

1 of 3

DOE/BC/14899-4
Distribution Category UC-122

A FINITE DIFFERENCE MODEL FOR
FREE SURFACE GRAVITY DRAINAGE

SUPRI TR 96

By
Francisco Roberto Couri
Henry J. Ramey, Jr.

September 1993

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

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

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

Prepared by
Stanford University
Petroleum Research Institute
Stanford, CA 94305-4042

MASTER

8

Abstract

The unconfined gravity flow of liquid with a free surface into a well is a classical well test problem which has not been well understood by either hydrologists or petroleum engineers. Paradigms have led many authors to treat an incompressible flow as compressible flow to justify the delayed yield behavior of a time-drawdown test.

A finite-difference model has been developed to simulate the free surface gravity flow of an unconfined single phase, infinitely large reservoir into a well. The model was verified with experimental results in sandbox models in the literature and with classical methods applied to observation wells in the Groundwater literature. The simulator response was also compared with analytical *Theis* (1935) and *Ramey et al.* (1989) approaches for wellbore pressure at late producing times.

The seepage face in the sandface and the delayed yield behavior were reproduced by the model considering a small liquid compressibility and incompressible porous medium.

The potential buildup (recovery) simulated by the model evidenced a different phenomenon from the drawdown, contrary to statements found in the Groundwater literature. Graphs of buildup potential vs. time, buildup seepage face length vs. time, and free surface head and sand bottom head radial profiles evidenced that the liquid refills the desaturating cone as a flat moving surface. The late time pseudo radial behavior was only approached after exaggerated long times.

Acknowledgements

This study would not be possible to be completed without the help and major financial support of my employer *PETROBRAS*, the Department of Energy under Heavy Oil Grant No. DE-FG22-93BC14899, and under Geothermal Contract No. DE-AS07-84ID12529 and Grant No. DE-FG07-90ID12934, and Stanford University.

Table of Contents

Abstract	i
Acknowledgements	ii
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Literature Review	8
2.1 Early History	8
2.2 Experimental and Analytical Models - Steady-State	9
2.3 Transient Analysis	11
2.4 Inner Boundary Condition	26
2.5 Unconfined Flow in the Petroleum Industry	33
2.6 Numerical Solutions	35
2.7 Contribution of the Present Work	37
3 Mathematical Model	38
3.1 Basic Assumptions	38
3.2 Partial Differential Equations and Boundary Values	39
3.3 Dimensionless Parameters	41
4 Finite Difference Model	43
4.1 Grid Distribution	46
4.2 Balance of Unknowns and Equations	49
4.3 Saturated Region	49
4.3.1 Discretization of the Saturated Region Equations	49
4.3.2 Saturated Region Inner Boundary Equations	52
4.3.3 Saturated Region Outer Boundary Blocks	55
4.4 Free Surface Blocks	57
4.4.1 Free Surface Layer Equations	57
4.4.2 Free Surface Inner Boundary Equation	61

4.4.3	Free Surface Outer Boundary Blocks	63
4.5	Lower Boundary Blocks	65
4.5.1	Saturated Region Lower Boundary Equations	66
4.5.2	Lower Inner Boundary Equation	67
4.5.3	Lower Outer Boundary Equation	68
4.6	Second Layer Region Blocks	69
4.6.1	Second Layer Equations	70
4.6.2	Second Layer Inner Boundary Equation	71
4.6.3	Second Layer Outer Boundary Equation	72
4.7	Capillary Fringe	74
4.7.1	Description of the Physical Mechanism of the Capillary Flow	74
4.7.2	Mathematical Representation of the Capillary Layer Boundary	76
4.7.3	Mathematical Representation of the Phreatic Layer	78
4.7.4	Inner Boundary Phreatic Layer Equation	79
4.7.5	Outer Boundary Phreatic Layer Equation	80
4.8	Sandface and Wellbore Conditions	81
4.9	Computer Model	85
5	Verification of the Numerical Model	87
5.1	Verification with Wyckoff, Botset and Muskat Experiments	87
5.2	Verification with the Cooley and Neuman Solutions	90
5.3	Verification with Hall Sandbox Experiments	95
6	Sensitivity	98
6.1	Grid Block Dimensioning	98
6.2	Time Weighting and Time Step Length	101
7	Discussion	107
7.1	Transient Wellbore Pressure Analysis	108
7.1.1	Drawdown	109
7.1.2	Buildup	116
7.2	Permeability	120
7.3	The Seepage Surface	122
7.4	Skin Effect	125
7.5	Anisotropy	126
7.6	Production Flow Rate	128
7.7	Original Liquid Height and Transmissivity	130
7.8	Considerations About Well Test Projects	133
8	Conclusions and Recommendations	141
8.1	Conclusions	141
8.2	Recommendations	142
Nomenclature		144

References	147
A Inner Boundary Condition	153
A.1 Partial Derivatives of the Sandface Potential	153
A.2 Liquid level in the Wellbore from Material Balance Equation	156
A.2.1 Quadratic Equation	156
A.2.2 Iterative Solution for the Wellbore Liquid Level	159
A.3 Material Balance Equation Partial Derivatives	162
B Groundwater and Petroleum Engineering Units	166
C Simplified Approaches	171
C.1 Wellbore Effects in the Theis Solution	172
C.2 Wellbore Effects in the p^2 Solution	175
D Verification of the Free Surface Boundary Condition Relationship with the Average Vertical Velocity	181
E SLM Computer Program	182
E.1 Structure of the Program	182
E.2 Description of some Characteristics	183
E.2.1 Radial Block Distribution	183
E.2.2 Axial Permeability	183
E.2.3 Sandface Block Transmissivity Control	184
E.2.4 Skin Effect	184
E.3 Input-Data File	185
E.4 Example of the Program Output	188
E.5 Computer Program Code Listing	193

List of Tables

5.1	Determination of the average constant "A" from the experimental Run No. 11	88
5.2	Heads (cm) at sand base from <i>Wyckoff et al.</i> experimental results compared with the <i>Stream Layer Model</i>	91
5.3	Data from <i>Cooley</i> (1971)	91
5.4	<i>SLM</i> results compared to the <i>Hall</i> series A experiments.	96
5.5	Comparison of potential heads between the <i>Hall</i> sandbox experiments and <i>SLM</i> ($M = 12, N = 24$), series A, Test No.6	97
6.1	Basic input data for the sensitivity runs	106
7.1	Basic data used as input to the <i>SLM</i>	108

List of Figures

1.1	Unconfined free surface gravity drainage reservoir producing to a single well - <i>Muskat</i> (1937)	2
1.2	Typical type curve for unconfined flow from the groundwater literature	3
2.1	Essential dimensions of well model - from <i>Hall</i> (1955)	11
2.2	Time for flow along the streamlines - <i>Hall</i> (1955)	12
2.3	From <i>Boulton</i> (1963) paper: delayed yield type curves.	17
2.4	Cumulative Production for Small Values of \bar{h}_i/r_e . - from <i>Matthews</i> and <i>Lefkovits</i> (1956).	33
3.1	Representation of a cross section of an unconfined reservoir	39
4.1	The <i>Stream Layer Model</i> grid	44
4.2	Geometric corrections based in the slope of the flow axis.	45
4.3	Matrix representation of a 5x4 cross sectional grid	46
4.4	Representation of a block containing the free surface. Convention of flow and coordinate directions.	58
4.5	Representation of the capillary fringe layer.	75
4.6	Potential and pressure profiles at a radial position r , under static conditions.	76
5.1	Liquid levels and heads in the sand model by <i>Wyckoff et al.</i> compared to the <i>SLM</i> results of runs 18 and 19.	90
5.2	Stream Layer Model compared to <i>Cooley, Theis, Ramey et al.</i> and <i>Neuman</i> . Observation well at 44.9 ft. Log-log plot.	93
5.3	Stream Layer Model compared to <i>Cooley, Theis, Ramey et al.</i> and <i>Neuman</i> . Observation well at 44.9 ft. Semilog plot.	93
5.4	Stream Layer Model compared to <i>Cooley, Theis, Ramey et al.</i> and <i>Neuman</i> . Observation well at 101.1 ft. Log-log plot.	94
5.5	Stream Layer Model compared to <i>Cooley, Theis, Ramey et al.</i> and <i>Neuman</i> . Observation well at 101.1 ft. Semilog plot.	94
6.1	Radial mesh variation, 100 vertical grid blocks	99
6.2	Vertical mesh variation, 60 radial grid blocks	100
6.3	Nonlinear flux variation in a block boundary during a time-step	101
6.4	Effects of time interpolation parameter in the <i>SLM</i> results	102

