliography	41
Appendix	43
A Solution of Differential Equation	43
B Hankel Solution for a Line Source Well	45
C Boulton (1963) Delayed Yield	48
D Numerical Considerations	51
E Computer Program	55

List of Figures

1.1	Gravity drainage drawdown test.	2
3.1	Dimensionless pressure drop at skin radius versus dimensionless vertical flow time for several dimensionless heights.	9
3.2	Pressure drop at skin radius versus flow time for $qB = 50$ bbl/day. Isotropic base case.	11
3.3	Pressure drop at skin radius versus flow time for $qB = 100$ bbl/day. Isotropic base case.	12
3.4	Pressure drop at skin radius versus flow time for $qB = 200$ bbl/day. Isotropic base case.	13
4.1	Dimensionless wellbore pressure drop versus dimensionless vertical flow time for several flowrates.	19
4.2	Dimensionless wellbore height versus dimensionless flowrate at $qB = 50$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes. . .	20
4.3	Dimensionless wellbore height versus dimensionless flowrate at $qB = 100$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes.	21
4.4	Dimensionless wellbore height versus dimensionless flowrate at $qB = 200$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes.	22
4.5	Dimensionless wellbore pressure drop versus dimensionless vertical flow time for gravity drainage and compressible flow. Log-log graph for isotropic base case.	24
4.6	Dimensionless wellbore pressure drop versus dimensionless vertical flow time for gravity drainage and compressible flow. Semilog graph for isotropic base case.	25
4.7	Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 50$ bbl/day.	27
4.8	Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 100$ bbl/day.	28
4.9	Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 200$ bbl/day.	29
4.10	Wellbore pressure drop versus flow time for several flowrates. Base isotropic case with $k = 240$ md.	30
4.11	Wellbore pressure drop versus flow time for several permeabilities. Base isotropic case with $qB = 100$ bbl/day.	31

4.12	Wellbore pressure drop versus flow time for several skin factors. Base isotropic case with $qB = 100$ bbl/day and $k = 240$ md.	32
4.13	Wellbore pressure drop versus flow time for several initial liquid heights. Base isotropic case with $qB = 100$ bbl/day and $k = 240$ md.	34
4.14	Wellbore pressure drop versus flow time for several anisotropy ratios. Base case with $qB = 100$ bbl/day and $k_H = 240$ md.	35
C.1	Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for analytical model and Boulton (1963) delayed yield model.	49
D.1	Cumulative CPU time versus drawdown time for the Stream Layer Model numerical simulator.	54

1. Introduction

Significant low pressure deposits of oil exist in the world. Some of these reservoirs produce by gravity drainage. The results of this study will aid evaluation of well tests for low pressure oil reservoirs as well as for water wells.

The problem considered in this study concerns a vertical well producing a slightly compressible, single phase liquid at a constant pumping rate. The producing formation is assumed horizontal, and the wellbore is exposed to the entire thickness of the formation. The lower formation boundary is an impermeable layer of rock, and the upper boundary is a free surface (watertable) which is initially static. The gas overlying the liquid has zero gauge pressure. As the liquid level in the reservoir drops as a function of time and radial location, the porous medium is drained. The ultimate yield of this process is the product of porosity and the difference between initial and residual liquid saturations. A schematic diagram of the process is shown in Fig. 1.1.

The bulk of research conducted on transient flow in gravity drainage systems has been reported in the groundwater literature. There are currently several analytical models available for matching pressure transient gravity drainage data. These models are based on a similar physical view of the problem. Although gravity drainage concerns a low pressure flow system, many hydrologists consider gravity drainage flow similarly to compressible flow.

Muskat (1937) considered the flow of water through porous media under the action of gravity. Experiments indicated that the liquid level in the wellbore is lower than that in the adjacent sand. This difference is named the seepage height (see Fig. 1.1). The free surface arises naturally for flow systems where gravity is the driving force, and it is difficult to obtain mathematical solutions for this upper boundary condition. An added complication is the surface of seepage. Liquid seeps into the wellbore above the wellbore liquid height in addition to conventional flow into the liquid filled portion of the well.

Ramey *et al.* (1989) analyzed drawdown and buildup data obtained from a gravity drainage field test, and their results indicated that drawdown and buildup mechanics are different. These authors suggested the use of conventional confined flow expressions for field data interpretation if the proper substitution for total compressibility were used. The purpose of the present study is to examine the mechanics of liquid production by gravity drainage.

Figure 1.1: Gravity drainage drawdown test.

2. Literature Review

An early theory describing the mechanism of gravity drainage was developed more than a century ago by Dupuit (1863) and Forchheimer (1886). This approach has become known as the Dupuit-Forchheimer theory of gravity flow systems. The streamlines are considered horizontal, and the liquid velocity is proportional to the slope of the free surface.

Muskat (1937) discussed the Dupuit-Forchheimer theory of gravity-flow systems and pointed out inconsistencies of the model. The horizontal velocity is a function of vertical and radial location. The vertical velocity distribution is not uniform. The average fluid velocity is not proportional to the slope of the free surface. The surface of seepage is not included in this theory.

An important paper in this field was published by Wyckoff *et al.* (1932). The results from gravity drainage sandbox experiments were discussed. The liquid outflow calculated using the Dupuit-Forchheimer formula agreed with the sandbox experiments. A theoretical description of the free-surface condition for gravity flow systems was presented. In the classical text by Muskat (1937), a chapter is devoted to gravity flow systems. Several of the topics covered there apply directly to this research on gravity drainage systems.

Methods for analyzing transient gravity drainage data are scarce in the petroleum engineering literature. The vast majority of the work done on this topic has been reported by groundwater researchers. The earliest pressure transient models were introduced by Boulton (1954a, 1954b, 1963). Prickett (1965) presented a type curve matching procedure based on work by Boulton. A large number of unconfined water wells were analyzed and successfully described in terms of the coefficient of transmissivity, elastic storage, gravity yield and delay index. Prickett (1965) also offered a graphical correlation between the delay index and the grain size of reservoir sands.

Dagan (1964) viewed a free-surface flow system as incompressible flow. Using a perturbation technique, the author derived linearized equations for the gravity flow problem. Dagan (1966) derived the solution and obtained an appropriate Green's function. Dagan (1967) presented the equations of unsteady flow toward a partially penetrating well in an unconfined aquifer, and provided a method for matching pressure data obtained from pumping tests. The gravity drainage solution presented here is based on work by Dagan.

Neuman (1972) and Streltsova (1972) treated an unconfined aquifer as a compressible system. These authors expressed delayed yield mathematically by treating total compressibility and specific yield as constants. Their solutions differ mainly in the degree of mathematical simplification. Neuman (1972) obtained a linearized solution to the three-dimensional problem using a perturbation technique, while Streltsova (1972) reduced the mathematical problem to two dimensions by expressing the solution in terms of the average liquid head.

The influence of the unsaturated zone above the free surface was investigated by Kroszynski and Dagan (1975). They concluded that the assumption of negligible capillary effects used in analytical models is justified.

Ramey *et al.* (1989) analyzed drawdown and buildup data obtained from a gravity drainage field test, and concluded that drawdown and buildup mechanics are different. Flow is incompressible and the apparent compressibility is proportional to the reciprocal of the

initial formation pressure. The ideal gas flow correlation by Jenkins and Aronofsky (1953) was applied to the gravity drainage problem.

Couri (1993) developed a finite difference model to simulate transient free-surface flow. The numerical model results agreed well with results from sandbox experiments. The behavior of gravity drainage wells was analyzed for both drawdown and buildup cases.

An analytical model for drawdown testing of gravity drainage wells is presented in this study. The analytical solution is in good agreement with numerical results, and the computation speed is higher for the analytical solution.

3. Uniform Wellbore Flux

A logical inner boundary wellbore condition for a gravity drainage flow problem is that of uniform wellbore potential. In spite of this, the uniform wellbore flux condition will be employed in this section. The uniform flux equation applies reasonably after a long drawdown time, and this solution will also be useful for the more general analysis presented in Section 4. Equations for wellbore storage and skin are presented, but these will not be employed until the next section.

3.1 Governing Equations

Dagan (1964) obtained a set of linearized equations for free-surface flow through porous media. In terms of conventional oilfield units, the dimensionless parameters may be expressed as:

$$\Phi_D(z_D, r_D, t_{DV}) = -\frac{k_H h_i}{141.2qB\mu}(p + \rho z/144) \quad (3.1)$$

$$z_D = z/h_i \quad (3.2)$$

$$t_{DV} = \frac{263.7 \cdot 10^{-6} k_V t}{\phi \mu p_i^{-1} h_i^2} \quad (3.3)$$

From Dagan (1967):

$$r_D = r\sqrt{k_V/k_H}/h_i \quad (3.4)$$

Combination of Darcys Law and a mass balance for incompressible flow yields the governing partial differential equation. There is no time-derivative since compressibility does not take part in gravity flow.

$$\frac{\partial^2 \Phi_D}{\partial z_D^2} + \frac{1}{r_D} \frac{\partial}{\partial r_D} r_D \frac{\partial \Phi_D}{\partial r_D} = 0 \quad (3.5)$$

The vertical boundaries are $z_D = -1$ and $z_D = 0$. For a line source well, the wellbore is located at r_D equal to zero and the reservoir is considered to be infinitely large. In Fig. 1.1 the wellbore is shown with a finite radius.

The boundary condition for the free surface was taken to be:

$$\frac{\partial \Phi_D(0, r_D, t_{DV})}{\partial t_{DV}} + \frac{\partial \Phi_D(0, r_D, t_{DV})}{\partial z_D} = 0 \quad (3.6)$$

Equation 3.6 states that the rate of change of pressure at the free surface is caused by vertical flow, and was derived using perturbation analysis (Dagan, 1964). At early times, vertical flow near the wellbore will be large since the potential changes rapidly as a function of time. The first term in Eq. 3.6 may be viewed as reservoir storage. The second term represents downward vertical flow. The radial flow component is excluded from Eq. 3.6, and the free surface is fixed at the initial position for all time.

The initial upper boundary condition is:

$$\Phi_D(0, r_D, 0) = 0 \quad (3.7)$$

There is no flow across the lower boundary:

$$\frac{\partial \Phi_D(-1, r_D, t_{DV})}{\partial z_D} = 0 \quad (3.8)$$

The well is a line source well of zero radius, and the flux into the line is uniform:

$$\lim_{r_D \rightarrow 0} r_D \frac{\partial \Phi_D(z_D, r_D, t_{DV})}{\partial r_D} = -1 \quad (3.9)$$

Since the reservoir is infinitely large, the potential at the outer boundary is constant:

$$\Phi_D(z_D, \infty, t_{DV}) = 0 \quad (3.10)$$

Dagan (1966) found the Green's function for this problem, and this solution will be employed in Section 4 to obtain the flux distribution near the wellbore. The solution was expressed in the real time domain, and it was therefore difficult to incorporate wellbore storage effects. In the next subsection the Laplace domain solution will be derived.

3.2 Laplace Domain Solution

Neuman (1968) presented the following convenient formula:

$$\mathcal{H}\left[\frac{1}{r_D} \frac{\partial}{\partial r_D} r_D \frac{\partial \bar{\Phi}_D(z_D, r_D, s)}{\partial r_D}\right] = -\lambda^2 \bar{\Phi}_D(z_D, \lambda, s) - \lim_{r_D \rightarrow 0} r_D \frac{\partial \bar{\Phi}_D(z_D, r_D, s)}{\partial r_D} \quad (3.11)$$

In this equation, the Laplace transformed $\Phi_D(z_D, r_D, t_{DV})$ is $\bar{\Phi}_D(z_D, r_D, s)$. The Hankel transformed $\bar{\Phi}_D(z_D, r_D, s)$ is $\tilde{\Phi}_D(z_D, \lambda, s)$.

From Eqs. 3.5, 3.9 and 3.11, an ordinary differential equation is obtained:

$$d^2\tilde{\Phi}_D/dz_D^2 - \lambda^2\tilde{\Phi}_D + 1/s = 0 \quad (3.12)$$

The first boundary equation results from Eqs. 3.6 and 3.7:

$$d\tilde{\Phi}_D(0, \lambda, s)/dz_D + s\tilde{\Phi}_D(0, \lambda, s) = 0 \quad (3.13)$$

The second boundary equation follows directly from Eq. 3.8:

$$d\tilde{\Phi}_D(-1, \lambda, s)/dz_D = 0 \quad (3.14)$$

The solution to this differential equation is given in Appendix A. The Laplace transform of the potential is:

$$\bar{\Phi}_D(z_D, r_D, s) = \int_0^\infty \frac{1}{s} \frac{J_0(\lambda r_D)}{\lambda} d\lambda - \int_0^\infty \frac{e^{\lambda(z_D+2)} + e^{-\lambda z_D}}{e^{2\lambda}(s+\lambda) + s - \lambda} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (3.15)$$

The standard relationship between the Laplace transform of Φ_D , $\bar{\Phi}_D$, and the Hankel transform of $\bar{\Phi}_D$, $\tilde{\Phi}_D$, was used to produce Eq. 3.15:

$$\bar{\Phi}_D(z_D, r_D, s) = \int_0^\infty \tilde{\Phi}_D(z_D, \lambda, s) \lambda J_0(\lambda r_D) d\lambda \quad (3.16)$$

The general use of Eq. 3.15 will be discussed in Section 4. The behavior for large values of time will be investigated in the next subsection.

3.3 Pseudoradial Flow

Couri (1993) investigated the wellbore pressure behavior for large values of time using his Stream Layer Model gravity drainage simulator. Semilog straight line behavior was observed at long flowtimes. There is only minor vertical flow near the well during this period, and the wellbore flux is nearly uniform. The term pseudoradial flow appears to be appropriate to describe this phenomenon.