6.5	Effects of time-step size in the wellbore potential drawdown	103
7.1	Wellbore flowing pressure log-log plot	109
7.2	Wellbore flowing pressure semilog graph	110
7.3	Wellbore flowing pressure-squared semilog plot	112
7.4	Sandface flow rate distribution	113
7.5	Seepage height vs. cumulative time for drawdown period	114
7.6	Dimensionless radial head profile during drawdown period	115
7.7	Iso-potential map	116
7.8	Wellbore buildup pressure and derivative - log-log plot	117
7.9	Wellbore buildup pressure - semilog plot	118
7.10	Wellbore buildup pressure - Horner plot	118
7.11	Seepage height vs. cumulative time for buildup period	119
7.12	Dimensionless radial head profiles during buildup period	121
7.13	Wellbore pressure drawdown of intermediate to low permeability reservoirs under gravity drainage	122
7.14	Wellbore pressure drawdown of intermediate to high permeability reservoirs under gravity drainage	123
7.15	Semilog plot of the seepage height: low permeability range	124
7.16	Semilog plot of the seepage height: high permeability range	124
7.17	Pressure drawdown vs. time for different skin parameters.	126
7.18	Radial head potential profile of a damaged well	127
7.19	Results from <i>SLM</i> for a 120 mD horizontal permeability and different vertical permeabilities	128
7.20	<i>SLM</i> results for different production flow rates	129
7.21	High flow rate solution compared with analytical approaches	130
7.22	Comparison between <i>SLM</i> runs of two tests in formations with the same q/k ratio	131
7.23	<i>SLM</i> results for different original static liquid level	132
7.24	Potential drawdown results for the same transmissivity and different permeabilities	132
7.25	Potential squared drawdown results for the same transmissivity and different permeabilities	133
7.26	Combined 24-hour flowing and buildup test for a set of different permeabilities. Skin = 0.	135
7.27	Combined 24-hour flowing and buildup test for a set of different permeabilities. Skin = 4.	136
7.28	Combined 24-hour flowing and buildup test for a set of different permeabilities. Skin = 8.	137
7.29	Combined 24-hour flowing and buildup test for a set of permeabilities and skin parameters.	138
7.30	Time in which the liquid level in the well reaches the free surface level in the sandface during buildup.	139

7.31 Buildup pressure drop behaviors for different production times	139
7.32 Liquid levels in the wellbore and at the sandface during drawdown and buildup periods	140
7.33 Potential drop in the well after a 24-hour production period.	140
C.1 <i>PDE</i> and analytical <i>Theis</i> (1935) solution. Semilog plot.	178
C.2 <i>PDE</i> and the analytical <i>Theis</i> (1935) solution. Log-log plot.	178
C.3 <i>PDE</i> and analytical <i>Theis</i> (1935) solution. Buildup pressure semilog plot. .	179
C.4 <i>FLIGRAM</i> and <i>Ramey et al.</i> approaches for the drawdown period, semilog plot.	179
C.5 <i>FLIGRAM</i> and <i>Ramey et al.</i> approaches for the drawdown period, log-log graph.	180
C.6 <i>FLIGRAM</i> and <i>Ramey et al.</i> approaches for the drawdown period, p-squared semilog graph.	180

1. Introduction

The main objective of this study is the development of a numerical simulator to study the transient drawdown and buildup behavior of a liquid in a gravity drainage well with a moving free surface boundary. The purpose of this work is to provide a tool to help understanding the flow mechanism in the vicinity of the wellbore where the known methods of interpretation fail to determine the transient pressure behavior or to establish physical grounds for creating a method of interpretation of pumping gravity well tests.

In the last few years, much attention has been dedicated to depleted petroleum reservoirs in the oil industry all over the world. The interdependency of oil and gas reserves, price, demand, and politics is a major factor affecting interest in developing alternative sources of energy and enhanced oil recovery from abandoned fields, heavy oil reservoirs, tar sands, bitumen shales, etc.

Among the potential targets for enhanced oil recovery projects are many depleted oil reservoirs, some with such a low pressure that the gas cap is at atmospheric pressure. This is the case when gas solubility is low, or for water flooded shallow reservoirs which have produced much of the oil, water and associated gas.

A common characteristic of such a reservoir is the lack of detailed information about the formation properties necessary to plan an enhanced oil recovery project. Unfortunately, the present petroleum well testing techniques do not consider the case of a well in a reservoir with a phreatic liquid table. In hydrology, a phreatic aquifer is defined as an unconfined reservoir with an air-water contact at atmospheric pressure. Pumping a well that penetrates a formation with a phreatic gas-liquid contact develops a downward movement of the free surface, under a gravity flow mechanism. Often, no other source of energy is present for driving the liquid into the wellbore. An exception is capillary forces acting in the capillary fringe above the phreatic line. A schematic view of the gravity well problem is presented in the Fig. 1.1, representing a cross section of a fully penetrating open hole well in an unconfined sandstone reservoir.

Until recently, most theory found in the petroleum literature about well test analysis has concerned confined flow of compressed fluid. However, the groundwater literature contains many studies where phreatic aquifers are a common subject. On the other hand, even hydrologists have not focused on pumping wells where the presence of a seepage face along the sandface adds further difficulties to a moving boundary problem. Since unconfined

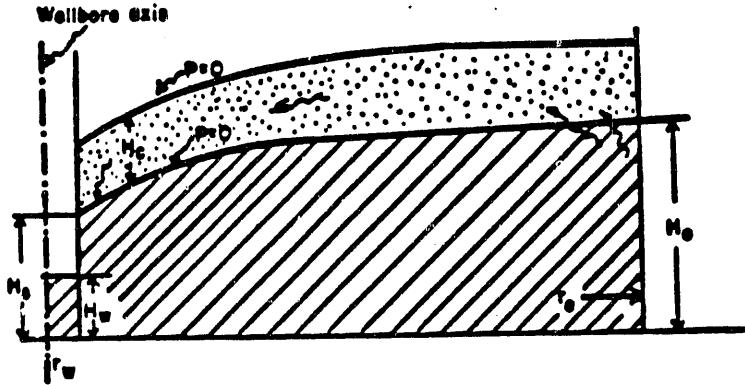


Figure 1.1: Unconfined free surface gravity drainage reservoir producing to a single well - Muskat (1937)

aquifers are often shallow and the wells are easy and cheap to drill, methods of interpretation often concerned interference tests.

The theory associated with the gravity flow mechanism with a variable free surface started in the last century with the *Dupuit* (1862) approach, generalized by *Forchheimer* (1886) in the so-called Dupuit-Forchheimer partial differential equation, which for radial incompressible flow is:

$$\frac{\partial^2 h^2}{\partial r^2} + \frac{1}{r} \frac{\partial h^2}{\partial r} = 0 \quad (1.1)$$

Eq. 1.1 concerns saturated flow in an incompressible radial system. Many assumptions were made such as formation isotropy and neglection of vertical flow gradients. The Dupuit-Forchheimer approach has become a standard reference for improved theories and interpretation methods.

An early rigorous attempt to solve the steady-state gravity problem under several different flow geometries using a hodographic transformation was made by *Wyckoff et al.* (1932) and *Muskat* (1937). A hodographic transformation is a representation of the dynamical system by coordinates that are the velocity components. This rather difficult method, although supporting the Dupuit-Forchheimer theory, fails in producing the shape of the free surface, as observed in Chapter 2 of this study.

The classic transient solution for compressed liquid flow (*Theis*, 1935) is often used

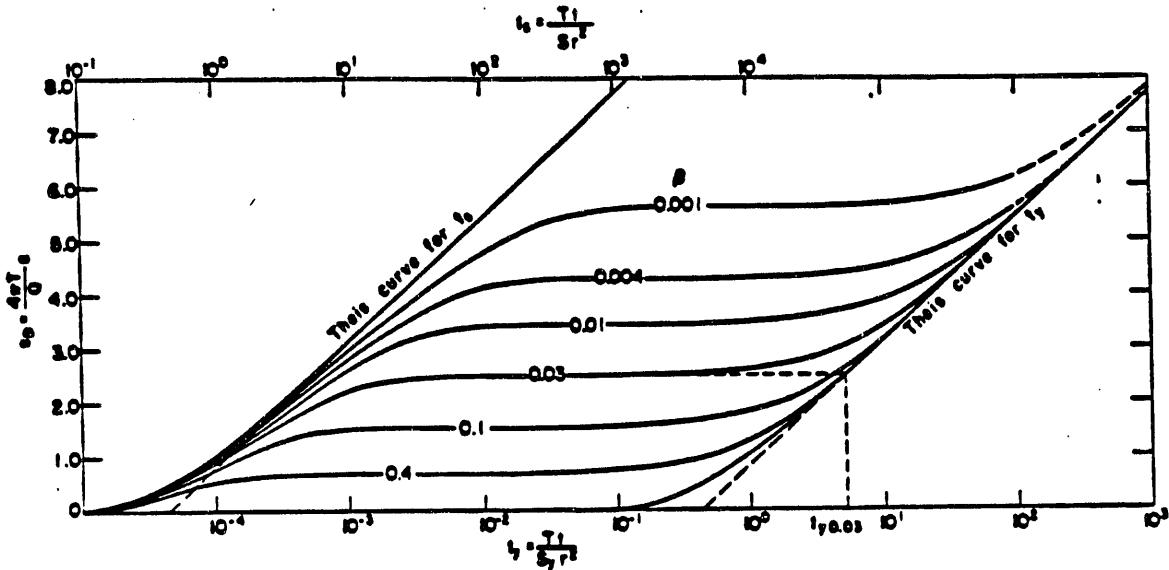


Figure 1.2: Typical type curve for unconfined flow from the groundwater literature: dimensionless potential drawdown vs. dimensionless time. - Neuman (1975-a).

today to represent the late time drawdown behavior in observation wells by well known groundwater authors like *Boulton*, *Neuman*, *Streltsova*, among many others. Although the hydrology literature is rich concerning this subject, there is a great deal of confusion about the physical principles governing unconfined flow - see Chapter 2.

Regarding transient drawdown in observation wells near a well producing at a constant pumping rate, there are three different flow regimes characterized by the S-shaped curve illustrated by Fig. 1.2. That figure shows set of semilog type curves as a function of a β parameter, which is related to the inverse of the radial distance. Observation wells located far from the producing well present a characteristic time-drawdown curve reflecting three stages of flow in the reservoir. The first stage is attributed to a pure radial flow by fluid expansion and compactation of the aquifer before the free surface starts a downward movement. The *Theis* solution applies, according to many hydrologists. This period corresponds to that dimensionless time defined as t , in Fig. 1.2. In this case, s stands for *specific yield*. As the free surface responds to the well pumping, an intermediate stage is marked by a reduction in the slope of the head drawdown, approaching zero in some circumstances. This is the period of desaturation of the cone around the well. The last stage, characterized by a dimensionless time t_y , y standing for the *delayed yield*, would again follow radial flow behavior because of stabilization of the free surface changes. The analytical methods presented by *Streltsova* (1972-a, 1972-b, 1973) and *Neuman* (1972, 1975) considered elastic and formation storage properties lumped into the specific yield parameter. To obtain the analytical solution for the boundary problem (see Chapter 2), Neuman imposed such

conditions as the head drawdown must be much smaller than the initial head, the well was a line-source well, wellbore storage and skin effects were neglected, and the seepage face was neglected. As a result, analytical methods were not applicable to a pumping well or its vicinity, and were insufficient to analyze the transient pressure behavior in the produced wellbore.

Using the *Dupuit-Forchheimer* approach, *Ramey et. al* (1989) derived a partial differential equation for incompressible radial flow into a single well located in an infinite reservoir. In terms of pressure, instead of liquid head, the partial derivative equation takes the form of Eq. 1.2, in Darcy units:

$$\frac{\partial^2 p^2}{\partial r^2} + \frac{1}{r} \frac{\partial p^2}{\partial r} = \frac{\phi \mu (p^{-1})}{k} \frac{\partial p^2}{\partial t}. \quad (1.2)$$

By analogy with the flow of ideal gas, the *Aronofsky and Jenkins* (1953) correlations were applied to obtain the approximate wellbore transient pressure solution, as follows in petroleum field units:

$$p_o^2 - p_w^2 = 325.2 \frac{q B \mu}{k(h/p)} \left\{ \log \left(\frac{0.000264 k t}{\phi \mu (p_o^{-1}) r_w^2} \right) + 0.351 + \frac{s}{1.151} \right\} \quad (1.3)$$

This appears to be the first approach for gravity drainage well test analysis found in the petroleum engineering literature since the classic studies of *Wyckoff et al* (1932) and *Muskat* (1937).

From the many concepts and ideas related to unconfined flow into a well, it is possible to consider the physical flow mechanism. First, start with the rest condition of an unconfined, single-phase reservoir, like that represented in Fig. 1.1. Above the phreatic surface, there is a partly-saturated region containing liquid called the capillary fringe where the pressures are negative. As a consequence, the static potentials are constant and equal to the potentials anywhere in the saturated region below the phreatic table, defined throughout this report by $\Phi = p + \gamma h$. The degree of liquid saturation in the capillary region and the elevation of liquid above the free surface is dependent on the surface tension between the liquid and the porous medium and the gas and the pore throats related to the shape and grain size distributions.

When the well is pumped at a constant rate in an infinite, unconfined reservoir, the liquid level in the wellbore drops at a velocity that depends on the well storage capacity. Due to the viscous forces in the formation, the potential drop in the well is not immediately

transmitted to the reservoir. This causes a discontinuity of potentials at the sandface. With increasing time, the potential drops in the formation causing the liquid to start moving into the well at increased rates as the potential differentials become large, and the pressure drop is transmitted radially and vertically through the formation. During this period, called the wellbore storage period, the withdrawal from the well is greater than the flow from the formation.

Vertically along the sandface in the well, there are two different regions of flow: an open face, where the liquid falls, and the liquid filled well region below the liquid level in the hole. The seepage length is the difference between the liquid levels at the sandface and in the well. Thus, at the sandface wall, the potentials vary from a maximum value at the intersection point of the phreatic free surface down to the liquid level in the well, and remain constant from there down to the bottom of the well.

Following an initial period of flow, the potential drawdown in the pumping well decelerates, due to an increasing contribution of flow from the formation. The withdrawal in the neighborhood of the well develops potential gradients driving the streamlines to bend gradually, with decreasing slopes from the free surface down to the sand base. The produced mass of liquid, considering a low compressibility system, leaves a desaturated region behind the free surface. Neglecting for a while the capillary region, which is discussed in Chapter 4.7, the desaturated region is a two-phase flow region where the liquid saturations successively drop to an irreducible value. Due to the low velocity of vertical displacements of the free surface, the resistance created to the gas entry into the porous spaces left by the liquid under the gravity flow are of minor importance to create a delay in the process, according to some authors, even around the wellbore. Also, the liquid retained by capillary forces in the desaturated zone, though known to exist in small amounts, is not rigorously considered in the present study.

After the wellbore storage effects cease and the desaturation cone starts developing in the formation, the pressure in the well stabilizes at a certain level as a result of equilibrium between the rates of pumping and the withdrawal from the surroundings. At this time, the seepage face tends to shrink. The cone of depression develops radially from the wellbore. As the depression of the free surface reaches remote regions, a small free surface displacement corresponds to a large volume of liquid. In other words, the storage capacity becomes large with increased radius.

Since an external boundary is not reached and the free surface presents a fairly stable shape, the potential changes reflect a confined system behavior corresponding to a third cycle of observed pressures in the well. During this period, one expects a slow, uniform downward displacement of the free surface, and the seepage face changes at a much lower rate.

The transmissivity has an important effect on the potentials and free surface profiles. The velocity along stream lines at any radial location, a consequence of the transmissivity and withdraw rates, is of interest. In other words, if we track two single stream flow lines from a remote region to the sandface, vertically spaced by a certain difference in height, we observe a reduction in the distance between them by the stream-line merging process. Since the material balance condition for two consecutive stream flow lines to converge is to double their velocities (case of incompressible flow), the viscous forces are the most responsible for the shape of the free surface, and also for the seepage face length. By itself, the viscous flow theory can explain the formation of the seepage face without invoking surface tension effects. Those capillary end effects, although present in the very top of the seepage face are of minor importance, according to some experimental sandbox models (*Hall*, 1955). *Wyckoff et al.* (1932) explained the influence of the resistance forces to the vertical flow in the formation neglecting capillary forces by expressing the potential in the free surface as a function of the average vertical velocity at some radial position:

$$Z_s = Z + \frac{\bar{v} Z_s}{k \gamma g}, \quad (1.4)$$

where:

- Z_s - sandface vertical distance from the sand bottom
- Z - head of liquid at the bottom
- \bar{v} - average vertical velocity of the liquid
- k - equivalent to the transmissivity
- γg - liquid gradient

In Appendix D, Eq. 1.4 is verified against $\Phi = p + \gamma h$, verifying the free surface boundary condition found in the literature.

The buildup process in gravity wells has been overlooked in most studies. Some references in the groundwater literature takes buildup as a reversible process similar to drawdown, but there has not been a careful investigation of the physical behavior during this period. *Ramey et al.* (1989), analyzing a field case, concluded that the pressure buildup is “essentially different” from the prior drawdown process, leading to a rapid establishment of a semilog straight line. According to *Ramey et al.*, radial flow occurs below the free surface during the buildup, and the free surface is displaced up by radial flow into the cone of depression. After pumping stops, the liquid flows into the wellbore, and starts refilling the well. Section 7.1.2 presents considerations of the buildup analysis.

The first attempt to study the gravity well problem was to find a two-dimensional numerical model that could answer most of the questions related to the real physics of the

mechanism. A moving boundary simulator was developed and named *Stream Layer Model*. A *stream layer* is a set of stream flow lines in a vertical section of an axial flow into the well. The next chapter presents a literature survey for gravity drainage well test analysis.

2. Literature Review

The mechanism of gravity drainage flow in a porous medium has not been completely studied in either groundwater research or the oil industry. In this chapter, a review of the main works in gravity drainage well test analysis is presented. Also, studies of two-dimensional well model simulators are reviewed both in groundwater and petroleum engineering.

Pumping tests are common in hydrologic evaluations, and gravity drainage evaluation methods have been necessary due to the fact that many shallow aquifers (phreatic) are free surface systems. Hence, the groundwater literature was the main source of information for the present investigation.

In most studies of unconfined radial flow, water is the only saturating fluid in the porous medium, and the presence of other phases is not considered, except in a few studies dealing with a desaturated region and/or a capillary fringe between the water table and the air-filled pores above it. Generally, most models consisted of a well located in the center of an infinite reservoir where the formation properties were sometimes considered anisotropic, and the presence of a damaged region was neglected. Wellbore effects were some of the major difficulties encountered in this study and only recent studies of wellbore effects have been made by some authors.

A small difficulty is relating petroleum engineering terms to the reservoir properties and parameters defined by hydrologists. A set of relationships and unit conversions between the definitions employed in both branches of study is given in Appendix B.