How does Eq. 3.15 compare to the classical line source confined liquid solution (Theis, 1935 or van Everdingen and Hurst, 1949) for large values of time? The radial flow equation

for a line source well was expressed in terms of Hankel integrals, and the derivation is presented in Appendix B. This formulation is numerically inferior to the Laplace space solution commonly used, and the equation is presented only for the purpose of comparison.

$$\bar{p}_D(r_D, s) = \int_0^\infty \frac{1}{s} \frac{J_0(\lambda r_D)}{\lambda} d\lambda - \int_0^\infty \frac{1}{s + \lambda^2} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (3.17)$$

The first term in Eqs. 3.15 and 3.17 is identical, the second term differs. The pressure drop described by Eq. 3.15 is a function of elevation, and the effect of the z -coordinate is expected to decrease at long flowtimes. The late time comparison between the two solutions is displayed in Fig. 3.1. The vertical boundaries are $z_D = -1$ and $z_D = 0$.

The gravity drainage solution is independent of elevation at long flowtimes, and the corresponding semilog straight line has the same slope as the line source solution. However, the inverse of the initial formation pressure is used instead of the compressibility in the confined solution.

The dimensionless time parameter in Fig. 3.1 is based on vertical permeability. The wellbore pressure drop during pseudoradial flow is governed by horizontal permeability, and the semilog straight line may be affected by the anisotropy ratio. The isotropic case is considered in Fig. 3.1. The effects of anisotropy will be discussed in Section 4.

The pressure drop in Fig. 3.1 is calculated at the skin radius. The gravity drainage problem has a discontinuity at the interface between the wellbore and the formation. The liquid level in the wellbore is lower than the level in the adjacent reservoir rock, and this vertical difference is called the seepage height. The discontinuity of the physical system is simplified by introducing a finite skin radius that is slightly greater than the wellbore radius. The liquid height in the sand adjacent to the wellbore is located at the skin radius. This gravity drainage system is shown in Fig. 1.1.

The upper boundary in the gravity drainage solution is fixed at the initial position for all times. The pressure drops calculated from Eq. 3.15 and plotted in Fig. 3.1 assume the wellbore inflow to be from the total initial thickness, even after a significant drop in wellbore liquid height has occurred. Equation 3.15 produces a higher pressure drop than Eq. 3.17 after the semilog straight line has been reached.

In Section 4, the inner boundary condition will be modified numerically by limiting the inflow height to the actual wellbore liquid height. The gravity drainage problem is nonlinear, and it will be shown that Eq. 3.15 agrees with Eq. 3.17 at long flowtimes as long as the flowrate is low.

Ramey *et al.* (1989) applied the ideal gas correlation of Jenkins and Aronofsky (1953) to the Dupuit-Forchheimer transient gravity drainage theory. In the early study by Jenkins and Aronofsky it was found that the compressibility in the diffusivity term can be evaluated at the initial pressure, and it was shown that the confined liquid solutions of Theis (1935) and van Everdingen and Hurst (1949) will work reasonably well as long as the production rate is not too large. The condition that flowrate is not too large for the Jenkins-Aronofsky gas solution is different from the limited drawdown assumption usually considered for gravity drainage. Jenkins and Aronofsky showed their correlation of the nonlinear gas flow problem depended on flowrate. Also, the ideal gas flow problem concerns compressible flow while the Dupuit-Forchheimer problem concerns incompressible flow. The inverse of the initial

Figure 3.1: Dimensionless pressure drop at skin radius versus dimensionless vertical flow time for several dimensionless heights.

pressure takes the place of compressibility when applied to gravity drainage.

The Dupuit-Forchheimer model is expected to approximate Eq. 3.15 for low flowrates after reaching pseudoradial flow. The base case parameters used in the numerical examples were: $k_H = k_V = 240$ md, $h_i = 189$ ft, $\mu = 1$ cp, $\rho = 62.4$ lbm/cuft, $r_w = 3.18$ inch = 0.265 ft and $\phi = 0.25$. Values of qB equal to 50, 100 and 200 bbl/day were selected for the comparison shown in Figs. 3.2, 3.3 and 3.4.

The models agree for the two low flowrates, both in terms of slope as well as pressure drop. For the high rate case of 200 bbl/day, the Dupuit-Forchheimer model produces a slope that is different from the slope calculated from the line source solution. This behavior is analogous to comparing the flow of ideal gas and liquid.

Of great practical interest is the drawdown time necessary to reach the semilog straight line. In Fig. 3.2, pseudoradial flow starts somewhere between 20,000 minutes (2 weeks) and 200,000 minutes (20 weeks). The exact time of start will be determined in Section 4 using the proper value of z_D .

3.4 Skin and Wellbore Storage

The Dupuit-Forchheimer solution (Ramey *et al.*, 1989) and Eq. 3.15 agree fairly well after reaching pseudoradial flow as long as the flowrate is modest. This approximate equality may be used to obtain an expression for the skin factor that will work reasonably well in Eq. 3.15. From Ramey *et al.* (1989):

$$\frac{(p_s - p_w)(p_s + p_w)}{2} \frac{144k_H}{141.2qB\mu\rho} = S \quad (3.18)$$

The pressure at the skin radius is p_s , and in the wellbore the pressure is p_w (see Fig. 1.1). The pressure drop due to wellbore damage is modelled as a cylindrical region of reduced permeability located between these two radii. This is discussed in Section 4. From Eq. 3.1, expressed in terms of a skin factor corresponding to the analytical gravity drainage solution:

$$(p_s - p_w)p_i \frac{144k_H}{141.2qB\mu\rho} = S^* \quad (3.19)$$

The skin factor S^* is the notation used for the gravity drainage skin. The term $(p_s - p_w)/2$ is small compared to p_w , and the desired skin formulation is obtained by comparing Eqs. 3.18 and 3.19. The apparent formation damage skin, S^* , is magnified by the increased flux due to reduced inflow height. This is similar to a partially penetrating confined well, if the effects of flow convergence are neglected.

$$S^* = \frac{h_i}{h_w} S \quad (3.20)$$

The dimensionless wellbore pressure drop is Φ_{wD} . The liquid in the wellbore is static, and the potential is independent of the z -coordinate. The expression for Φ_{wD} may be obtained from Eq. 3.1:

Figure 3.2: Pressure drop at skin radius versus flow time for $q_B = 50$ bbl/day. Isotropic base case.

Figure 3.3: Pressure drop at skin radius versus flow time for $q_B = 100$ bbl/day. Isotropic base case.

Figure 3.4: Pressure drop at skin radius versus flow time for $qB = 200$ bbl/day. Isotropic base case.

$$\Phi_{wD}(t_{DV}) = \frac{k_H h_i}{141.2qB\mu} [p_i - p_w(t)] \quad (3.21)$$

The liquid level in the wellbore is initially equal to that in the reservoir. As flow is initiated, the first liquid produced at the surface will come from the wellbore. The initial sandface flowrate is zero, and the reservoir production approaches the surface production as flowtime increases. Lee (1982) presented a derivation leading to the wellbore storage coefficient for the moving liquid level case. The quantity of liquid stored in the wellbore for a gravity drainage well is a function of wellbore radius and liquid density. The duration of wellbore storage, however, is affected by the deliverability of the well in addition to the wellbore storage coefficient.

$$C = (144/5.615)A_{wb}/\rho \quad (3.22)$$

The wellbore storage coefficient was defined in the classical study by van Everdingen and Hurst (1949). The rate of wellbore unloading, q_{wb} , is directly proportional to the rate of change of wellbore pressure. At early times, the wellbore unloading rate is large, causing a rapid change of bottomhole pressure.

$$q_{wb}(t) = 24Cd(p_i - p_w)/dt \quad (3.23)$$

The sandface production rate, q_{sf} , is equal to the surface flowrate minus the rate of wellbore unloading. This mass balance is combined with Eqs. 3.3, 3.21 and 3.23 to give:

$$\frac{q_{sf}(t)}{q} = 1 - \frac{k_V}{k_H} \frac{5.615C}{2\pi h_i \phi p_i^{-1} h_i^2} \frac{d\Phi_{wD}}{dt_{DV}} \quad (3.24)$$

The group of parameters in front of $d\Phi_{wD}/dt_{DV}$ in Eq. 3.24 may be combined into a single dimensionless parameter. This is the dimensionless wellbore storage coefficient for the analytical gravity drainage solution:

$$C_D = \frac{k_V}{k_H} \frac{5.615C}{2\pi h_i \phi p_i^{-1} h_i^2} \quad (3.25)$$

The superposition theorem was used by van Everdingen and Hurst (1949) to obtain a solution for a line source well with wellbore storage. The convolution integral was transformed to Laplace space and a solution for wellbore pressure drop was obtained. The same technique may be applied to the linearized gravity drainage solution in this study. The constant-rate potential with zero skin is $\bar{\Phi}_D$, and the solution is Eq. 3.15. The wellbore potential is $\bar{\Phi}_{wD}$, and this response includes the effects of skin and wellbore storage:

$$\bar{\Phi}_{wD} = \frac{s\bar{\Phi}_D + S^*}{s[1 + C_D s(s\bar{\Phi}_D + S^*)]} \quad (3.26)$$

The expression for the skin factor S^* is a function of the solution because it contains the wellbore liquid height. It is necessary to apply an iteration process to obtain a solution of Eq. 3.26. At the initial step, a value of h_w is assumed and $\bar{\Phi}_{wD}$ is calculated. This calculated value of the wellbore potential is used to obtain a new estimate of S^* . Convergence is usually obtained within three or four iterations using h_i as the initial guess.

3.5 Summary

The Laplace space uniform flux formulation for free-surface flow was derived. This solution was based on the linearized equations by Dagan (1964). The pseudoradial flow behavior was compared with the line source confined liquid solution (Theis, 1935 or van Everdingen and Hurst, 1949) and the Dupuit-Forchheimer model (Ramey *et al.*, 1989). The pseudoradial semilog slope produced by the gravity drainage equation appears to be the same as that for confined liquid flow. Expressions for skin factor and wellbore storage were derived, and these expressions will be employed in Section 4 together with the more general uniform wellbore potential condition.

4. Uniform Wellbore Potential

The uniform wellbore potential condition is used in this section to yield a wellbore flux distribution that is more realistic than that considered in Section 3. The analytical model will be compared with results obtained using the Stream Layer Model (Couri, 1993) numerical gravity drainage simulator as well as the confined liquid solution by van Everdingen and Hurst (1949).

4.1 Green's Function Solution Procedure

Dagan (1966) derived the Green's function for free-surface flow of a homogeneous and incompressible liquid. The porous medium was considered homogeneous, isotropic and nondeformable. The solution was later extended to anisotropic reservoirs (Dagan, 1967).

$$G(z_D, r_D, z_{D_o}, t_{DV}) = \frac{1}{4\pi} \left[\frac{1}{\sqrt{r_D^2 + (z_D - z_{D_o})^2}} + \frac{1}{\sqrt{r_D^2 + (z_D + z_{D_o})^2}} \right. \\ \left. - 2 \int_0^\infty \frac{\cosh \lambda(z_{D_o} + 1) \cosh \lambda(z_D + 1) e^{-\lambda t_{DV} \tanh \lambda}}{\sinh \lambda \cosh \lambda} J_0(\lambda r_D) d\lambda \right. \\ \left. + 2 \int_0^\infty \frac{\cosh \lambda z_{D_o} \cosh \lambda z_D e^{-\lambda}}{\sinh \lambda} J_0(\lambda r_D) d\lambda \right] \quad (4.1)$$

The real time solution corresponding to Eq. 3.15 may be obtained by integrating Eq. 4.1 with respect to z_{D_o} from the bottom of the reservoir to the top. The dimensionless wellbore flowrate used by Dagan as the inner boundary condition was equal to $-1/(2\pi)$, whereas in this study the flowrate was -1 (see Eq. 3.9). This explains why the factor 2π appears in Eq. 4.2.

$$\Phi_D(z_D, r_D, t_{DV}) = 2\pi \int_{-1}^0 G(z_D, r_D, z_{D_o}, t_{DV}) dz_{D_o} \quad (4.2)$$

The presence of a free liquid surface is characteristic of low pressure gravity-flow systems. This is different from high-pressure reservoirs where the liquid must fill the entire porous medium. The free surface will terminate above the wellbore liquid level as indicated in Fig. 1.1. The vertical interval between the top of the wellbore liquid level and the free surface is termed the seepage height. The liquid is exposed to atmospheric pressure at the seepage face, but this surface is not a streamline. Capillary forces are important at the seepage face. Below this interval the liquid enters the wellbore as regular flow with the streamlines perpendicular to the well.

Equation 4.1 was derived assuming small drawdown of the liquid table. An investigation using the Stream Layer Model (Couri, 1993) indicated the seepage flowrate to be negligible compared to the total flowrate for the range of parameters reported in this study.

In the following formulations the seepage yield is neglected. The wellbore liquid height is divided into M segments:

$$\Phi_D(z_D, r_{sD}, t_{DV}) = 2\pi \frac{h_i}{h_w} \int_{-1}^{h_w/h_i-1} q_D(z_{Do}, t_{DV}) G(z_D, r_{sD}, z_{Do}, t_{DV}) dz_{Do} \quad (4.3)$$

$$\doteq 2\pi \frac{h_i}{h_w} \sum_{m=1}^M q_{Dm}(t_{DV}) \int_{x_{m-1}}^{x_m} G(z_D, r_{sD}, z_{Do}, t_{DV}) dz_{Do} \quad (4.4)$$

The dimensionless flowrate $q_{Dm} = q_m/q$ is introduced in the middle of each segment. The uniform flux wellbore condition gives q_m equal to unity regardless of elevation. The uniform potential wellbore condition is expected to yield q_m greater than unity towards the top of the interval and q_m smaller than unity towards the bottom. For $m = 1, M$:

$$z_{Dm} = [(2m - 1)/(2M)]h_w/h_i - 1 \quad (4.5)$$

The Green's function in Eq. 4.4 is integrated from the bottom of each segment to the top. The integration limits are x_{m-1} and x_m . The value of the integral is multiplied by the dimensionless flowrate for that segment. The contributions from all the wellbore segments are summed to give the potential at the skin radius for the value of z_D of interest. For $j = 0, M$:

$$x_j = (j/M)h_w/h_i - 1 \quad (4.6)$$

Muskat (1932) used a similar technique to obtain uniform wellbore potential for a partially penetrating well. The objective is to determine the strengths of the q_m s that give the same potential regardless of elevation. This process requires the solution of $M + 1$ linear equations. M equations are obtained by setting Φ_{wD} equal at the midpoint of each wellbore segment. The last equation is $\sum_{m=1}^M q_{Dm} = M$. The $M + 1$ unknowns are the M dimensionless flowrates and the wellbore potential. Several values for M were tried, and ten segments were used in calculations. The evaluation of Eq. 4.4 is time consuming, and ten segments give a reasonable compromise between speed and accuracy.

The potentials at the skin and wellbore radii are related through Darcys Law. The skin zone is a vertical cylinder with inner radius of r_w and outer radius of r_s (see Fig. 1.1). For radial flow across the skin zone and into the well:

$$\Phi_{wD}(t_{DV}) - \Phi_D(z_D, r_{sD}, t_{DV}) = \frac{h_i(S + \ln r_s/r_w)}{h_w(t_{DV})} q_D(z_D, t_{DV}) \quad (4.7)$$

The Hawkins (1956) form of a finite thickness skin factor was used to obtain Eq. 4.7. The horizontal permeability for the skin zone is k_H^* . If there is no damage near the wellbore, k_H^* is equal to k_H . For a finite thickness skin:

$$S = (k_H/k_H^* - 1) \ln r_s/r_w \quad (4.8)$$

Equation 4.4 may be integrated between $z_D = -1$ and $z_D = 0$ to produce an average dimensionless potential. The difference between the wellbore potential and the average potential at the skin radius represents the seepage height. From Eq. 4.7, the seepage height has a direct relationship to the skin factor.

The effects of wellbore storage and skin will be discussed later. First, the behavior of Eq. 4.4 for intermediate and long flowtimes will be analyzed for the isotropic case. The wellbore liquid height h_w is the desired solution. A trial and error process is necessary because the wellbore liquid level appears in the solution.

The dimensionless wellbore pressure drop was calculated for $qB = 50, 100$ and 200 bbl/day. Figure 4.1 shows a comparison with the confined liquid solution of van Everdingen and Hurst (1949). As the flowrate approaches zero, Eq. 4.4 appears to coincide with Eq. 3.17 after reaching pseudoradial flow. For high flowrates, the gravity drainage solution produces a greater pressure drop than the confined solution. The slope is the same regardless of flowrate for the isotropic case.

The anisotropy effects will be analyzed in Subsection 4.3. The dimensionless horizontal flow time is:

$$t_{DH} = t_{DV} \frac{k_H}{k_V} \quad (4.9)$$

From Eqs. 3.3 and 4.9:

$$t_{DH} = \frac{263.7 \cdot 10^{-6} k_H t}{\phi \mu p_i^{-1} h_i^2} \quad (4.10)$$

4.2 Flux Distribution

Wellbore flux distributions calculated from Eq. 4.4 were compared with results from the Stream Layer Model (Couri, 1993). A total of thirty segments were used in the analytical model to produce a smooth curve. The wellbore flux profiles for $qB = 50, 100$ and 200 bbl/day after 200 minutes of production are presented in Figs. 4.2 through 4.4. The solid horizontal straight line indicates the top of the wellbore liquid column calculated using the analytical model. The base case parameters defined in Subsection 3.3 were used, and the system was isotropic with $k_H = k_V = 240$ md. The wellbore flux distribution calculated using the analytical model was virtually independent of flowrate. The wellbore liquid height was smaller the higher the flowrate, and a slight shift to high flux values was observed for the analytical model as the flowrate increased. The only major effect of increasing flowrate was a decreased inflow at the top of the liquid column.

Figure 4.1: Dimensionless wellbore pressure drop versus dimensionless vertical flow time for several flowrates.

Figure 4.2: Dimensionless wellbore height versus dimensionless flowrate at $qB = 50$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes.

Figure 4.3: Dimensionless wellbore height versus dimensionless flowrate at $q_B = 100$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes.

Figure 4.4: Dimensionless wellbore height versus dimensionless flowrate at $q_B = 200$ bbl/day for analytical model and finite difference model. Flowtime is 200 minutes.

The flux distribution calculated using the stream layer numerical simulator appeared reasonable. The flow from the seepage face appeared clearly for $qB = 100$ and 200 bbl/day. For the low rate of 50 bbl/day, there was negligible seepage yield.

The flux at the wellbore is a function of flowrate as indicated by numerical results. Increased flowrate causes a more uniform flux distribution as well as a higher seepage yield. The inflow profile calculated using the analytical model agrees well with that of the Stream Layer Model for the case of $qB = 200$ bbl/day, except towards the top of the wellbore liquid column. The analytical model neglects the presence of seepage yield, and Eq. 4.4 produces a large dimensionless flowrate at the top of the liquid column to compensate for the small available inflow interval. For the low flowrates of 50 and 100 bbl/day, the Stream Layer Model produces lower inflow at the well bottom than does the analytical model. At the top of the wellbore liquid column, the behavior is reversed.

4.3 Pressure Drawdown Behavior

In Eq. 4.4, the potential at the skin radius is expressed in the real time domain, and wellbore storage effects cannot be incorporated readily. The uniform flux solution, Eq. 3.15, is expressed in Laplace space, and this equation may be used to calculate the wellbore pressure drop if the proper value of z_D is used.

The uniform potential solution, Eq. 4.4, was used to obtain the wellbore liquid height for constant sandface flowrate. This result was compared with the uniform flux solution, Eq. 3.15, to determine the location of the equivalent average potential point. The wellbore pressure drop with wellbore storage and skin was obtained from Eq. 3.26 using the average value of z_D . A similar procedure was used by Gringarten and Ramey (1975) for a partially penetrating well.

Figures 4.5 and 4.6 present the gravity drainage solution for $qB = 50, 100$ and 200 bbl/day and the confined liquid solution by van Everdingen and Hurst (1949). Dimensionless wellbore pressure drop is graphed versus dimensionless vertical flow time for the isotropic base case in Subsection 3.3. In the definitions of dimensionless time and wellbore storage coefficient used by van Everdingen and Hurst (1949), the total system compressibility is replaced by the inverse of initial pressure. The resulting C_D for the confined flow model is equal to 2.0 , which for this model is equivalent to zero wellbore storage.

The log-log graph is presented in Fig. 4.5, and the unit slope indicates the presence of wellbore storage. The definition of dimensionless wellbore storage coefficient for the gravity drainage solution is different from that of the equivalent confined liquid flow. For the isotropic case, the gravity drainage C_D is equal to the confined flow C_D multiplied by $(r_w/h_i)^2$. This ratio is also the difference between the two definitions of dimensionless time. In spite of the smaller dimensionless wellbore storage coefficient for the gravity drainage solution, the different definition of dimensionless time causes the unit slope to last longer.

The wellbore storage effect associated with gravity drainage wells is caused by the moving wellbore liquid level, and the magnitude of C is much greater than the wellbore storage coefficient due to compressibility effects. The latter case is realized in high-pressure reservoirs completed with a packer.

Three distinct flow periods may be identified from the semilog graph on Fig. 4.6. After the end of wellbore storage effects, there is a long period of nearly constant wellbore pressure. The flat middle period is a distinct feature of a gravity drainage drawdown test.

Figure 4.5: Dimensionless wellbore pressure drop versus dimensionless vertical flow time for gravity drainage and compressible flow. Log-log graph for isotropic base case.

Figure 4.6: Dimensionless wellbore pressure drop versus dimensionless vertical flow time for gravity drainage and compressible flow. Semilog graph for isotropic base case.

At long flowtimes, the semilog straight line appears, and this is similar to confined liquid flow.

The gravity drainage solution is nonlinear because the dimensionless wellbore pressure drop increases as flowrate increases. The analytical solution is linear, and a nonlinearity was introduced at the wellbore by adjusting the inflow interval according to the wellbore liquid height. The solution by van Everdingen and Hurst (1949) may be used at low flowrates during pseudoradial flow.

The anisotropy effects will be analyzed next. The pressure drop during pseudoradial flow is controlled mainly by horizontal permeability. However, in Figs. 4.5 and 4.6 the dimensionless wellbore pressure drop was graphed versus a dimensionless time based on vertical permeability.

The calculated values for $k_V/k_H = 0.2, 1.0$ and 4.0 are shown in Fig. 4.7. The dimensionless wellbore pressure drop was graphed versus the dimensionless horizontal flow time defined as Eq. 4.9. The flowrate was 50 bbl/day and the horizontal permeability was 240 md. The slope appears to be the same regardless of the anisotropy ratio. Horizontal permeability may be calculated from the semilog straight line, and vertical permeability is obtainable from the desaturation data. Similar type curves are shown in Figs. 4.8 and 4.9 for flowrates of 100 bbl/day and 200 bbl/day. The use of t_{DH} as a correlating parameter for anisotropy effects works fairly well even for the highest flowrate of 200 bbl/day.

In the remainder of this subsection, the wellbore pressure drop calculated using the analytical model will be compared with results from the Stream Layer Model (Couri, 1993) for different flowrates, permeabilities, skin factors, initial liquid heights, and formation anisotropy ratios. The pressure drop calculated using the van Everdingen and Hurst (1949) line source solution is included for comparison at long flowtimes. The base case parameters defined in Subsection 3.3 were used for calculations.

In Fig. 4.10, the wellbore pressure drop for $qB = 50, 100$ and 200 bbl/day is graphed versus the logarithm of drawdown time. There is good agreement between the analytical model and the numerical simulator for the two lowest flowrates. The presence of an intermediate flow period of nearly constant wellbore pressure is verified by the Stream Layer Model. The early time wellbore storage and late time pseudoradial flow periods agree reasonably also. For the highest flowrate, the numerical simulator produces a larger pressure drop than the analytical solution. The behavior during the intermediate time portion is interesting. The wellbore pressure appears to increase with time, although one would expect a decrease in flowing pressure. Perhaps a reduction of the seepage face as the upper portion of the porous medium continues to drain is responsible. On the other hand, numerical phenomenon may be responsible.

Figure 4.11 presents the effect of permeability. Isotropic cases of $k = 120, 240$ and 480 md were considered. The three sets of curves represent $k/qB = 1.2, 2.4$ and 4.8 md/(bbl/day), and these ratios are the same as plotted in Fig. 4.10. Except for a horizontal time shift, the pressure profiles in Figs. 4.10 and 4.11 are identical. The discussion regarding flowrate variations given in the previous paragraph applies to the analysis of permeability differences.

The pressure drop for skin factor $S = 0, 4$ and 8 is presented in Fig. 4.12. The analytical gravity drainage model is in fair agreement with numerical results, although the pressure drop appears a little high for $S = 8$. For confined liquid flow, the dimensionless wellbore pressure drop at long time is obtained by adding the skin factor to the dimensionless

Figure 4.7: Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 50 \text{ bbl/day}$.

Figure 4.8: Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 100$ bbl/day.

Figure 4.9: Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for several anisotropy ratios. Flowrate $qB = 200 \text{ bbl/day}$.

Figure 4.10: Wellbore pressure drop versus flow time for several flowrates. Base isotropic case with $k = 240$ md.

Figure 4.11: Wellbore pressure drop versus flow time for several permeabilities. Base isotropic case with $qB = 100$ bbl/day.

Figure 4.12: Wellbore pressure drop versus flow time for several skin factors. Base isotropic case with $qB = 100$ bbl/day and $k = 240$ md.

pressure drop for zero skin. For gravity drainage flow, however, the skin factor should be multiplied by a factor greater than unity before the skin is added to the dimensionless solution for zero wellbore damage. This observation is in agreement with Eq. 3.20 presented in Section 3. The pressure values calculated using the analytical gravity drainage solution are based on Eq. 3.20 at early and intermediate flowtimes, while the pseudoradial pressure points are based on Eq. 4.7 only.

In Fig. 4.13 the effect of changing the initial reservoir liquid height is displayed. The values of $h_i = 94.5, 189$ and 378 ft were considered. The three sets of curves in this figure have the same kh_i value as those in Fig. 4.11, but there is an appreciable difference between changing permeability and changing initial liquid height. During the flat desaturation period, decreased permeability causes a larger pressure drop than a corresponding decrease in initial liquid height. If the permeability and initial liquid height are increased, the reverse is true. These observations are correct for both the analytical solution and the Stream Layer Model. The nonlinearity of the gravity drainage problem is apparent for the upper curve. The confined liquid solution does not hold at long times for the low value of h_i .

Figure 4.14 shows the effects of changing vertical permeability. Three permeability ratios $k_V/k_H = 0.20, 1.0$ and 4.0 were used for the calculations. The seepage face is more developed in the case of high vertical flow resistance, and modelling the gravity drainage problem becomes difficult for low values of k_V/k_H . Effective vertical permeability is usually lower than horizontal permeability for reservoirs made up of horizontal beds. The analytical gravity drainage values differ from those of the numerical Stream Layer Model for $k_V = 48$ md at intermediate flowtimes. During this period, the liquid is drained vertically, and the degree of anisotropy is an important parameter. There is also a noticeable difference at the end of the wellbore storage period. The discrepancy might indicate limitations of the analytical solution. The reason may also be numerical. During the pseudoradial flow period, the wellbore pressure drop is governed mainly by horizontal permeability. If vertical flow effects are negligible, all curves should coincide. There is only one curve for the confined flow model. This solution was derived assuming horizontal flow, and vertical permeability does not appear in the confined flow model solution.

4.4 Summary

The uniform wellbore potential condition was employed in this section. The wellbore flux distribution calculated using the analytical solution was compared with results of the Stream Layer Model (Couri, 1993) numerical gravity drainage simulator, and the analytical model is in fair agreement. The calculated wellbore pressures were compared, and three separate transient flow periods were observed: the wellbore storage period, a flat intermediate period, and finally, the pseudoradial period. The flat intermediate period is a distinct feature of a gravity drainage drawdown test. The confined liquid solutions by Theis (1935) and van Everdingen and Hurst (1949) were found to coincide with the gravity drainage solution during pseudoradial flow for low flowrates.