2.1 Early History

Apparently, the first important contribution to the gravity flow problem was the *Dupuit* (1863) analytical solution which assumed a radial, steady-state parabolic distribution of head with the distance from the wellbore. A simplification of the mathematical steady-state problem was made by assuming that for a small slope of the water table, which is also a streamline, the Darcy's Law velocity vector is proportional to the tangent of the angle the streamline makes with the horizontal, rather than the sin of the angle. Thus,

instead of:

$$v_s = -\frac{k}{\mu} \frac{\partial \Phi}{\partial s} = -\frac{k}{\mu} \frac{\partial z}{\partial s} = -\frac{k}{\mu} \sin \theta, \quad (2.1)$$

$\sin \theta$ was replaced by $\tan \theta = \frac{dh}{dx}$, where $h = h(x)$ is the vertical coordinate of the water table position. In the above, K is the mobility $\frac{k}{\mu}$. This assumption permitted a derivation of the steady-state flow equation by *Dupuit*, which for radial flow is:

$$Q_w = 2\pi \frac{K}{\mu} r h \frac{\partial h}{\partial r} = \frac{\pi k (h_e^2 - h_w^2)}{\mu \log \left(\frac{r_e}{r_w} \right)} \quad (2.2)$$

Forchheimer (1886) applied the *Dupuit* assumption to the continuity equation and produced the partial differential equation for radial flow:

$$\frac{\partial^2 h^2}{\partial r^2} + \frac{1}{r} \frac{\partial h^2}{\partial r} = 0 \quad (2.3)$$

Equation 2.3 is the well known *Dupuit-Forchheimer* partial differential equation for steady-state gravitational radial flow.

2.2 Experimental and Analytical Models - Steady-State

Wyckoff et al. (1932) investigated gravity flow using an experimental 15° sector sand box to simulate the radial flow of a single liquid (water) saturating the reservoir. Their results provided important informations about the free surface and streamline behavior along the reservoir and around the wellbore. The radial sector was laterally bounded by glass walls which permitted visual inspection of the liquid level in the porous medium, and streamlines traced by ink injected at points at the outer boundary. Eighteen complete steady-state experimental runs were presented, varying initial heads and flow rates in order to compare the results with the *Dupuit* model and other models. The capillary fringe effects were important at the laboratory scale, and most of the the deviations from the *Dupuit* model were attributed to the capillary fringe. The tables summarizing the results of each experiment presented the liquid heads taken at the sand bottom, and the liquid level (free surface position) profiles were graphed for runs No. 18 and 19 in the paper. There were no direct seepage face length measurements, but visual inspections and the graphs presented in the paper indicated the seepage face existence.

Muskat (1937) dedicated an entire chapter to the analysis of gravity flow systems. His study of steady state flow for different configurations (*Muskat*, 1935, 1937) of porous medium included physical experiments such as sandbox and electrical models (by other investigators). He pointed out the weakness of the *Dupuit* assumptions, and raised the concept of a seepage face along the downstream boundary that introduces a complication to the problem once the water table causes a discontinuity in the wellbore. Since the *Dupuit* equation depends only on radial distance, this equation fails to determine the head distribution along the vertical coordinate in the vicinity of the wellbore. Far from the well, horizontal flow dominates, and the *Dupuit* results are reasonable. *Muskat* concluded the *Dupuit* equation, although based on incorrect assumptions, gives fortuitously accurate results in the calculation of the discharge of a well when the head values are taken at the base of the formation.

Muskat (1935, 1937) solved the steady gravity flow problem analytically through two methods: an exact solution using the hodograph method, and an approximate solution by Fourier series expansion of the boundary conditions. These solutions are not simple even for an isotropic medium. The method of hodographs, though powerful, is difficult to apply. Analytical and experimental results were compared in several figures in the *Muskat* (1937) book.

Babbit and Caldwell (1948) used one electrical and two sandbox models (small and large scales) to corroborate the observations of *Wyckoff et al.* (1932) and *Muskat* (1937). The effect of the capillary fringe was omitted in this work which generated an empirical equation to locate the free surface as a function of the distance from the wellbore axis:

$$Q = \frac{\pi k h_e (h_e - h_x)}{2.3 C_x \log r_e / 0.1 h_e} \quad (2.4)$$

where h_x and h_e are respectively the free surface positions at distances r_x and r_e from the wellbore center, and C_x is an empirical constant obtained as a function of the dimensionless radial distance.

The most valuable experimental work is a major reference in the groundwater literature by *Hall* (1955). *Hall* presented four series of sandbox experiments analyzing capillary effects, the hysteresis due to air invading the desaturating cone, and changes in the surface tension in the rock-fluid system. Figure 2.1 from the *Hall* paper shows a 15° sector and the essential sandbox model dimensions. Series A of tests performed by *Hall* presented eight steady-state steps of successive flow rates starting from a completely water saturated system with an average capillary fringe height of 3.5 inches. It is remarkable to observe the behavior of the free surface as the flow rate increases. From run A-4 to A-8, the free surface position at the sandface varied only -3.6%, while the wellbore liquid level dropped from half

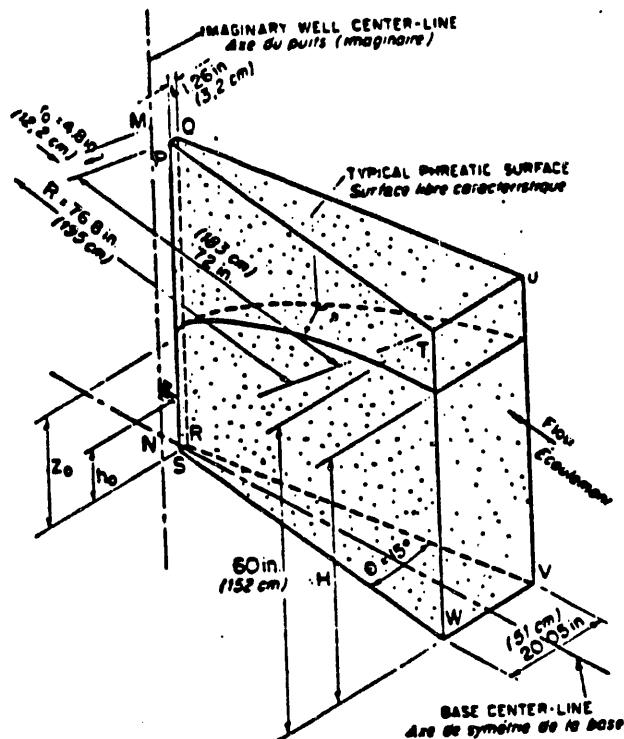
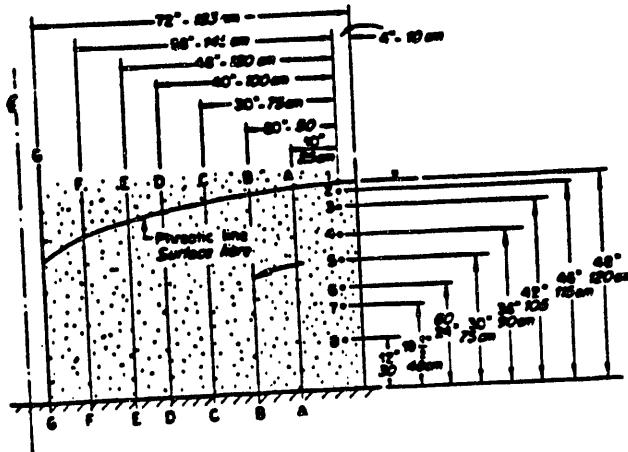


Figure 2.1: Essential dimensions of well model - from *Hall* (1955).

of the initial height to zero, and the production flow rate increased more than 32%. This result is a difficult mathematical condition to be represented by only a few variables as in the *Dupuit* formula. The streamline traces were qualitatively and quantitatively studied, demonstrating the importance of the vertical position of a streamline related to the time necessary to flow from the outer source to the inner boundary. See Fig. 2.2. The uppermost streamline requires twice as long to reach the wellbore as the lowermost streamline. There are two possible reasons for this phenomenon. First, the uppermost inner boundary potentials are always greater than the lowermost potentials. See Ch. 3. Second, the uppermost streamlines are much longer than the lowermost streamlines.

2.3 Transient Analysis

An early important study of compressible transient flow was presented by *Theis* (1935). Previous groundwater hydraulic studies had been based on steady flow conditions. *Theis* applied the theory of heat conduction in solids using the line-source solution adapted to the flow of fluids in porous medium. In the paper, the limitations of the use of the exponential integral, $E_i(-x)$, in analyzing unconfined flow were considered. Early time



POINT OF INJECTION Point d'injection	TIME, IN MINUTES, REQUIRED TO REACH SECTION Temps, en minutes, nécessaire pour atteindre la section							REMARKS Observations
	4-A	B-B	C-C	D-D	E-E	F-F	G-G	
1	2.8	6.0	9.0	11.3	13.5	15.0	16.0	Capillary layer.
2	2.2	4.5	6.7	8.5	10.1	11.2	12.0	
3	2.0	4.2	6.2	8.0	9.3	10.3	11.0	
4	2.0	4.0	6.2	7.8	9.0	10.0	10.5	
5	2.0	3.9	5.8	7.3	8.4	9.3	9.8	
6	1.8	3.8	5.5	7.0	8.0	8.8	9.0	
7	1.8	3.6	5.2	6.8	7.7	8.4	9.0	
8	1.8	3.5	5.0	6.6	7.6	8.0	8.5	

Figure 2.2: Time for flow along the streamlines - Hall (1955).

effects due to a finite radius wellbore and gravitational flow around the wellbore were considered. *Theis* presented the technique later used by *Horner* (1951) to analyze the pressure-time behavior in well recovery (buildup) analysis by plotting pressure-drawdown versus the dimensionless time relation $(t_p + \theta)/\theta$.