Figure 4.13: Wellbore pressure drop versus flow time for several initial liquid heights. Base isotropic case with $qB = 100$ bbl/day and $k = 240$ md.

Figure 4.14: Wellbore pressure drop versus flow time for several anisotropy ratios. Base case with $qB = 100$ bbl/day and $k_H = 240$ md.

5. Discussion and Conclusions

The design and execution of a well test depends upon the test objectives. Is the goal to determine reservoir conductivity, wellbore damage or location of drainage boundaries? Or is the purpose to monitor reservoir performance and near-wellbore pressure losses over time? Generally speaking, the two key parameters in well test design are flowrate and test duration. The flowrate should be high enough to cause a measurable pressure drop, and it is necessary to gather enough pressure-time data to ensure a reliable analysis.

Practical values for reservoir and fluid parameters corresponding to a water well were used in the numerical examples analyzed in this study, and the maximum pressure drop calculated was about 5 psi. The initial reservoir pressure was considered equal to the hydrostatic liquid head. Both the relative and absolute pressures associated with gravity drainage systems are lower than for conventional oil reservoirs, and the pressure tools should be selected accordingly.

Gravity drainage is a slow process, and the liquid is brought to the surface using an artificial lifting device. It is preferable to produce at a low rate. The numerical examples show clearly the nonlinearity of the gravity drainage problem, and the drawdown behavior becomes more complex for high flowrates. The analytical gravity drainage solution presented here is limited to small drawdowns.

Three distinct flow periods were observed from the plots made using the analytical gravity drainage solution, and this behavior was verified using a numerical gravity drainage simulator. At early times, the reservoir response was masked by the moving liquid level in the wellbore. During this time, the log-log graph exhibited the unit slope characteristic of wellbore storage. At intermediate times, there is a long period of nearly constant wellbore pressure. During this period the free surface moves downward, and the liquid is produced mainly by vertical drainage. This is often called the delayed yield period. At long flowtimes, the flow is nearly horizontal, and the semilog plot revealed a straight line with the same slope as the line source solution for compressible flow.

The duration of wellbore storage is a function of reservoir transmissivity, anisotropy ratio, skin factor, and wellbore storage coefficient. Of these parameters, usually only the storage coefficient is known beforehand. The storage effect is independent of flowrate. The numerical examples indicated the end of wellbore storage after a short production time.

The flat desaturation period lasts for a long time. Depending upon the reservoir parameters, the onset of pseudoradial flow may require several months. The wellbore pressure level during this period is sensitive to vertical permeability. The numerical results indicated increasing wellbore pressure with time during this period, especially in cases with a well developed seepage face. The analytical solution did not produce such behavior.

Once the semilog straight line is reached, reliable estimates for horizontal transmissivity and wellbore damage may be obtained from the slope. At low flowrates, the confined liquid solutions of Theis (1935) and van Everdingen and Hurst (1949) appear to coincide with the analytical gravity drainage solution. For high flowrates, the pseudoradial slope is the same, but there is a difference in pressure. The gravity drainage solution produces high wellbore pressure because of the nonlinear inner boundary condition.

The dimensionless time parameter in the gravity drainage solution is expressed in

terms of vertical permeability, and horizontal permeability does not appear in the analytical model. The pseudoradial flow behavior is controlled by horizontal permeability. The gravity drainage solution produces the same slope as the line source solution for compressible flow. In Section 3, the gravity drainage solution was calculated for inflow into the initial wellbore liquid height without considering the drop of the wellbore liquid level with time. The gravity drainage solution was independent of flowrate, and the pseudoradial pressure drop was larger than calculated from the line source solution. In Section 4, the inflow interval was adjusted according to the wellbore liquid height, and the solution became dependent upon flowrate. For low flowrates, the pseudoradial pressure drop was approximately the same as the line source solution for compressible flow.

The gravity drainage solution was analyzed for different anisotropy ratios by introducing a dimensionless time parameter based on horizontal permeability. Dimensionless wellbore pressure drop was graphed versus dimensionless horizontal flow time, and the calculated pressure values for different anisotropy ratios merged after reaching pseudoradial flow. The dimensionless wellbore pressure is a function of flowrate, and type curves for $qB = 50, 100$ and 200 bbl/day were presented. These type curves yield both vertical and horizontal permeabilities.

The well test data obtained during the desaturation period contains information about vertical permeability, and the pseudoradial flow behavior is governed by horizontal permeability. To ensure a confident estimate for horizontal permeability, a complete log cycle of pseudoradial pressure-time data may be required if conventional semilog straight line techniques are used. The duration of the drawdown test may exceed practical limits. A solution to this problem could be the use of nonlinear regression. Instead of evaluating each flow period separately, the entire data set is analyzed together.

The conclusions from this study are:

An analytical model was developed for pressure drawdown testing of gravity drainage wells. The solution is not valid for pressure buildup analysis. The governing equations were developed by Dagan (1964) using a perturbation technique, and the model is limited to small drawdowns. The free-surface flow is viewed as incompressible, and anisotropy effects are included in the solution. The well is a line source well, and the reservoir is infinitely large.

The uniform flux solution was derived in Laplace space, and the results were compared with the confined liquid solution (Theis, 1935 or van Everdingen and Hurst, 1949). The gravity drainage model and the confined flow solutions yielded the same slope after reaching the semilog straight line, but the pressure was different. The gravity drainage solution was modified at the wellbore by adjusting the inflow interval according to the wellbore liquid height. After this modification, the two models agreed both in terms of pressure and slope as long as the flowrate was low. For high flowrates, the gravity drainage model produces an increased pressure drop, but the slope is the same. The nonlinearity introduced at the wellbore is in agreement with results obtained using a numerical gravity drainage simulator.

The uniform wellbore potential inner boundary condition was modelled using the proper Green's function. The discontinuity at the wellbore was solved by introducing a finite skin radius, and the formulation produces a seepage face. The gravity drainage model ignores the inflow above the free surface in the well since the seepage yield is small compared to the total flowrate.

The flux distribution at the wellbore was in fair agreement with results obtained

using numerical simulation. The calculated wellbore pressures agreed also. Three distinct flow periods were observed. The wellbore storage period was caused by the moving liquid level, and the duration was short. During the long intermediate flow period, the wellbore pressure was nearly constant. In this period, the free surface moved downwards, and the liquid was produced mainly by vertical drainage. At long times, the semilog straight line for pseudoradial flow appeared.

Equations for the gravity drainage skin factor and dimensionless wellbore storage coefficient were developed. The effects of these parameters were incorporated in the convolution integral using the Laplace space uniform flux solution. The proper value of the vertical space coordinate was determined using the Green's function.

New type curves were presented. Dimensionless wellbore pressure drop was graphed versus dimensionless horizontal flow time, and these type curves yield both vertical and horizontal permeabilities.

Nomenclature

A_{wb} = cross sectional wellbore area, sqft
 B = liquid volume factor, bbl/STB
 C = wellbore storage coefficient, STB/psi
 C_D = dimensionless wellbore storage coefficient
 C_t = total system compressibility, 1/psi
 D = delayed yield parameter, ft
 h_i = initial reservoir liquid height, ft
 h_w = wellbore liquid height, ft
 k = isotropic permeability, md
 k_H = horizontal permeability, md
 k_V = vertical permeability, md
 p = pressure, psi
 p_D = dimensionless pressure for confined flow
 $\bar{p}_D$ = Laplace transformed dimensionless pressure for confined flow
 $\hat{p}_D$ = Hankel transformed dimensionless pressure for confined flow
 p_i = initial pressure, psi
 p_s = pressure at skin radius, psi
 p_{sD} = dimensionless pressure at skin radius for confined flow
 p_w = wellbore pressure, psi
 p_{wD} = dimensionless wellbore pressure for confined flow
 q = production rate, STB/day
 q_D = dimensionless production rate
 q_{sf} = sandface flowrate, STB/day
 q_{wb} = wellbore flowrate, STB/day
 r = radial distance, ft
 r_D = dimensionless radial distance
 r_{D1} = dimensionless radial distance for confined flow
 r_s = skin radius, ft
 r_{sD} = dimensionless skin radius
 r_w = wellbore radius, ft
 r_{wD} = dimensionless wellbore radius
 S = skin factor
 S^* = gravity drainage skin factor
 s = Laplace parameter
 t = flow time, hr
 t_D = dimensionless isotropic flow time
 t_{D1} = dimensionless flow time for confined flow

t_{DH} = dimensionless horizontal flow time

t_{DV} = dimensionless vertical flow time

z = vertical distance, ft

z_D = dimensionless vertical distance

α^{-1} = delay index, hr

λ = Hankel parameter

μ = viscosity, cp

ρ = density, lbm/cuft

ϕ = porosity

Φ_D = dimensionless potential

$\bar{\Phi}_D$ = Laplace transformed dimensionless potential

$\tilde{\Phi}_D$ = Hankel transformed Laplace transformed dimensionless potential

Φ_{sD} = dimensionless potential at skin radius

Φ_{wD} = dimensionless wellbore pressure drop

$\bar{\Phi}_{wD}$ = Laplace transformed dimensionless wellbore pressure drop

$\mathcal{H}$ = Hankel operator

Bibliography

- [1] Boulton, N.S.: "The Drawdown of the Water-Table under Non-Steady Conditions Near a Pumped Well in an Unconfined Formation," *Proceeding, Institution of Civil Engineers* (1954a) **3**, Part 3, 564-579.
- [2] Boulton, N.S.: "Unsteady Radial Flow to a Pumped Well Allowing for Delayed Yield from Storage," *International Association of Scientific Hydrology, Rome* (1954b) Tome 2, 472-477.
- [3] Boulton, N.S.: "Analysis of Data from Non-Equilibrium Pumping Tests Allowing for Delayed Yield from Storage," *Proceeding, Institution of Civil Engineers* (1963) **26**, 469-482.
- [4] Couri, F.R.: *A Finite Difference Model for Free Surface Gravity Drainage*, Ph.D. Dissertation, Stanford University (1993).
- [5] Dagan, G.: "Second Order Linearized Theory of Free Surface Flow in Porous Media," *La Houille Blanche*, Grenoble, France (1964) **8**, 901-910.
- [6] Dagan, G.: "The Solution of the Linearized Equations of Free-Surface Flow in Porous Media," *Journal de Mécanique* (1966) **5**(2), 207-215.
- [7] Dagan, G.: "A Method of Determining the Permeability and Effective Porosity of Unconfined Anisotropic Aquifers," *Water Resources Research* (1967) **3**(4), 1059-1071.
- [8] Dupuit, J.: *Études Theoriques et Pratiques sur le Mouvement des Eaux*, Paris (1863).
- [9] Forchheimer, Ph.: *Zeits. Arch. Ing. Ver.*, Hannover (1886).
- [10] Gringarten, A.C. and Ramey, H.J., Jr.: "An Approximate Infinite Conductivity Solution for a Partially Penetrating Line-Source Well," *Society of Petroleum Engineers Journal* (April 1975) 140-148.
- [11] Hawkins, M.F., Jr.: "A Note on the Skin Effect," *Transactions, AIME* (1956) **207**, 356-357.
- [12] Jenkins, R. and Aronofsky, J.S.: "Unsteady Radial Flow of Gas Through Porous Media," *Journal of Applied Mechanics* (June 1953) 210-214.

- [13] Kroszynski, U.I. and Dagan, G.: "Well Pumping in Unconfined Aquifers: The Influence of the Unsaturated Zone," *Water Resources Research* (1975) **11**(3), 479-490.
- [14] Lee, J.: *Well Testing*, Society of Petroleum Engineers of AIME, Dallas (1982).
- [15] Muskat, M.: "Potential Distributions in Large Cylindrical Disks with Partially Penetrating Electrodes," *Physics* (1932) **2**, 329-364.
- [16] Muskat, M.: *The Flow of Homogeneous Fluids Through Porous Media*, Chapter 6, McGraw-Hill Book Co., New York (1937).
- [17] Neuman, S.P.: *Transient Flow of Ground Water to Wells in Multiple-Aquifer Systems*, Ph.D. Dissertation, University of California, Berkeley (December 1968).
- [18] Neuman, S.P.: "Theory of Flow in Unconfined Aquifers Considering Delayed Response of the Water Table," *Water Resources Research* (1972) **8**(4), 1031-1045.
- [19] Prickett, T.A.: "Type-Curve Solution to Aquifer Tests under Water-Table Conditions," *Ground Water* (1965) **3**(3), 5-14.
- [20] Ramey, H.J., Jr., Lichtenberger, G. and Davitt, H.J.: "Well Test Analysis for Gravity Drainage Systems," paper NMT 890017 presented at the Centennial Symposium at New Mexico Tech, Socorro, NM, October 16-19, 1989.
- [21] Streltsova, T.D.: "Unsteady Radial Flow in an Unconfined Aquifer," *Water Resources Research* (1972) **8**(4), 1059-1066.
- [22] Theis, C.V.: "The Relation between the Lowering of the Piezometric Surface and the Rate and Duration of Discharge of a Well Using Ground-Water Storage," *Transactions*, Am. Geophys. Union, 16th Annual Meeting, Vol. 16 (1935).
- [23] van Everdingen, A.F. and Hurst, W.: "The Application of the Laplace Transformation to Flow Problems in Reservoirs," *Transactions*, AIME (1949) **186**, 305-324.
- [24] Wyckoff, R.D., Botset, H.G. and Muskat, M.: "Flow of Liquids through Porous Media under the Action of Gravity," *Physics* (1932) **3**, 90-113.

Appendix A. Solution of Differential Equation

From Eq. 3.12, substituting $\tilde{\Phi}_D = y$, $z_D = x$ and $\lambda = k$:

$$d^2y/dx^2 - k^2y = -1/s \quad (\text{A.1})$$

The desired solution is the summation of the homogeneous and particular solutions:

$$y = y_h + y_p \quad (\text{A.2})$$

The homogeneous solution is considered first. Substitution of $y = e^{ux}$ and its derivatives yields the characteristic equation:

$$u = \pm k \quad (\text{A.3})$$

The homogeneous solution is:

$$y_h = c_1 e^{kx} + c_2 e^{-kx} \quad (\text{A.4})$$

The next step is to determine the particular solution. Try:

$$y_p = (1/s - e^{vx})/k^2 \quad (\text{A.5})$$

From Eq. A.5:

$$d^2y_p/dx^2 = -v^2 e^{vx}/k^2 \quad (\text{A.6})$$

Combination of Eqs. A.1, A.5 and A.6 yields:

$$v = \pm k \quad (\text{A.7})$$

A particular solution is given from Eqs. A.5 and A.7:

$$y_p = (1/s - e^{kx})/k^2 \quad (\text{A.8})$$

From Eqs. A.2, A.4 and A.8:

$$y(x) = c_1 e^{kx} + c_2 e^{-kx} + (1/s - e^{kx})/k^2 \quad (\text{A.9})$$

The final step is to determine the constants c_1 and c_2 . The two boundary conditions follow from Eqs. 3.13 and 3.14:

$$dy(0)/dx + sy(0) = 0 \quad (\text{A.10})$$

$$dy(-1)/dx = 0 \quad (\text{A.11})$$

From Eqs. A.9 and A.10:

$$c_1 = -c_2(s - k)/(s + k) - (1 - s - k)/(sk^2 + k^3) \quad (\text{A.12})$$

From Eqs. A.9 and A.11:

$$c_1 = c_2 e^{2k} + 1/k^2 \quad (\text{A.13})$$

From Eqs. A.12 and A.13:

$$c_2 = -\frac{1}{k^2} \frac{1}{e^{2k}(s + k) + s - k} \quad (\text{A.14})$$

From Eqs. A.13 and A.14:

$$c_1 = \frac{1}{k^2} - \frac{e^{2k}}{k^2} \frac{1}{e^{2k}(s + k) + s - k} \quad (\text{A.15})$$

The final expression is obtained from Eqs. A.9, A.14 and A.15:

$$y(x) = \frac{1}{k^2 s} - \frac{e^{k(2+x)} + e^{-kx}}{k^2 [e^{2k}(s + k) + s - k]} \quad (\text{A.16})$$

Appendix B. Hankel Solution for a Line Source Well

The diffusivity equation for compressible fluid flow in a porous medium was solved by van Everdingen and Hurst (1949) using the Laplace transformation. The same equation will be solved here for a line source well using the Hankel transformation.

$$\frac{\partial^2 p_D}{\partial r_{D1}^2} + \frac{1}{r_{D1}} \frac{\partial p_D}{\partial r_{D1}} = \frac{\partial p_D}{\partial t_{D1}} \quad (\text{B.1})$$

The initial boundary condition is:

$$p_D(r_{D1}, 0) = 0 \quad (\text{B.2})$$

The inner and outer boundary conditions are:

$$\lim_{r_{D1} \rightarrow 0} r_{D1} \frac{\partial p_D(r_{D1}, t_{D1})}{\partial r_{D1}} = -1 \quad (\text{B.3})$$

$$p_D(\infty, t_{D1}) = 0 \quad (\text{B.4})$$

The dimensionless parameters are:

$$p_D = \frac{k_H h_i}{141.2qB\mu} (p_i - p_w) \quad (\text{B.5})$$

$$r_{D1} = r/r_w \quad (\text{B.6})$$

$$t_{D1} = \frac{263.7 \cdot 10^{-6} k_H t}{\phi \mu p_i^{-1} r_w^2} \quad (\text{B.7})$$

The relationship between p_D and the Hankel transform $\hat{p}_D$ is obtained from Eq. 3.11:

$$\mathcal{H}\left[\frac{\partial^2 p_D}{\partial r_{D1}^2} + \frac{1}{r_{D1}} \frac{\partial p_D}{\partial r_{D1}}\right] = -\lambda_1^2 \hat{p}_D - \lim_{r_{D1} \rightarrow 0} r_{D1} \frac{\partial p_D}{\partial r_{D1}} \quad (\text{B.8})$$

Combination of Eqs. B.1, B.3 and B.8 yields:

$$d\hat{p}_D/dt_{D1} + \lambda_1^2 \hat{p}_D = 1 \quad (\text{B.9})$$

Equation B.9 was solved using the integrating factor $e^{\lambda_1^2 t_{D1}}$. The integration constant was determined from Eq. B.2.

$$\hat{p}_D(\lambda_1, t_{D1}) = (1 - e^{-\lambda_1^2 t_{D1}})/\lambda_1^2 \quad (\text{B.10})$$

From Eq. 3.16:

$$p_D(r_{D1}, t_{D1}) = \int_0^\infty \hat{p}_D(\lambda_1, t_{D1}) \lambda_1 J_0(\lambda_1 r_{D1}) d\lambda_1 \quad (\text{B.11})$$

The real time solution of the flow problem may be obtained by combining Eqs. B.10 and B.11:

$$p_D(r_{D1}, t_{D1}) = \int_0^\infty \frac{1 - e^{-\lambda_1^2 t_{D1}}}{\lambda_1} J_0(\lambda_1 r_{D1}) d\lambda_1 \quad (\text{B.12})$$

The dimensionless wellbore radius is introduced for convenience:

$$r_{wD} = r_w \sqrt{k_V/k_H/h_i} \quad (\text{B.13})$$

The relationship between confined flow and gravity drainage dimensionless radii are obtained from Eqs. 3.4, B.6 and B.13:

$$r_{D1} = r_D/r_{wD} \quad (\text{B.14})$$

Similarly, from Eqs. 3.3, B.7 and B.13, the relation between t_{D1} and t_{DV} is obtained.

$$t_{D1} = t_{DV}/r_{wD}^2 \quad (\text{B.15})$$

From Eqs. B.12, B.14 and B.15, after changing the integration variable from λ_1 to λ :

$$p_D(r_D, t_{DV}) = \int_0^\infty \frac{J_0(\lambda r_D)}{\lambda} d\lambda - \int_0^\infty e^{-\lambda^2 t_{DV}} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (\text{B.16})$$

Transformation of Eq. B.16 to Laplace space yields the desired solution:

$$\bar{p}_D(r_D, s) = \int_0^\infty \frac{1}{s} \frac{J_0(\lambda r_D)}{\lambda} d\lambda - \int_0^\infty \frac{1}{s + \lambda^2} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (\text{B.17})$$

The pair t_{D1} and r_{D1} is interchangeable with the pair t_{DV} and r_D .

Appendix C. Boulton (1963) Delayed Yield

Boulton (1954b) viewed gravity drainage as a combination of elastic storage at early times and specific yield at late times. According to this theory, an empirical delay index describes the pressure response during the intermediate time period. Boulton (1963) presented delayed yield type curves based on this theory, and these curves will be compared with the analytical model of the present study. From Boulton (1963):

$$\Phi_{wD} = \int_0^\infty [1 - e^{-u_1} (\cosh u_2 + \frac{\alpha t \eta (1 - \lambda^2)}{2u_2} \sinh u_2)] J_0(\frac{\lambda r}{vD}) \frac{d\lambda}{\lambda} \quad (C.1)$$

$$2u_1 = \alpha t \eta (1 + \lambda^2) \quad (C.2)$$

$$2u_2 = \alpha t \sqrt{\eta^2 (1 + \lambda^2)^2 - 4\eta \lambda^2} \quad (C.3)$$

$$v^2 = (\eta - 1)/\eta \quad (C.4)$$

The delay index is α^{-1} , and the parameter D^2 was defined as:

$$D^2 = t_D h_i^2 \alpha^{-1} / t \quad (C.5)$$

The parameter η describes the relationship between elastic storage and specific yield:

$$\eta = \frac{\rho h_i \phi C_t / 144 + \phi}{\rho h_i \phi C_t / 144} \quad (C.6)$$

The total system compressibility for a water well is the contribution from the reservoir rock and the water occupying the pore space. A value for C_t equal to $6 \cdot 10^{-6}$ /psi and the base case parameters defined in Subsection 3.3 were used for the numerical comparison. The resulting value for η was equal to 2000. The calculated values from the Boulton (1963) delayed yield model and the gravity drainage model of the present study are shown in Fig. C.1.

There is no agreement between the two gravity drainage models at early and intermediate flowtimes. After reaching pseudoradial flow, however, the two models agree.

Figure C.1: Dimensionless wellbore pressure drop versus dimensionless horizontal flow time for analytical model and Boulton (1963) delayed yield model.

Compressible flow theory was used by Boulton (1963) to describe the pressure draw-down at early flowtimes. Neuman (1972) and Streltsova (1972) had a similar view of gravity drainage flow. This approach is inadequate for evaluating the pressure response at the wellbore. During the early time period, the wellbore pressure behavior is dominated by wellbore storage effects. The elastic properties of the reservoir system are unimportant.

Appendix D. Numerical Considerations

The evaluation of the integrals in Eqs. 3.15 and 4.1 may be simplified by considering the behavior of the integrands for small values of λ . For both equations, the objective is to replace the lower integration limit equal to zero by a value greater than zero. The value of the integral between zero and the nonzero value will be obtained analytically. Equation 4.1 will be considered first. The integrals in the Green's function solution were:

$$I_1 = -2 \int_0^\infty \frac{\cosh \lambda(z_{D_o} + 1) \cosh \lambda(z_D + 1) e^{-\lambda t_{DV} \tanh \lambda}}{\sinh \lambda \cosh \lambda} J_0(\lambda r_D) d\lambda \\ + 2 \int_0^\infty \frac{\cosh \lambda z_{D_o} \cosh \lambda z_D e^{-\lambda}}{\sinh \lambda} J_0(\lambda r_D) d\lambda \quad (D.1)$$

For x smaller than 0.001, the following approximations are used:

$$e^x = 1 + x \quad (D.2)$$

$$e^{-x} = 1 - x \quad (D.3)$$

For x smaller than 0.001, the following approximations may be obtained using the definitions of the hyperbolic functions:

$$\cosh x = 1 \quad (D.4)$$

$$\sinh x = x \quad (D.5)$$

$$\tanh x = x \quad (D.6)$$

For λr_D less than 0.005, the following approximation is valid:

$$J_0(\lambda r_D) = 1 \quad (D.7)$$

For λ_1 less than 0.001, $\lambda_1 r_D$ less than 0.005 and $\lambda_1^2 t_{DV}$ less than 0.001, the following approximation may be obtained:

$$I_1 = -2 \int_{\lambda_1}^{\infty} \frac{\cosh \lambda(z_{D_0} + 1) \cosh \lambda(z_D + 1) e^{-\lambda t_{DV} \tanh \lambda}}{\sinh \lambda \cosh \lambda} J_0(\lambda r_D) d\lambda \\ + 2 \int_{\lambda_1}^{\infty} \frac{\cosh \lambda z_{D_0} \cosh \lambda z_D e^{-\lambda}}{\sinh \lambda} J_0(\lambda r_D) d\lambda + t_{DV} \lambda_1^2 - 2\lambda_1 \quad (D.8)$$

The Laplace space solution, Eq. 3.15, will be considered next.

$$I_2 = \int_0^{\infty} \frac{1}{s} \frac{J_0(\lambda r_D)}{\lambda} d\lambda - \int_0^{\infty} \frac{e^{\lambda(z_D+2)} + e^{-\lambda z_D}}{e^{2\lambda}(s+\lambda) + s - \lambda} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (D.9)$$

Utilizing Eqs. D.2, D.3 and D.7, it follows for λ_2 less than 0.0005 and $\lambda_2 r_D$ less than 0.005:

$$I_2 = \int_0^{\lambda_2} \frac{1}{s} \frac{\lambda}{\lambda^2 + s\lambda + s} d\lambda + \int_{\lambda_2}^{\infty} \frac{1}{s} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \\ - \int_{\lambda_2}^{\infty} \frac{e^{\lambda(z_D+2)} + e^{-\lambda z_D}}{e^{2\lambda}(s+\lambda) + s - \lambda} \frac{J_0(\lambda r_D)}{\lambda} d\lambda \quad (D.10)$$

For s greater than or equal to 4, and λ_2 less than 0.0005:

$$\int_0^{\lambda_2} \frac{1}{s} \frac{\lambda}{\lambda^2 + s\lambda + s} d\lambda = \frac{1}{s^2} [\lambda_2 - \ln(\lambda_2 + 1)] \quad (D.11)$$

For s between 0 and 4:

$$\int_0^{\lambda_2} \frac{1}{s} \frac{\lambda}{\lambda^2 + s\lambda + s} d\lambda = \frac{1}{2s} \ln \frac{Q^2 + (\lambda_2 + 0.5s)^2}{s} \\ - \frac{1}{2Q} \left[\tan^{-1} \frac{\lambda_2 + 0.5s}{Q} - \tan^{-1} \frac{0.5s}{Q} \right] \quad (D.12)$$

The parameter Q^2 is:

$$Q^2 = (4s - s^2)/4 \quad (D.13)$$

The computer program is listed in Appendix E. The previous modifications greatly reduced the CPU time. Runs were made on Abqaiq and Pangea, School of Earth Sciences, for both the analytical gravity drainage model and the Stream Layer Model numerical simulator (Couri, 1993). The isotropic base case was considered.

The analytical model uses both the Laplace space solution and Green's function solution during the wellbore storage period. The average CPU time needed to calculate each pressure point during this period was 38 seconds on Abqaiq and 24 seconds on Pangea. After the end of wellbore storage, only the Green's function solution was used for calculations. The average CPU time for these calculations was 24 seconds on Abqaiq and 15 seconds on Pangea.

The Stream Layer Model consumed more CPU time than the analytical model. The cumulative CPU time is graphed versus simulated drawdown time in Fig. D.1. The run on Abqaiq was terminated after 20 hours of CPU time, while the run on Pangea lasted for 25 hours. The maximum time value in Fig. D.1 is less than the range of time used for the graphs showed in Sections 3 and 4.

Figure D.1: Cumulative CPU time versus drawdown time for the Stream Layer Model numerical simulator.