A paper by *Walton* (1960), written in two parts, considered pumping test analysis for both artesian and unconfined aquifers. The author pointed out that the true behavior of water table models deviates from the line-source solution, mainly at early times. Three different segments of the time-drawdown curve were recognized: a very early one in which the well starts producing under decompression of the water and formation (as short as a few seconds or minutes) followed by a gravity drainage period when vertical flow near the wellbore controls and the free surface changes rapidly, and finally a late period where the rate of change of the water table is no longer important. In the first and last periods, the so-called nonequilibrium formula (*Theis* solution or its logarithmic approximation) could be applied, because the flow is dominantly horizontal. However, when and where gravity drainage is the most important mechanism, solutions like *Boulton* (1954-b) should be used.

In 1954, *Boulton* (1954-a) introduced a new approach to represent unconfined transient flow behavior. Using *Darcy's* Law and the continuity equation for flow of incompressible liquids, the following partial differential equation in terms of the liquid potential $\Phi = p/\gamma_w + z$ was posed in the case of isotropic aquifers:

$$\nabla^2(\Phi) = \frac{\partial^2 \Phi}{\partial r^2} + \frac{1}{r} \frac{\partial \Phi}{\partial r} + \frac{\partial \Phi^2}{\partial z^2} = 0 \quad (2.5)$$

where r is the radial distance, and z is the vertical coordinate.

The water table boundary condition was expressed by:

$$\Phi(r, z, t) - z = 0 \quad (2.6)$$

The differentiation following the motion of a particle located on the free surface streamline was defined by the operator:

$$\frac{D}{Dt}(\Phi - z) = 0 \quad (2.7)$$

In Eq. 2.7:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{v_r}{S} \frac{\partial}{\partial r} + \frac{v_z}{S} \frac{\partial}{\partial z} \quad (2.8)$$

Applying the differential operator as defined, the following nonlinear equation was obtained:

$$\frac{\partial \Phi}{\partial t} = \left\{ \left(\frac{\partial \Phi}{\partial r} \right)^2 + \left(\frac{\partial \Phi}{\partial z} \right)^2 - \frac{\partial \Phi}{\partial z} \right\} \quad (2.9)$$

Equation 2.9 was simplified by the assumption that the Φ gradients are small, and when $z = h$:

$$\frac{\partial \Phi}{\partial t} + \frac{k}{S} \frac{\partial \Phi}{\partial z} = 0 \quad (2.10)$$

The inner boundary condition considered a line sink at a constant pumping rate per unit length:

$$Q = 2\pi k h_e r \frac{\partial \Phi}{\partial r}; \quad r \rightarrow 0; \quad 0 \leq z \leq h_e \quad (2.11)$$

A general solution of Eqs. 2.6 to 2.11 and the boundary conditions was reached after simplifying assumptions:

$$h_e - \Phi = \frac{Q}{2\pi k h_e} \int_0^\infty \frac{J_o(\beta r)}{\beta} \left\{ 1 - \frac{\cosh(\beta z)}{\cosh(\beta h_e)} \exp\left(-\frac{k}{S} t \beta \tanh(\beta h_e)\right) \right\} d\beta \quad (2.12)$$

Further simplifications were made and a definite integral denoted by $V(r, t)$ was introduced:

$$V(\rho, \tau) = \int_0^\infty \frac{J_o(\lambda \rho)}{\lambda} \{1 - \exp[-\lambda \tau \tanh(\lambda)]\} d\lambda \quad (2.13)$$

$$s = \frac{Q}{2\pi T} V(\rho, \tau) \quad (2.14)$$

where ρ and τ are dimensionless radius and time, respectively:

$$\rho = \frac{r}{h_e} \quad (2.15)$$

and:

$$\tau = \frac{k t}{S h_e} \quad (2.16)$$

Values of the integral in Eq. 2.13 were tabulated, as well as corrections for deviations due to the several simplifications. *Boulton* (1954-a) also handled modifications to consider anisotropy and head recovery (buildup), and a correction for a finite wellbore radius.

The second *Boulton* paper (1954-b) contains a solution that became an important reference in the groundwater literature during the subsequent 15 to 20 years, because it introduced a new theory for unsteady radial flow in an unconfined isotropic aquifer. Two different cases were considered, but the one that fitted the gravity drainage model best assumed a delayed yield from storage caused by an idealized bed of fine-grained material overlaying the aquifer with the stabilized water table within this section of low permeability. The rate of delayed yield was empirically represented by:

$$\delta s \propto S' e^{-\alpha(t-\tau)} \quad (2.17)$$

where α is an empirical constant, and S' is the delayed yield per unit area, per unit drawdown. Neglecting the vertical components of flow, the final form of the partial differential equation considering delayed yield was:

$$\frac{\partial^2 s}{\partial r^2} + \frac{1}{r} \frac{\partial s}{\partial r} = \frac{S}{T} \frac{\partial s}{\partial t} + \frac{\alpha S'}{T} \int_0^t \frac{\partial s}{\partial t} e^{-\alpha(t-\tau)} dt \quad (2.18)$$

The following solution for Eq. 2.18 was obtained properly using the Laplace transform technique:

$$s = \frac{Q}{4\pi k h_e} \left\{ \int_0^\alpha (1 - e^{-ut}) J_0 \left[r \sqrt{\frac{u}{a} \frac{\eta\alpha - u}{\alpha - u}} \right] \frac{du}{u} + \int_{\eta\alpha}^\infty (1 - e^{-ut}) J_0 \left[r \sqrt{\frac{u}{a} \frac{\eta\alpha - u}{\alpha - u}} \right] \frac{du}{u} \right\} \quad (2.19)$$

where:

$$\eta = 1 + \frac{a c}{\alpha} = \frac{S + S'}{S}, \quad a = \frac{T}{S}, \quad \text{and} \quad c = \frac{\alpha S'}{T} \quad (2.20)$$

Some approximations to the solutions were prepared by assuming limiting values of η . Tables containing numerical values for different approximations of Eq. 2.19 were provided.

The explanation used to justify the *Boulton* model was weak and had little physical connection with the pure gravity flow mechanism. However, the mathematical formulation was shown to match many sets of field data, and later *Boulton* (1963) presented the same model, and a group of *delayed yield* type curves generated by a computer program - see Fig. 2.3. The early and late time behaviors were analyzed, and two different sets of curves were plotted on the same graph at two different dimensionless time scales. The shapes of these curves, according to the author, show different flow patterns. Under the consideration of the initial and internal boundary conditions at the very early time of pumping, the time-drawdown curve follows the *Theis* solution for an artesian aquifer. Next, as the yield from the water table becomes effective, the drawdown behavior deviates from the line-source solution due to the contribution by the water table fall, and the curves bend to the right (Type A curves), trending to an asymptotic behavior. After a time, the approximations of the integral show flat curves which start to bend up (type B curves), merging into the *Theis* solution. The dimensionless time at which the departure from *Theis* solution drawdown and the calculated type B curve from the *Boulton* model equals 0.02 generates a new parameter called the *delayed index* ($\frac{1}{\alpha}$). This delayed index is a characteristic of the formation and is physically related to the gravity vertical flow around the wellbore. As in the previous work (*Boulton*, 1954-b), only isotropic formations were considered.

Prickett (1965) applied the *Boulton* (1963) solution to eighteen pumping tests and found good agreement with the theory. The relationship obtained in the tests between the delayed index and the grain size of sand demonstrated a physical correlation. Also, an interesting set of graphs shows the individual variation of different parameters. Theoretical time-drawdown curves indicated the effects of changing only one variable of the basic equation (such as $1/\alpha$, S , Sy , r , and T), while all other coefficients remained fixed.

To consider the vertical flow around a well, *Boulton* (1970) presented, without derivation, an equation based on the delayed yield (*Boulton*, 1954-b and 1963) and the vertical velocity-component of the flow (*Boulton*, 1954-a). However, this equation was not applied in the 1970 study. Instead, some considerations were suggested in order to have "the most reliable method of analyzing pumping test data for an unconfined aquifer", using previous theories:

- (a) the use of matching distance-drawdown by type-curves based on a constant coefficient of Storage (such as the *Boulton* V-function, *Boulton* 1954-a), and
- (b) the application of the delayed yield analysis (*Boulton*, 1963).