Appendix E. Computer Program

```

C-----
C----- GREEN.FOR -----
C-----
C
C   This program calculates the analytical drawdown solution for a
C   gravity drainage well. Two options are available: calculation
C   with and without wellbore storage effects. The timesteps have
C   logarithmic spacing. The confined liquid solution of van
C   Everdingen and Hurst (1949) is calculated for comparison.
C-----
C
C   Active output files:
C
C   (1,file='t_delpw')   Values of time [minutes] and wellbore
C                       pressure drop [psi].
C
C   (2,file='tDV_phiwD') Values of dimensionless vertical time and
C                       dimensionless wellbore pressure drop.
C
C   (3,file='tDV_pwD')   Values of dimensionless vertical time and
C                       dimensionless wellbore pressure drop from
C                       confined liquid solution by VEH.
C
C   (4,file='tDH_phiwD') Values of dimensionless horizontal time and
C                       dimensionless wellbore pressure drop.
C
C   (5,file='tDH_pwD')   Values of dimensionless horizontal time and
C                       dimensionless wellbore pressure drop from
C                       confined liquid solution by VEH.
C
C   Passive output files:
C
C   (11,file='zD0_G')    Values of zD0 and G(zD,rD,zD0,tD).
C
C   (12,file='#Green')   Number of Green's function evaluations for
C                       each wellbore segment.
C
C   (13,file='qD_zD')    Values of qD and zD at each timestep.
C
C   (14,file='#hank1')   Number of function evaluations in subroutine
C                       hankel for Laplace space solution.
C
C   (15,file='#hank2')   Number of function evaluations in subroutine
C                       hankel for Green's function solution.
C
C   (16,file='screen')   Useful runtime information.
C
C   (17,file='cpu')      Cumulative and incremental cpu time in
C                       minutes for different drawdown times.
C-----
C
C   Control parameter:
C
C   wbs = 0: Green's function solution only. No wellbore storage.
C
C   wbs = 1: Green's function and Laplace space solutions. Wellbore
C           storage effects included. The Laplace space solution is
C           no longer used when two consecutive pressure points are
C           within 0.1 percent.
C-----
C
C   double precision qb, kh, kv, skin, hi, visc, dens, por, rw
C   integer wbs, mm, nsteh, msteh

```

```

double precision pi_inv, areawb, c, cd, rwd, hw
integer i, j, k, jj, switch
double precision tmin(90), td(90), thr, zdj, v(20), pwd
double precision delpw(90), phiwd(90), qd(100)
double precision zd1, zd2, s_star, zdstar, dummy, cpusec, last
real zd1_, zd2_
open(1,file='t_delpw',status='unknown')
open(2,file='tDV_phiwD',status='unknown')
open(3,file='tDV_pwD',status='unknown')
open(4,file='tDH_phiwD',status='unknown')
open(5,file='tDH_pwD',status='unknown')
C11      open(11,file='zD0_G',status='unknown')
C12      open(12,file='#Green',status='unknown')
C13      open(13,file='qD_zD',status='unknown')
C14      open(14,file='#hank1',status='unknown')
C15      open(15,file='#hank2',status='unknown')
C16      open(16,file='screen',status='unknown')
open(17,file='cpu',status='unknown')

c
qb = 50.
kh = 240.
kv = 1920.
skin = 0.
hi = 189.
visc = 1.
dens = 62.4
por = .25
rw = 3.18/12.

c
wbs = 1
mm = 10
nsteh = 8
msteh = 1

c
pi_inv = 144./(dens*hi)
areawb = 3.14159*rw*rw
c = (144./5.615)*areawb/dens
cd = 5.615*c*kv/(6.28319*hi*por*pi_inv*hi*hi*kh)
rwd = rw*dsqrt(kv/kh)/hi
hw = hi

c
j = 0
last = .0
write(17,*)'----- Green:'
do 10 i = 2,8
  do 20 k = 1,10,2
    j = j + 1
    tmin(j) = 0.01*10.**((k-1.)/10.)*10.**(i-1)
    thr = tmin(j)/60.
    td(j) = 263.7e-6*kv*thr/(por*visc*pi_inv*hi*hi)
C16      write(16,*)'-----'
C16      &-----'
C16      write(16,*)'-----'
C16      &-----'
C16      write(16,*) 1,'/8', k,'/10',' tD =', td(j)
C16      write(16,*)'-----'
C16      &-----'
      call star3(mm,hw,hi,qb,skin,
&          visc,dens,kh,rwd,td(j),qd,delpw(j),phiwd(j))
      call cputime(cpusec)
      write(17,102) tmin(j), cpusec/60., (cpusec - last)/60.
      last = cpusec
C13      write(13,*)'2 '
C13      write(13,*)'1 -1'
C13      write(13,*)'1 ', hw/hi - 1.
C13      write(13,*)'2 '

```



```

C13          write(13,*)'0 ', hw/hi - 1.
C13          write(13,*)'2 ', hw/hi - 1.
C13          write(13,*) mm
C13          do 30 jj = 1,mm
C13             zdj = (2.*jj - 1.)*hw/(2.*mm*hi) - 1.
C13             write(13,*) qd(jj), zdj
C1330        continue
C13          write(13,*)'-----'
C13          write(13,*)'tmin = ', tmin(j)
C13          write(13,*)'-----'
C13          write(13,*)
20          continue
10          continue
          if (wbs .eq. 1) then
             zd1 = -1.
             zd2 = .0
             switch = 1
             write(17,*)'----- Laplace:'
             do 40 i = 1,j
                s_star = skin/(1. - delpw(i)*pi_inv)
                call find0(zd1,zd2,phiwd(i),rwd,td(i),s_star,zdstar)
                call llnv(nsteh,td(i),v,msteh,phiwd(i),switch,
&                  zdstar,rwd,s_star,cd)
                call cputime(cpusec)
                write(17,102) tmin(i), cpusec/60., (cpusec - last)/60.
                last = cpusec
                delpw(i) = phiwd(i)*141.2*qb*visc/(kh*hi)
                if (dabs((phiwd(i-1) - phiwd(i))/phiwd(i)) .lt. 1.d-3) then
                   goto 41
                endif
                zd1_ = max(-1.,real(zdstar - .1))
                zd2_ = min(.0,real(zdstar + .1))
                zd1 = zd1_
                zd2 = zd2_
40          continue
          endif
41          switch = 4
          dummy = .0
          if (wbs .eq. 1) then
             cd = 5.615*c/(6.28319*hi*por*pi_inv*rw*rw)
          else
             cd = .0
          endif
          write(1,*) j
          write(2,*) j
          write(3,*) j
          write(4,*) j
          write(5,*) j
          do 50 i = 1,j
             write(1,101) tmin(i), delpw(i)
             write(2,101) td(i), phiwd(i)
             write(4,101) td(i)*kh/kv, phiwd(i)
             call llnv(nsteh,td(i)/(rwd*rwd),v,msteh,
&                  pwd,switch,dummy,dummy,skin,cd)
             write(3,101) td(i), pwd
             write(5,101) td(i)*kh/kv, pwd
50          continue
          close(1)
          close(2)
          close(3)
          close(4)
          close(5)
C11          close(11)
C12          close(12)
C13          close(13)
C14          close(14)

```

```

C15      close(15)
C16      close(16)
        close(17)
101      format(g9.2,f9.4)
102      format(g9.2,f9.4,f9.4)
        end

c-----
c----- START subroutine star3 -----
c-----
c
c      This subroutine calculates the dimensionless flowrates at the
c      wellbore nodes and the wellbore liquid height.
c-----
c
c      Input:  mm = number of segments (max = 100)
c              hw = assumed value of wellbore height [ft]
c              hi = reservoir thickness [ft]
c              qb = flowrate [bbl/day]
c              skin = skin factor [-]
c              visc = viscosity [cp]
c              dens = density [lbm/cuft]
c              kh = horizontal permeability [md]
c              rwd = dimensionless wellbore radius [-]
c              td = dimensionless time [-]
c
c      Output: hw = calculated value of wellbore height [ft]
c              qd(mm) = dimensionless flowrates at the nodes [-]
c              delpw = wellbore pressure drop [psi]
c              phiwd = dimensionless wellbore pressure drop [-]
c-----
c      subroutine star3(mm,hw,hi,qb,skin,
c      & visc,dens,kh,rwd,td,qd,delpw,phiwd)
c      integer mm
c      double precision hw, hi, qb, skin, visc, dens, kh, rwd, td, rdstar
c      double precision qd(100), a(100,100), b(100,1), istar(100,100)
c      double precision zdj, xmdown, xmup, beta, sigma, alpha
c      double precision delpw, hwnew, error, phiwd, left, right
c      integer ihw, j, m, ngauss, mgauss, n
c
c      rdstar = 1.5*rwd
c
c      do 1 ihw = 1,20
c        alpha = hw/(6.28319*hi)
c        beta = (dlog(rdstar/rwd) + skin)/6.28319
C16      write(16,101)'Wellbore height iteration = ', ihw
C16      write(16,102)'Wellbore height [ft] = ', hw
c        do 10 j = 1,mm
c          zdj = (2.*j - 1.)*hw/(2.*mm*hi) - 1.
c          do 11 m = 1,mm
C16      write(16,*)'*****', j, m, mm
c            xmdown = (m-1.)*hw/(mm*hi) - 1.
c            xmup = m*hw/(mm*hi) - 1.
C11      write(11,*) 'zD = ', zdj
C11      write(11,*) 'tD = ', td
C12      write(12,*) 'zD = ', zdj
C12      write(12,*) 'tD = ', td
c          if (zdj .gt. xmdown .and. zdj .lt. xmup) then
c            call integ3(xmdown,zdj,left,rdstar,zdj,td,n)
C12      write(12,103)'zD0 interval: ', xmdown, zdj
C12      write(12,*) n,' Greens function evaluations.'
c            call integ3(zdj,xmup,right,rdstar,zdj,td,n)
C12      write(12,103)'zD0 interval: ', zdj, xmup
C12      write(12,*) n,' Greens function evaluations.'
C12      write(12,*)'-----

```

```

            istar(j,m) = left + right
        else
            call integ3(xmdown,xmup,istar(j,m),rdstar,zdj,td,n)
C12      write(12,103)'zD0 interval: ', xmdown, xmup
C12      write(12,*) n,' Greens function evaluations.'
C12      write(12,*)'-----'
        endif
11      continue
10      continue
        b(1,1) = mm*(istar(1,1) + beta)
        do 20 j = 2,mm
            b(j,1) = mm*istar(j,1)
20      continue
        do 21 j = 1,mm
            a(j,1) = alpha
21      continue
        do 22 m = 2,mm
            a(1,m) = istar(1,1) + beta - istar(1,m)
22      continue
        do 23 j = 2,mm
            do 24 m = 2,mm
                a(j,m) = istar(j,1) - istar(j,m)
                if (j .eq. m) then
                    a(j,m) = a(j,m) - beta
                endif
            enddo
24      continue
23      continue
        ngauss = mm
        mgauss = 1
        call gaussj(a,ngauss,b,mgauss)
        sigma = .0
        phiwd = b(1,1)
        do 30 m = 2,mm
            qd(m) = b(m,1)
            sigma = sigma + qd(m)
30      continue
        qd(1) = mm - sigma
        delpw = phiwd*141.2*qb*visc/(kh*hi)
        hwnew = hi - 144.*delpw/dens
        error = (hwnew - hw)/hwnew
        if (dabs(error) .lt. 1.d-4) then
            goto 41
        endif
1      continue
        pause 'No convergence for wellbore height.'
41      return
101     format(1x,a,i2)
102     format(1x,a,f5.1)
103     format(1x,a,f6.3,2x,f6.3)
        end

c-----
c----- END subroutine star3 -----
c-----
c-----
c-----
c----- START subroutine gaussj -----
c-----
c-----
c
c      This subroutine is from Press et al, Numerical Recipes.
c
c      A is an input matrix of N by N elements, stored in an array of
c      physical dimensions NP by NP. B is an input matrix of N by M
c      containing the M right hand side vectors, stored in an array of
c      physical dimensions NP by MP. On output, A is replaced by its
c      matrix inverse, and B is replaced by the corresponding set of

```

```

c      solution vectors.
c
c-----
      SUBROUTINE GAUSSJ(A,N,B,M)
      integer n, m, nmax
      PARAMETER (NMAX=100)
      double precision A(100,100), B(100,1)
      integer IPIV(NMAX), INDXR(NMAX), INDXC(NMAX)
      integer i, j, k, irow, icol, l, ll
      double precision big, dum, pivinv
      DO 11 J=1,N
         IPIV(J)=0
11      CONTINUE
      DO 22 I=1,N
         BIG=0.
         DO 13 J=1,N
            IF(IPIV(J).NE.1)THEN
               DO 12 K=1,N
                  IF (IPIV(K).EQ.0) THEN
                     IF (dABS(A(J,K)).GE.BIG)THEN
                        BIG=dABS(A(J,K))
                        IROW=J
                        ICOL=K
                     ENDIF
                  ELSE IF (IPIV(K).GT.1) THEN
                     PAUSE 'Singular matrix'
                  ENDIF
               CONTINUE
            ENDIF
13      CONTINUE
            IPIV(ICOL)=IPIV(ICOL)+1
            IF (IROW.NE.ICOL) THEN
               DO 14 L=1,N
                  DUM=A(IROW,L)
                  A(IROW,L)=A(ICOL,L)
                  A(ICOL,L)=DUM
14      CONTINUE
               DO 15 L=1,M
                  DUM=B(IROW,L)
                  B(IROW,L)=B(ICOL,L)
                  B(ICOL,L)=DUM
15      CONTINUE
            ENDIF
            INDXR(I)=IROW
            INDXC(I)=ICOL
            IF (A(ICOL,ICOL).EQ.0.) PAUSE 'Singular matrix.'
            PIVINV=1./A(ICOL,ICOL)
            A(ICOL,ICOL)=1.
            DO 16 L=1,N
               A(ICOL,L)=A(ICOL,L)*PIVINV
16      CONTINUE
            DO 17 L=1,M
               B(ICOL,L)=B(ICOL,L)*PIVINV
17      CONTINUE
            DO 21 LL=1,N
               IF(LL.NE.ICOL)THEN
                  DUM=A(LL,ICOL)
                  A(LL,ICOL)=0.
                  DO 18 L=1,N
                     A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18      CONTINUE
                  DO 19 L=1,M
                     B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19      CONTINUE
                  ENDIF
21      CONTINUE