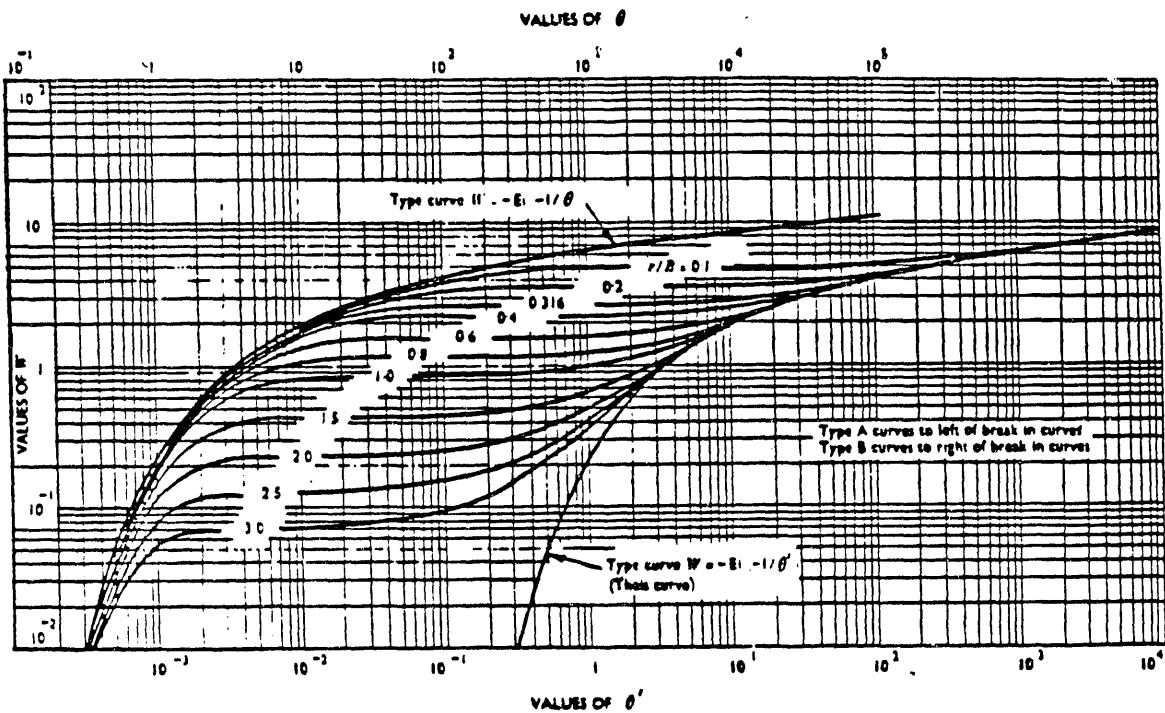


Figure 2.3: From *Boulton* (1963) paper: delayed yield type curves: dimensionless pressure (W) vs. dimensionless time (θ)

A practical example of these methods applied to observation wells was given. There is a comment on page 375, in which *Boulton* (1970) explains the deviation of the early time behavior from the type A (early *Theis*) solution, shown in his Fig. 2 by the need for a “type-curve based on a differential equation which contains an additional term for the short-term delayed yield”. He justified the simplification of the problem by stating that the parameters involved in this early period of flow are “of small importance”.

In order to enhance the solution (*Boulton*, 1954-a, 1954-b, 1963) for drawdown of an unconfined aquifer, an additional empirical term allowing for early time delayed yield from storage was considered by *Boulton* and *Pontin* (1971). The new solution enabled one to consider anisotropy and partial penetration, as well as to calculate the specific yield with “greater confidence”. By considering a second delayed yield term, the new results at early times were not associated with the previous early Type A curve. A practical example of application was given, using the same pumping test reported in a previous paper (*Boulton*, 1970). The figures presented showed a better match of the theoretical solution with the data, even at early times.

Dagan (1964, 1967) proposed a perturbation technique to linearize the equations of unsteady flow toward a partially penetrating well in an unconfined aquifer which considered

the velocity potential and Darcy's Law to describe the motion of the liquid in an anisotropic aquifer of infinite extent. The resulting first and second order linearized equations were solved using Green's Functions for different sink configurations, considered as point or line sources. The restrictions requiring small drawdowns and vanishing sinks contained in the *Dagan* solutions limit use in the neighborhood of the sandface. However, the *Dagan* solutions have advantages over the *Boulton* (1963) solution, by the consideration of the vertical component of velocities. At least, better results than the *Theis* approach would be expected in the vicinity of a well. The problem posed by *Dagan* neglected compressible effects in the unconfined aquifers, later included by *Neuman* (1974) by incorporating the aquifer compressibility in the *Dagan* (1967) equations.

Streltsova (1972-a) and (1972-b) considered slow drainage as the delayed process of vertical transfer, and not a delayed yield as did *Boulton* (1954b, 1963). She reduced the tri-dimensional problem to a two-dimensional problem by considering an average head along the vertical coordinates, given by the relationship:

$$h = \frac{1}{h^o} \int_0^{h^o} \Phi(r, z) dz \quad (2.21)$$

Therefore, the unsteady radial free surface flow partial differential equation is:

$$T \left(\frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial h}{\partial r} \right) = S \frac{\partial h}{\partial t} + S' \frac{\partial h^o}{\partial t} \quad (2.22)$$

The specific rate of the vertical transfer of flow was assumed to vary linearly with the difference between the average head h and the free surface h^o :

$$\frac{\partial h^o}{\partial t} = \alpha (h - h^o) \quad (2.23)$$

The relationship between α and other properties is:

$$\alpha = \frac{1}{S'} \Phi = \frac{k}{S' b} \quad (2.24)$$

where k and b are the average value of the permeability coefficient and the thickness of the vertical flow, respectively, and Φ is the specific hydraulic conductivity. A constant rate inner boundary condition given by the following equation was considered, as $r \rightarrow r_w$:

$$Q = 2\pi k H r \left(\frac{\partial h(r, t)}{\partial t} \right) = \text{constant} \quad (2.25)$$

and the initial condition is:

$$h(r, 0) = h^o(r, 0) = H. \quad (2.26)$$

The combination of the differential equations permitted the partial differential equation (Eq. 2.22) to be represented in terms of h and the radial distance from the well axis r :

$$T \left(\frac{\partial^2 h}{\partial r^2} + \frac{1}{r} \frac{\partial h}{\partial r} \right) = S \frac{\partial h}{\partial t} + S' \int_0^t \frac{\partial h}{\partial \tau} \left[1 - e^{-\alpha(t-\tau)} \right] d\tau \quad (2.27)$$

Equation 2.27 has a similar solution to the *Boulton* (1963) problem for which compressible flow was neglected, i.e., $S = 0$:

$$h = H - \frac{Q}{2\pi T} \int_0^\infty \frac{J_0(\frac{r}{B}x)}{x} \left\{ 1 - \frac{1}{x^2 + 1} e^{\left(-\frac{\alpha t x^2}{x^2 + 1}\right)} \right\} dx \quad (2.28)$$

A solution for h^o , which is the position of the free surface with time, was also obtained by assuming an exponential rate of discharge resulting from the discontinuity in head at a well:

$$Q^o = Q \left(1 - e^{-\alpha t} \right) \quad (2.29)$$

and so:

$$h^o = H - \frac{Q}{2\pi kT} \int_0^\infty \frac{J_0(\frac{r}{B}x)}{x} \left\{ 1 - \exp \left(-\frac{\alpha t x^2}{x^2 + 1} \right) \right\} dx \quad (2.30)$$

In a subsequent paper, *Streltsova* (1973) analyzed the contribution of the unsaturated zone (capillary forces and additional retarded flow) above the free surface to the head drawdown response in the wellbore, and concluded that the unsaturated zone contributions were of minor importance compared with the vertical fall of the water table. An important observation in this paper was the conclusion that the specific yield for vertical flow is a constant property of the formation, and the relationship between the unsteady water table heights and the flux at the water table averaged over the area is also constant. However, the free surface and the flux have an exponential change with time. In this same paper, *Streltsova* showed results from an experimental radial flow model using an analog electrical resistance network for simulating unconfined flow around the well. These results were compared with results of analytical solutions from the *Boulton* (1963) type-curves for average drawdown, and with the *Barenblatt et al.* (1960) solution for a similar equation for the free surface drawdown applied to unconfined flow. A good match between experimental

and analytical results was used to verify the theory, and to show the degree of reliability of assumptions of neglected flow from the unsaturated region and capillary pressure effects.

Complementing the previous work, *Streltsova and Rushton* (1973) presented a table for the definite integral $W_o[r/B, u]$ defined in Eq. 2.30, and compared the *Streltsova* (1972-b) solution based on the previous assumptions and the *Boulton* (1963) integral $V[r/H, t]$ given by Eq. 2.13. The basic difference between the two methods is that the former's main assumption is a finite difference approximation for the free surface condition (see Eq. 2.23), whereas the latter makes an assumption of constant discharge per unit length over the entire depth of the well and establishes an empirical relationship creating the delayed yield parameter. The comparison is presented in both table and graph. A computer code for the main part of a program to calculate the position of the free surface according to the *Streltsova* approach is available from the author.

The elastic properties of the aquifer were considered by *Streltsova* (1974) in solving the gravity drainage flow problem for a partially penetrating line-sink. Different boundary conditions and a more rigorous mathematical approach were used, instead of the finite difference approximation described before. In the conclusions, it was emphasized that the elastic properties of the aquifer were essential during the early periods of pumping before gravity drainage has started.

A new analytical approach was presented by *Neuman* (1972) which defined a set of equations, considering formation anisotropy and constant values of specific storage and specific yield with no empirical considerations. According to the author, this method eliminates the conceptual difficulties in the *Boulton* theory of "delayed yield" involved in the early time flow behavior of an unconfined aquifer, and suggested calling the physical process "delayed response". Also, a comparison with the results from a numerical solution using the *Cooley* (1971) finite difference model showed the unsaturated region flow to be important to the drawdown in a free surface aquifer. The *Neuman* model treated the unconfined aquifer as a compressible system and the free surface as a moving boundary. One common assumption in the *Neuman* theory as well as that of others is the treatment of the well as a line sink. This simplification of the problem, using the author's words, introduces "a certain error in the solution near the well bore" by neglecting the presence of a seepage face. The governing set of equations for the three-dimensional problem is:

$$(0 < z < \xi) \quad K_r \frac{\partial^2 s}{\partial r^2} + \frac{K_r}{r} \frac{\partial s}{\partial r} + K_v \frac{\partial^2 s}{\partial z^2} = S_s \frac{\partial h}{\partial t} \quad (2.31)$$

$$s(r, z, 0) = 0 \quad (2.32)$$

$$\xi(r, 0) = 0 \quad (2.33)$$

$$s(\infty, z, t) = 0 \quad (2.34)$$

$$K_r \frac{\partial s}{\partial r} n_r + K_z \frac{\partial s}{\partial z} n_z = \left(S_y \frac{\partial \xi}{\partial r} - I \right) n_z \quad \text{at} \quad (r, \xi, t) \quad (2.35)$$

$$\xi(r, t) = b - s(r, z, t) \quad (2.36)$$

$$\lim_{r \rightarrow 0} \int_0^\xi r \frac{\partial s}{\partial r} dz = - \frac{Q}{2\pi k_r} \quad (2.37)$$

Equations 2.31 and 2.37 were solved using a perturbation technique and Laplace and Hankel transforms. The solution is given in the Appendix of *Neuman* (1972) and is:

$$s(r, z, t) = \frac{Q}{4\pi T} \int_0^\infty 4x J_0[x\sqrt{k_D}] \left[w_o(x) + \sum_{n=1}^{\infty} w_n(x) \right] dx \quad (2.38)$$

where, $w_o(x)$ and $w_n(x)$ are expressions given by Eq. 15 in the original paper. That solution is a function of the coordinate z . In order to find an average solution over the aquifer thickness in an observation well, the following expression was used:

$$s(r, z, t) = \frac{1}{z_2 - z_1} \int_{z_2}^{z_1} s(r, z, t) dz \quad (2.39)$$

where z_1 and z_2 are elevations of the open sandface in the observation well. For a well that completely penetrates the aquifer, the final result for the average drawdown is calculated from Eq. 2.38 after redefining $w_o(x)$ and $w_n(x)$ as in Eqs. 20 and 21 in *Neuman* (1972). The influence of each dimensionless parameter was studied and interesting graphs presented. See Figs. 4 and 5 in *Neuman*, 1972, pp. 1037-1038. *Neuman*'s Fig. 4 suggests an approximate linear relationship between the logarithm of water table drawdown and the logarithm of time (curve corresponding to $z_D = 1$), and Fig. 5 presents a dimensionless constant position $z_D \approx 0.57$ for the average drawdown over the entire aquifer. The *Neuman* (1972) solution was supplemented by a subsequent paper (*Neuman*, 1973) whose main purpose was to modify the original Eq. 27 (*Neuman*, 1972), allowing for anisotropy. The result presented in Eq. 2.40 is similar to the *Boulton* (1954-b) solution:

$$s(r, z, t) = \frac{Q}{4\pi T} \int_0^\infty 2J_0[y\sqrt{\eta}] \left\{ 1 - \exp(-t_y \eta y \tan(y)) \frac{\cosh(yz_D)}{\cosh(y)} \right\} \frac{dy}{y} \quad (2.40)$$

The new theory (*Neuman*, 1972, 1973, 1974) was applied to pumping tests for fully and partially penetrating wells by *Neuman* (1975-a). Besides the general procedure of curve-matching, this paper introduced a method of interpretation based on the semilogarithmic relationship between drawdown and time. For type-curve analysis, the methodology follows that previously described by *Prickett* (1965). The semilog interpretation is based on the linear behavior of drawdown with the logarithm of time at late times when the rate of change in the free surface has become small. However, due to the assumption of a line-sink well, the theoretical early time behavior is also supposed to fall on a semilog straight line since the initial flow from the formation into the wellbore is essentially horizontal. The practical example of application of this method in analyzing pump test data seems to be conclusive with respect to the late time behavior, but not well fitted to the early time data points for a pumping well. In describing this method, *Neuman* suggests deleting the early time data analysis if the slope of the straight line obtained differs noticeably from the straight line fitted to the late time data points. In this case, the method of interpretation should be complemented by matching of type-curves. Another important consideration made in the paper by *Neuman* (1975-a) concerns recovery (buildup) tests. Because no flow from the unsaturated region is considered in this method, and the unconfined aquifer is treated as a compressible system, no hysteresis is expected. As a consequence, the same analytical equations and interpretation methods should apply for both drawdown and buildup. However, *Ramey et al.* (1989) observed that the semilog graphs for drawdown and buildup show much different times for the start of the straight line behavior for the same well test data. This and similar physical observations make questionable the reversibility of the process of gravity drainage under the free surface. In this same paper, *Neuman* (1975-a) relates the *Boulton* reciprocal delay index α with aquifer properties by establishing a numerical relationship between *Boulton* type-curves expressed by the dimensionless parameter $(r/B)^2/b$ versus s_D . On semilogarithmic paper, a linear relationship was obtained by regression analysis:

$$\frac{(r/\beta)^2}{\beta} = 3.063 - 0.567 \log \beta \quad (2.41)$$

Using definitions of r/B , β , and α , Eq. 2.41 becomes:

$$\alpha = \frac{k_z}{s_y b} \left[3.063 - 0.567 \log \left(\frac{k_D r^2}{b^2} \right) \right] \quad (2.42)$$

Equation 2.42 represents a connection between the *Boulton* and *Neuman* theories. *Neuman* also recognized the relationship of the α -parameter and physical properties of the aquifer in the *Streltsova* study, but claims that the relationship exists only as a function of distance

r (see Eq. 2.42), and not of time, as is clear from the dependence of s_{wT} in the following Eq. 2.43 by *Streltsova* (1972-b):

$$\alpha = \frac{3k_z}{s_y(b - s_{wT})} \quad (2.43)$$

Streltsova (1976) believes that the *Neuman* boundary condition for the free surface (*Neuman*, 1972) leads to a mathematical formulation equivalent to *Boulton* (1954-b, 1963) and *Streltsova* (1972). Therefore, she concludes that the solutions are identical, and claims that once the α -parameter is determined, it is possible to determine the degree of anisotropy by calculating μ from the relationship with α :

$$\alpha = \frac{3k_z}{S' h^o} = \frac{3\mu^2}{h^o S'}, \quad (2.44)$$

where:

$$\mu = \frac{k_z}{k_r}$$

This procedure yields a different k_z from the *Neuman* theory due to the approximation in the *Streltsova* model relating the average drawdown and the drawdown of the free surface by a finite difference relationship given by Eq. 2.23. Plots of β and $(r/B)_2$ versus s_D are matched to demonstrate the correlation between the dimensionless parameters. These considerations were denied by *Neuman* (1976), asserting that the matching was "grossly misleading".

The discussions with respect to delayed yield are summarized in a paper (*Neuman*, 1979) which compares the relevant theories on delayed yield. *Neuman* seems to have adopted the term "delayed yield" to characterize the phenomena which he previously suggested (*Neuman*, 1972) should be referred to as "delayed gravity response". A brief speculation about *Boulton* theory raises the possibility of delayed yield being caused by delayed air entry in the high water saturation porous region above the free surface. However, referring to several authors, and specifically to works by *Kroszynski* and *Dagan* (1975), *Neuman* (1975-b) and *Neuman* et al. (1974), *Neuman* concludes that the unsaturated flow has a minor influence on the time-drawdown response during gravity drainage of liquid. A justification for large apparent storage coefficients S in unconfined aquifers compared to artesian aquifers is explained by entrapped air bubbles or dissolved air in the saturated zone. Thus, the fluid compressibility could have large values. Therefore, *Neuman* uses this argument against the idea of *Bouwer* and *Rice* (1978) that the delayed air entry is due to the slow movement of liquid in the unsaturated porous space above the free surface. The liquid in the capillary fringe is drained slowly as a consequence of a negative pressure differential created when the liquid surface moves vertically down. *Neuman* again referred to the rate of head recovery

(buildup) as a fully-reversible mirror-image process, and, therefore, incompatible with the theory of the air entry concept of *Bouwer* and *Rice*. *Neuman* criticized the idea that the influence of radial distance on the *Prickett* (1965) and *Gambolati* (1976) theories that at a certain time during the desaturating period the region around the wellbore may be divided into three different sub-regions of different flow patterns. *Neuman* stated that “the cone of depression is not distorted, and its slope varies monotonically with radial distance”.

The most important theories of gravity drainage flow so far reviewed do not consider the flow in the unsaturated zone rigorously. Even the *Boulton* (1954-b) model does not consider the characteristics of the region above the free surface directly. Instead, the *Boulton* model approximates this effect by creating the delayed yield parameter α . In the following review, flow in the unsaturated zone will be considered in regard to the gravity flow problem.