```

```

22      CONTINUE
      DO 24 L=N,1,-1
        IF (INDXR(L) .NE. INDXC(L)) THEN
          DO 23 K=1,N
            DUM=A(K,INDXR(L))
            A(K,INDXR(L))=A(K,INDXC(L))
            A(K,INDXC(L))=DUM
          23      CONTINUE
        ENDIF
      24      CONTINUE
      RETURN
      END

C-----
C-----      END subroutine gaussj      -----
C-----
C
C-----
C-----      START module integ3      -----
C-----
C
C      This module performs Romberg integration between a closed interval
C      [a,b] using subroutines by Press et al., Numerical Recipes.
C
C      It is used in star3 to integrate the Green's function between
C      zD0 = x_m-1 and zD0 = x_m.
C
C      When calling the internal subroutines, the last character in the
C      names is replaced by '3' to avoid mixup with subroutine integ4.
C
C      Input:  a = lower limit of zD0
C              b = upper limit of zD0
C              rd = dimensionless sandface radius
C              zdj = value of zD
C              td = value of tD
C
C      Output: ss = calculated integral
C              count = number of function evaluations
C
C-----
C-----      START subroutine integ3      -----
C-----
C      SUBROUTINE integ3(A,B,SS,rd,zdj,td,count)
C      double precision a, b, ss, eps, dummy, dss, rd, zdj, td
C      integer jmax, jmaxp, k, km, j, l, count
C      PARAMETER (EPS=5.d-5, JMAX=20, JMAXP=JMAX+1, K=5, KM=4)
C      double precision S(JMAXP), H(JMAXP)
C      count = 0
C      dummy = .0
C      H(1)=1.
C      DO 11 J=1,JMAX
C12      write(12,*) 'TRAPZ3 ', j
C      CALL TRAPZ3(A,B,S(J),J,rd,zdj,td,count)
C      IF (J.GE.K) THEN
C        L=J-KM
C        CALL POLIN3(H(L),S(L),K,dummy,SS,DSS)
C        IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
C      ENDIF
C      S(J+1)=S(J)
C      H(J+1)=0.25*H(J)
C11      CONTINUE
C      PAUSE 'integ3: Too many steps.'
C      END

C-----
C-----      END subroutine integ3      -----
C-----
C

```

```

C-----
C----- START subroutine polin3 -----
C-----
SUBROUTINE POLIN3(XA,YA,N,X,Y,DY)
integer n, nmax
double precision xa(n), ya(n), x, y, dy
PARAMETER (NMAX=10)
double precision C(NMAX),D(NMAX), dif, dift, ho, hp, w, den
integer ns, i, m
NS=1
DIF=dABS(X-XA(1))
DO 11 I=1,N
  DIFT=dABS(X-XA(I))
  IF (DIFT.LT.DIF) THEN
    NS=I
    DIF=DIFT
  ENDIF
  C(I)=YA(I)
  D(I)=YA(I)
11 CONTINUE
Y=YA(NS)
NS=NS-1
DO 13 M=1,N-1
  DO 12 I=1,N-M
    HO=XA(I)-X
    HP=XA(I+M)-X
    W=C(I+1)-D(I)
    DEN=HO-HP
    IF (DEN.EQ.0.) PAUSE
    DEN=W/DEN
    D(I)=HP*DEN
    C(I)=HO*DEN
12 CONTINUE
  IF (2*NS.LT.N-M) THEN
    DY=C(NS+1)
  ELSE
    DY=D(NS)
    NS=NS-1
  ENDIF
  Y=Y+DY
13 CONTINUE
RETURN
END

C-----
C----- END subroutine polin3 -----
C-----
C-----
C----- START subroutine trapz3 -----
C-----
SUBROUTINE TRAPZ3(A,B,S,N,rd,zdj,td,count)
double precision a, b, s, f, fl, del, x, sum, rd, zdj, td
integer n, it, tnm, j, count
IF (N.EQ.1) THEN
  call gree3(zdj,rd,a,td,f)
  call gree3(zdj,rd,b,td,fl)
  count = count + 2
  S=0.5*(B-A)*(f + fl)
  IT=1
ELSE
  TNM=IT
  DEL=(B-A)/TNM
  X=A+0.5*DEL
  SUM=0.
  DO 11 J=1,IT
    call gree3(zdj,rd,x,td,f)

```

```

        count = count + 1
        SUM=SUM + f
        X=X+DEL
11      CONTINUE
        S=0.5*(S+(B-A)*SUM/TNM)
        IT=2*IT
      ENDIF
      RETURN
    END

C-----
C----- END subroutine trapz3 -----
C-----
C-----
C----- START subroutine gree3 -----
C-----
C
C   This subroutine calculates the Green's function.
C
C   Input:  zD, rD, zD0, tD
C
C   Output: G(zD,rD,zD0,tD)
C-----
C
C   subroutine gree3(zd,rd,zd0,td,gg)
C     double precision zd, rd, zd0, td, gg
C     double precision r, c(3), area, den1, den2, lam1, areal
C     integer nfcn, count
C     real lam1_
C
C     nfcn = 2
C     r = rd
C     c(1) = zd
C     c(2) = zd0
C     c(3) = td
C
C     den1 = dsqrt(rd*rd + (zd - zd0)*(zd - zd0))
C     den2 = dsqrt(rd*rd + (zd + zd0)*(zd + zd0))
C     lam1_ = min(1.e-6,real(2.5d-5/(r*r)),real(1.d-3/c(3)))
C     lam1 = sqrt(lam1_)
C     areal = c(3)*lam1_*lam1 - 2.*lam1
C     area = 1./den1 + 1./den2 + areal
C     call hankel(nfcn,r,c,area,count)
C     gg = .07957747*(1./den1 + 1./den2 + areal + area)
C11    write(11,101) zd0, gg
C15    write(15,*)'-----'
C15    write(15,101) zd0, gg
C15    write(15,*) count
C     return
101   format(f9.4,f9.4)
C     end

C-----
C----- END subroutine gree3 -----
C-----
C-----
C----- END module integ3 -----
C-----
C-----
C----- START module hankel -----
C-----
C
C   This module performs Romberg integration from zero to infinity for
C   a function containing the Bessel function J0, and its main use is
C   to calculate the inverse Hankel transform. The total interval
C   is subdivided into intervals according to where J0 is zero.

```

```

c
c      Input:  nfcn = function index
c              r = r-value
c              c(3) = pertinent parameters
c              area = already calculated contribution to total
c
c      Output: area = total integral from zero to infinity
c
c-----
      subroutine hankel(nfcn,r,c,area,count)
      integer n, nfcn, count
      double precision area, a, b, ss, tol
      double precision r, c(3), rr, cc(3), area0
      real a_
      parameter (tol = 1.d-5)
      common /hank/ rr, cc
      rr = r
      cc(1) = c(1)
      cc(2) = c(2)
      cc(3) = c(3)
      area0 = area
      area = .0
      if (nfcn .eq. 2) then
         a_ = min(1.e-6,real(2.5d-5/(r*r)),real(1.d-3/c(3)))
         a = sqrt(a_)
      else
         a_ = min(real(.005/r),.0005)
         a = a_
      endif
      do 10 n = 1,5000
         call j0_0(n,b)
         call integ4(nfcn,a,b,ss,count)
         area = area + ss
         if (dabs(ss/(area + area0)) .lt. tol) then
            return
         else
            a = b
         endif
10      continue
      pause 'hankel: 5000 iterations exceeded'
      stop
      end
c-----
c----- START subroutine j0_0 -----
c-----
c
c      This subroutine calculates the zeros of J0 using the formula given
c      by Watson, A Treatise on the Theory of Bessel Functions, 2nd ed.,
c      1952, p. 505. For small values of n the values given in Table VII
c      on page 748 are used.
c
c-----
c
c      Input:  n = index for the nth root of J0  [INTEGER]
c
c      Output: xn = x-value corresponding to n  [DBL PREC]
c
c-----
      subroutine j0_0(n,xn)
      integer n
      double precision xn, pi
      parameter (pi = 3.141592654)
      if (n .gt. 2) then
         xn = n*pi - pi/4. + 1./(8.*(n*pi - pi/4.)) - 31./(384.*
+         (n*pi - pi/4.）**3) + 3779./(15360.*(n*pi - pi/4.）**5)
      elseif (n .eq. 1) then

```



```

      xn = 2.4048256
    elseif (n .eq. 2) then
      xn = 5.5200781
    else
      pause 'j0_0: n out of bound'
    endif
    return
  end

c-----
c----- END subroutine j0_0 -----
c-----
c
c-----
c----- START subroutine bessj0 -----
c-----
c
c      Source: Press et al., Numerical Recipes.
c
c-----
      subroutine BESSJ0(X,j0)
      double precision x, j0
      double precision Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,
+      R6,S1,S2,S3,S4,S5,S6, ax, z, xx
      DATA P1,P2,P3,P4,P5/1.D0,-.1098628627D-2,.2734510407D-4,
*      -.2073370639D-5,.2093887211D-6/, Q1,Q2,Q3,Q4,Q5/-.1562499995D-
*1,
*      .1430488765D-3,-.6911147651D-5,.7621095161D-6,-.934945152D-7/
      DATA R1,R2,R3,R4,R5,R6/57568490574.D0,-13362590354.D0,651619640.7D
*0,
*      -11214424.18D0,77392.33017D0,-184.9052456D0/,
*      S1,S2,S3,S4,S5,S6/57568490411.D0,1029532985.D0,
*      9494680.718D0,59272.64853D0,267.8532712D0,1.D0/
      IF (dABS(X).LT.8.) THEN
        Y=X**2
        j0=(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6))))
*      /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6))))
      ELSE
        AX=dABS(X)
        Z=8./AX
        Y=Z**2
        XX=AX-.785398164
        j0=dSQRT(.636619772/AX)*(dCOS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y
*      *P5))))-Z*dSIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5))))
      ENDIF
      RETURN
      END

c-----
c----- END subroutine bessj0 -----
c-----
c
c-----
c----- START submodule integ4 -----
c-----
c
c      This module performs Romberg integration between a closed interval
c      [a,b] using subroutines by Press et al., Numerical Recipes.
c
c      Input:  nfcn = index for function
c              a = lower limit
c              b = upper limit
c
c      Output: ss = calculated integral
c              count = number of function evaluations
c
c-----
c----- START subroutine integ4 -----

```

```

c-----
SUBROUTINE integ4(nfcn,A,B,SS,count)
double precision a, b, ss, eps, dummy, dss
integer jmax, jmaxp, k, km, j, l, nfcn, count
PARAMETER(EPS=1.d-5,JMAX=20,JMAXP=JMAX+1,K=5,KM=4)
double precision S(JMAXP),H(JMAXP)
dummy = .0
count = 0
H(1)=1.
DO 11 J=1,JMAX
  CALL TRAPZD(nfcn,A,B,S(J),J,count)
  IF (J.GE.K) THEN
    L=J-KM
    CALL POLINT(H(L),S(L),K,dummy,SS,DSS)
    IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
  ENDIF
  S(J+1)=S(J)
  H(J+1)=0.25*H(J)
11 CONTINUE
PAUSE 'integ4: Too many steps.'
END

c-----
c----- END subroutine integ4 -----
c-----
c-----
c-----
c----- START subroutine polint -----
c-----
SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
integer n, nmax
double precision xa(n), ya(n), x, y, dy
PARAMETER (NMAX=10)
double precision C(NMAX),D(NMAX), dif, dift, ho, hp, w, den
integer ns, i, m
NS=1
DIF=dABS(X-XA(1))
DO 11 I=1,N
  DIFT=dABS(X-XA(I))
  IF (DIFT.LT.DIF) THEN
    NS=I
    DIF=DIFT
  ENDIF
  C(I)=YA(I)
  D(I)=YA(I)
11 CONTINUE
Y=YA(NS)
NS=NS-1
DO 13 M=1,N-1
  DO 12 I=1,N-M
    HO=XA(I)-X
    HP=XA(I+M)-X
    W=C(I+1)-D(I)
    DEN=HO-HP
    IF (DEN.EQ.0.) PAUSE
    DEN=W/DEN
    D(I)=HP*DEN
    C(I)=HO*DEN
12 CONTINUE
  IF (2*NS.LT.N-M) THEN
    DY=C(NS+1)
  ELSE
    DY=D(NS)
    NS=NS-1
  ENDIF
  Y=Y+DY
13 CONTINUE

```

```

RETURN
END
C-----
C----- END subroutine polint -----
C-----
C-----
C----- START subroutine trapzd -----
C-----
SUBROUTINE TRAPZD(nfcn,A,B,S,N,count)
double precision a, b, s, f, f1, del, x, sum
integer n, it, tnm, j, nfcn, count
IF (N.EQ.1) THEN
    call fcn(nfcn,a,f)
    call fcn(nfcn,b,f1)
    count = count + 2
    S=0.5*(B-A)*(f + f1)
    IT=1
ELSE
    TNM=IT
    DEL=(B-A)/TNM
    X=A+0.5*DEL
    SUM=0.
    DO 11 J=1,IT
        call fcn(nfcn,x,f)
        count = count + 1
        SUM=SUM + f
        X=X+DEL
11    CONTINUE
    S=0.5*(S+(B-A)*SUM/TNM)
    IT=2*IT
ENDIF
RETURN
END
C-----
C----- END subroutine trapzd -----
C-----
C-----
C----- START subroutine fcn -----
C-----
C
C    This subroutine calculates the integrand.
C
C    nfcn = 1: Uniform flux Laplace domain solution.
C    nfcn = 2: Green's function in real time.
C
C-----
subroutine fcn(nfcn,lambda,funk)
integer nfcn
double precision lambda, funk, r, c(3)
double precision aa, bb, j0, arg, dummy, lam, term1, term2
common /hank/ r, c
if (nfcn .eq. 1) then
    arg = lambda*r
    call bessj0(arg,j0)
    if (lambda .le. 100.) then
        dummy = (c(2) + lambda)*dexp(2.*lambda) + c(2) - lambda
        aa = dexp(lambda*(c(1) + 2.))/dummy
        bb = dexp(-lambda*c(1))/dummy
    else
        if (c(2) .gt. 1.d99) then
            pause'fcn: s out of bound'
        endif
        dummy = c(2) + lambda
        aa = dexp(lambda*c(1))/dummy
    endif
endif

```

```

        bb = dexp(-lambda*(c(1) + 2.))/dummy
    endif
    funk = j0*(1./c(2) - aa - bb)/lambda
    dummy = c(3)
    return
elseif (nfcn .eq. 2) then
    lam = lambda
    arg = lam*r
    call bessj0(arg,j0)
    if (lam .gt. 300.) then
        term2 = .0
        term1 = dexp(lam*(c(2) + c(1) - c(3)))
    else
        term2 = dexp(-lam)*dcosh(lam*c(2))*dcosh(lam*c(1))/dsinh(lam)
        term1 = dexp(-lam*c(3)*dtanh(lam))*dcosh(lam*(c(2)+1.))
        &      *dcosh(lam*(c(1)+1.))/(dsinh(lam)*dcosh(lam))
    endif
    funk = 2.*(term2 - term1)*j0
    return
else
    pause'integ4: nfcn out of bound'
    return
endif
end
c-----
c----- END subroutine fcn -----
c-----
c----- END submodule integ4 -----
c-----
c----- END module hankel -----
c-----
c-----
c----- START subroutine linv -----
c-----
c
c   This subroutine uses Algorithm 368 Numerical Inversion of Laplace
c   Transforms (D5) presented by H. Stehfest, Communications of the
c   ACM, January 1970, pp.47-49.
c-----
c
c   Input:  n = optimum number depending upon the function
c
c           t = argument of F(t)
c
c           v(20) = array used when n = m
c
c           m = formal parameter
c               * first call:  m different from n
c               * subsequent calls for the same n:  m equal to n
c
c           switch = function switch
c               * switch = 4:  VEH confined solution
c               * switch = 1:  Gravity drainage uniform flux solution
c
c           zdstar = equivalent average potential point (switch = 1)
c
c           rwd = dimensionless wellbore radius (switch = 1)
c
c           skin = skin factor
c
c           cd = dimensionless wellbore storage coefficient
c

```

```

c
c      Output:  v(20) = array returned for use if next call is for the
c                  same n
c
c                  m = parameter equal to n
c
c                  fa = the calculated approximation of F(t)
c
-----
c
c      Valid range of input parameter:  n = 2,4,...,20
c
-----
c      subroutine linv(n,t,v,m,fa,switch,zdstar,rwd,skin,cd)
c      double precision v(20), fa, t, zdstar, rwd, skin, cd
c      integer n, m, switch
c      double precision h(10), a, g(21), s, p
c      integer i, ih, k, nh, sn, start, end
C14      if (switch .eq. 1) then
C14          write(14,*)'-----'
C14          write(14,*)'tD = ', t
C14      endif
      if (m .eq. n) then
          goto 31
      endif
      g(1) = 1.
      nh = n/2
      do 10 i = 1,n
          g(i+1) = g(i)*i
10      continue
      h(1) = 2./g(nh)
      do 20 i = 2,nh
          h(i) = i**nh*g(2*i+1)/(g(nh-i+1)*g(i+1)*g(i))
20      continue
      sn = 2*mod(nh,2) - 1
      do 30 i = 1,n
          v(i) = .0
          if (i .lt. nh) then
              end = i
          else
              end = nh
          endif
          start = int((i + 1)/2)
          do 40 k = start,end
              v(i) = v(i) + h(k)/(g(i-k+1)*g(2*k-i+1))
40      continue
          v(i) = sn*v(i)
          sn = -sn
30      continue
      m = n
31      fa = .0
      a = .693147/t
      do 50 i = 1,n
          s = i*a
          if (switch .eq. 1) then
              call fcn1(s,p,zdstar,rwd,cd,skin)
          elseif (switch .eq. 4) then
              call fcn4(s,p,skin,cd)
          else
              pause 'LINV: switch out of bound'
          endif
          fa = fa + v(i)*p
50      continue
      fa = a*fa
      return
      end

```

```

C-----
C----- END subroutine linv -----
C-----
C
C-----
C----- START subroutine fcn1 -----
C-----
C
C      switch = 1: Gravity drainage uniform flux solution.
C
C-----
C      subroutine fcn1(s,p,zdstar,rd,cd,skin)
C      double precision s, p, zdstar, rd, cd, skin, dummy, c(3)
C      double precision lam1, areal, qq, q
C      integer nfcn, count
C      real lam1_
C      nfcn = 1
C      c(1) = zdstar
C      c(2) = s
C      c(3) = .0
C      lam1_ = min(real(.005/rd),.0005)
C      lam1 = lam1_
C      if (s .ge. 4.) then
C          areal = (lam1 - dlog(lam1 + 1.))/(s*s)
C      else
C          qq = .25*(4.*s - s*s)
C          q = dsqrt(qq)
C          areal = .5*dlog((qq + (lam1 + .5*s)*(lam1 + .5*s))/s)/s
C          areal = areal - .5*(datan((lam1 + .5*s)/q) - datan(.5*s/q))/q
C      endif
C      p = areal
C      call hankel(nfcn,rd,c,p,count)
C      p = p + areal
C      dummy = s*(1. + cd*s*(s*p + skin))
C      p = (s*p + skin)/dummy
C14      write(14,*)'s = ', s, ' count = ', count
C      return
C      end
C
C-----
C----- END subroutine fcn1 -----
C-----
C
C-----
C----- START subroutine fcn4 -----
C-----
C
C      switch = 4: VEH confined solution.
C
C      IBC: Constant rate with skin and wellbore storage.
C      OBC: Infinite acting.
C
C-----
C      subroutine fcn4(s,p,skin,cd)
C      double precision s, p, skin, cd
C      double precision i0, k0, i1, k1
C      call mod_0(dsqrt(s),i0,k0)
C      call mod_1(dsqrt(s),i1,k1)
C      p = (k1 + cd*dsqrt(s)*k0 + s*cd*skin*k1)*s*dsqrt(s)
C      p = (k0 + skin*dsqrt(s)*k1)/p
C      return
C      end
C
C-----
C----- END subroutine fcn4 -----
C-----
C
C-----

```

```

c----- START subroutine mod_0 -----
c
c
c   This subroutine calculates the modified Bessel functions of the
c   first and second kind of order zero, I0(x) and K0(x), using the
c   polynomial approximations given by Abramowitz and Stegun.
c
c-----
c
c   Input:      x = argument  [DBL PREC]
c
c   Output:     i0 = I0(x)    [DBL PREC]
c               k0 = K0(x)    [DBL PREC]
c
c-----
c
c   Valid range of input parameter:  x = <0,*>
c-----
c
c   subroutine mod_0(x,i0,k0)
c   double precision x, i0, k0
c   double precision t, u
c   if (x .le. .0) then
c       write(*,*)'--- mod_0 out of bound ---'
c       stop
c   elseif(x .le. 2.) then
c       t = x*x/14.0625
c       u = x*x/4.
c       i0 = 1. + t*(3.5156229 + t*(3.0899424 + t*(1.2067492 + t*(.2659
+       732 + t*(.0360768 + .0045813*t))))))
c       k0 = -.5*log(u)*i0 - .57721566 + u*(.42278420 + u*(.23069756 +
+       u*(.03488590 + u*(.00262698 + u*(.00010750 + .00000740*u))
+       )))
c       return
c   elseif(x .le. 3.75) then
c       t = x*x/14.0625
c       u = 2./x
c       i0 = 1. + t*(3.5156229 + t*(3.0899424 + t*(1.2067492 + t*(.2659
+       732 + t*(.0360768 + .0045813*t))))))
c       k0 = (exp(-x)/sqrt(x))*(1.25331414 - u*(.07832358 - u*(.0218956
+       8 - u*(.01062446 - u*(.00587872 - u*(.00251540 - .00053208
+       *u))))))
c       return
c   else
c       t = 3.75/x
c       u = 2./x
c       i0 = (exp(x)/sqrt(x))*(.39894228 + t*(.01328592 + t*(.00225319
+       + t*(-.00157565 + t*(.00916281 + t*(-.02057706 + t*(.02635
+       537 + t*(-.01647633 + .00392377*t)))))))))
c       k0 = (exp(-x)/sqrt(x))*(1.25331414 - u*(.07832358 - u*(.0218956
+       8 - u*(.01062446 - u*(.00587872 - u*(.00251540 - .00053208
+       *u))))))
c       return
c   endif
c   end
c-----
c
c----- END subroutine mod_0 -----
c
c----- START subroutine mod_1 -----
c
c
c   This subroutine calculates the modified Bessel functions of the
c   first and second kind of order one, I1(x) and K1(x), using the

```

```

c      polynomial approximations given by Abramowitz and Stegun.
c
c-----
c
c      Input:      x = argument  [DBL PREC]
c
c      Output:     i1 = I1(x)   [DBL PREC]
c
c                  k1 = K1(x)   [DBL PREC]
c
c-----
c
c      Valid range of input parameter:  x = <0,*>
c
c-----
c      subroutine mod_1(x,i1,k1)
c      double precision x, i1, k1
c      double precision t, u
c      if (x .le. .0) then
c        write(*,*)'--- mod_1 out of bound ---'
c        stop
c      elseif(x .le. 2.) then
c        t = x*x/14.0625
c        u = x*x/4.
c        i1 = x*(.5 + t*(.87890594 + t*(.51498869 + t*(.15084934 + t*(.0
+        2658733 + t*(.00301532 + .00032411*t))))))
c        k1 = .5*log(u)*i1 + 1./x + u*(.15443144 + u*(-.67278579 + u*(-.
+        18156897 + u*(-.01919402 + u*(-.00110404 - .00004686*u))))
+        )/x
c        return
c      elseif(x .le. 3.75) then
c        t = x*x/14.0625
c        u = 2./x
c        i1 = x*(.5 + t*(.87890594 + t*(.51498869 + t*(.15084934 + t*(.0
+        2658733 + t*(.00301532 + .00032411*t))))))
c        k1 = (exp(-x)/sqrt(x))*(1.25331414 + u*(.23498619 + u*(-.036556
+        20 + u*(.01504268 + u*(-.00780353 + u*(.00325614 - .000682
+        45*u))))))
c        return
c      else
c        t = 3.75/x
c        u = 2./x
c        i1 = (exp(x)/sqrt(x))*(.39894228 + t*(-.03988024 + t*(-.0036201
+        8 + t*(.00163801 + t*(-.01031555 + t*(.02282967 + t*(-.028
+        95312 + t*(.01787654 - .00420059*t))))))
c        k1 = (exp(-x)/sqrt(x))*(1.25331414 + u*(.23498619 + u*(-.036556
+        20 + u*(.01504268 + u*(-.00780353 + u*(.00325614 - .000682
+        45*u))))))
c        return
c      endif
c      end
c
c-----
c-----
c-----
c-----
c      START subroutine find0
c-----
c
c      This subroutine finds the point z where f(z) is equal to 0. The
c      method used is given by Gerald & Wheatley, 4th edition, on p. 11,
c      and it is referred to as Modified Linear Interpolation Method.
c
c-----
c
c      Input:  zd1 = lower bracket

```



```

c          zd2 = upper bracket
c          phiwd = phiwD calculated using uniform potential
c          rwd = dimensionless wellbore radius
c          td = dimensionless time
c          s_star = gravity drainage skin factor
c
c          Output: zdstar = equivalent average potential point
c
c-----
c          subroutine find0(zd1,zd2,phiwd,rwd,td,s_star,zdstar)
c          double precision zd1, zd2, phiwd, rwd, td, s_star, zdstar
c          double precision f1, f2, fsave, zd3, f3
c          double precision v(20), cd, fa
c          integer i, n, m, switch
c          switch = 1
c          n = 8
c          m = 1
c          cd = .0
c          call linv(n,td,v,m,fa,switch,zd1,rwd,s_star,cd)
C16      write(16,*) zd1, fa, phiwd
c          .1 = fa - phiwd
c          call linv(n,td,v,m,fa,switch,zd2,rwd,s_star,cd)
C16      write(16,*) zd2, fa, phiwd
c          f2 = fa - phiwd
c          fsave = f1
c          if (f1*f2 .gt. .0) then
c              pause'find0: f1 and f2 must be of opposite sign'
c              return
c          endif
c          do 10 i = 1,50
c              zd3 = zd2 - f2*(zd2 - zd1)/(f2 - f1)
c              call linv(n,td,v,m,fa,switch,zd3,rwd,s_star,cd)
C16      write(16,101)'***** find0: i = ', i,'      zD3 = ', zd3,
C16      &          '      fa = ', fa
c              f3 = fa - phiwd
c              if (dabs(f3/phiwd) .lt. 1.d-4) then
c                  zdstar = zd3
c                  return
c              endif
c              if (f3*f1 .lt. .0) then
c                  zd2 = zd3
c                  f2 = f3
c                  if (f3*fsave .gt. .0) then
c                      f1 = .5*f1
c                  endif
c              else
c                  zd1 = zd3
c                  f1 = f3
c                  if (f3*fsave .gt. .0) then
c                      f2 = .5*f2
c                  endif
c              endif
c              fsave = f3
10      continue
c          pause'find0: no convergence'
c          return
101     format(1x,a,i2,2(a,f7.4))
c          end
c-----
c----- END subroutine find0 -----
c-----
c----- START subroutine cputime -----
c-----
c

```

```
c   This subroutine returns the cpu time in seconds.  It works on
c   Abqaiq and Pangea compilers.
```

```
c-----
c       subroutine cputime(cpusec)
c       real tarray(2), etime
c       double precision cpusec
c       cpusec = etime(tarray)
c       return
c       end
c-----
c----- END subroutine cputime -----
c-----
```

**DATE
FILMED**

12 / 16 / 93

END