A complete analytical solution reflecting the flow in the unsaturated region above the free surface, an anisotropic formation, and a partially penetrating line-sink well were considered by *Kroszynski* and *Dagan* (1975). According to *Neuman* (1979), this is “... the most definitive study about the role of unsaturated flow ...” published so far. Using the same original notation, the following equation is used to represent the flow in the unsaturated zone:

$$\frac{D'}{S_v} \frac{d\theta}{d\Psi'} \frac{d\Phi}{dt} = \nabla (k(\Psi) \nabla \Phi) \quad (2.45)$$

where D' is the initial saturated thickness of the aquifer, θ is the volumetric moisture content, Ψ' is the capillary head, k is the relative hydraulic conductivity (function of θ or Ψ), and Φ is the total head corresponding to the isolated sink solution. The other equations and the analytical solution will not be presented here. The model was tested by comparing results with numerical solutions from a modified program after *Neuman* (1975-b). The results indicate little influence of the unsaturated flow on the time-drawdown response. Only for special conditions of thin and low permeability aquifers were significant effects observed at early times. Another important consideration is the finite wellbore radius used in the numerical model to simulate wellbore storage effects. Unfortunately, *Kroszynski* and *Dagan* did not apply this model to the producing well, and the closest observation well was located at 4.4 ft from the symmetry axis.

Bouwer and *Rice* (1978) presented a physical basis for delayed yield response as a restriction of the unsaturated zone to air entry due to the fall of the water table. As the free surface moves down, air penetration into the rock pores is impeded by the high initial water content of this region reducing the relative permeability to air. The initial specific yield is small, but increases with time as vertical drainage occurs. This process is important if the draining region contains layers of fine-textured materials which have

high water retention. In this case, the entry capillary pressure retards the air movement downward. Once started, the pressure directly above the water table varies from vacuum to equilibrium at atmospheric pressure. The time lag is reflected in the variation of the specific yield. *Bouwer* and *Rice* used these principles to run laboratory experiments to explore field observations. Apparently, the typical inflection in the time-drawdown curve observed for pumping tests could not be reproduced in the laboratory, and, according to *Neuman* (1979), the results were not conclusive.

Another physical explanation for delayed yield is attributed to the hysteresis effect on the capillary fringe and transition zone above the water-table, and also underneath the free surface (*Bouwer*, 1979). According to *Bouwer*, pumping tests are often done in wells that were previously operated. As a consequence, the region of the reservoir around the wellbore, once drained by pumping the well, is not completely resaturated after the recovery of the initial head after pumping is stopped. Hence, bubbles of air, entrapped in the initial capillary fringe or in the previously saturated zone, will create a more compressible system than the original one, causing a delay in releasing significant amount of water from the pores. Therefore, the fall of the water-table will be retarded by an increase of the compressibility in that region. In *Bouwer*'s opinion, the air compression effects could be a common cause of delayed yield, in addition to the elastic response of the aquifer and the restricted air entry. There was no mention of a reduction in transmissivity caused by the presence of another phase.

Analyzing pumping tests for an unconfined aquifer in Canada, *Nwankwor et al.* (1984) made a comparative study of their determinations of specific yield with results from other methods. The tests were run in a pattern formed by a centered pumping well and observer wells distributed into two perpendicular lines to monitor the drawdown at different levels in the aquifer. Using the *Boulton* (1963) and *Neuman* (1972) theories, the authors observed that at early times very low results for specific yield were calculated from both methods. As time increased, specific yield increased slightly, but never approached the values calculated from the volume-balance method or determined by laboratory measurements. The volume-balance method consists of calculating the cumulative water produced V_w and dividing it by the volume of the cone of water-table drawdown V_c , obtained from observation well drawdown mapping. Despite that discrepancy, *Nwankwor et al.* concluded the various methods of interpretation give similar results for transmissivity. The apparent reasons for the differences in the long-term specific yield were attributed to neglecting the delayed drainage from above the water table in the *Neuman* (1972) theory, and the downward hydraulic gradients in the *Boulton* theory. Therefore, a more rigorous model considering flow in the unsaturated zone and vertical flow processes simultaneously should be more appropriate.

The large differences in the specific yield described before caused concerns about

the reason for such deviations. A convincing explanation for this problem was given by *Neuman* (1987). A superficial analysis of the *Nwankwor et al.* (1984) study shows an invalid assumption concerning the volume-balance method. The criteria used in performing that balance is that the total volume of water produced by a well comes from the depressed cone in the reservoir. Therefore, applying the ratio of volumes, the specific yield would vary, and reach high values after a long time. Regarding laboratory results, one is supposed to accept the concept of specific yield as the difference between the saturated core and residual water in the pores of the rock. Apparently only a fraction of the produced water comes from the depressed cone. *Nwankwor et al.* (1984) neglected the volume of water that comes from the outer reservoir. *Neuman* (1987) demonstrated this idea in two different ways: an approximate calculation based on the presented data (*Nwankwor et al.*, 1984), and a theoretical approach. In other words, the water balance was not valid because only a small fraction of the water comes from the observed cone of depression. *Neuman*'s theoretical analysis was also confirmed by the pumping test data from *Nwankwor et al.*, and substantially supports the idea that the variation of the free surface "in response to pumping is relatively insensitive to residual drainage in the unsaturated zone".

2.4 Inner Boundary Condition

Until the middle sixties, the inner boundary condition had not been treated adequately in hydrology, and only the line source well (*Theis*, 1935) had been considered in pumping tests. Wellbore storage and finite radius well effects were neglected, mainly in shallow and unconfined aquifers where most of the pressure observations in pumping tests were usually performed in shut-in observation wells, not in the producer. In petroleum engineering, an early application of the finite well radius and well storage was presented by *Moore et al.* (1933). This was a remarkable early study of transient flow which included an example of field well test data interpretation. Sonic annulus liquid levels were measured and sandface flow rates found by subtracting production from the wellbore storage. A variable rate analysis was used to interpret the well test data. One of the authors, *W. Hurst* was a co-author of the classic paper *van Everdingen and Hurst* (1949) which included both finite-producing well radius solutions and defined wellbore storage and produced a log-log storage type curve which is the basis of modern type curve data interpretation.

Wellbore conditions were completed defined in 1953 by the *van Everdingen and Hurst* (1953) skin effect concept. The skin effect defined well condition quantitatively by adding a dimensionless resistance to flow at the sandface. In the following, a brief analysis of selected studies of wellbore condition is presented.

In 1964, *Hantush* reviewed solutions of the continuity equation for a well of finite radius in a confined aquifer, without storage effects. Consideration of wellbore storage in analyzing pumping tests for a confined aquifer was presented by *Papadopoulos and Cooper* (1967), who appeared to be unaware of the earlier study by *van Everdingen and Hurst* (1949). The wellbore storage inner boundary condition can be represented by a material balance equation:

$$\boxed{\text{sandface flow rate}} + \boxed{\text{wellbore flow rate}} = \boxed{\text{pumping rate}} \quad (2.46)$$

$$2\pi r_w T \frac{\partial s(r_w t)}{\partial r} - \pi r_c^2 \frac{\partial s(t)}{\partial t} = -Q \quad (2.47)$$

The solution in Laplace space is:

$$\bar{s} = \frac{Q K_o(q r)}{\pi p [r_c^2 p K_o(q r_w) + 2 r_w T q K_1(q r_w)]}, \quad (2.48)$$

where, \bar{s} is the Laplace transform of the drawdown, p is the Laplace parameter, and $q = \left(\frac{pS}{T}\right)^{1/2}$. The inversion, obtained from an analogous problem in heat flow (*Carslaw and Jaeger*, 1959), gives the drawdown at any distance r from the well bore center:

$$s = \frac{2Q\alpha}{\pi^2 T} \int \left(1 - e^{-\beta^2/4u_w}\right) \{J_o(\beta r/r_w) [\beta Y_o(\beta) - Y_1(\beta)] - Y_o(\beta r/r_w) [\beta J_o(\beta) - 2\alpha J_1(\beta)]\} \frac{d\beta}{\beta^2 \Delta(\beta)} \quad (2.49)$$

where, $\alpha = r_s^2 S / r_c^2$, and $u_w = r_s^2 S / 4Tt$. When $r = r_w$, Eq. 2.49 simplifies to:

$$s_w = \frac{Q}{4\pi T} F(u_w, \alpha), \quad (2.50)$$

where

$$F(u_w, \alpha) = \frac{32\alpha^2}{\pi^2} \int_0^\infty \frac{\left(1 - e^{-\beta^2/4u_w}\right)}{\beta^3 \Delta(\beta)} d\beta. \quad (2.51)$$

In the *Papadopoulos* and *Cooper* paper, Fig. 2 shows a set of type curves based on this solution, and compares several results for parameters α (related to the well bore storage), with the *Theis* and *Hantush* solutions. As an addendum to the previous work (*Papadopoulos* and *Cooper*, 1967), *Papadopoulos* (1967) presented an additional logarithmic plot of $s/Q/4pT$ versus $r^2S/4Tt$ for several dimensionless distances from the well.

Cooper et al. (1967) use a procedure similar to *Papadopoulos* and *Cooper* (1967) to study the “slug-test” in a confined aquifer. Equation 2.50 is used as an inner boundary condition, and the final solution for the head in Laplace space has the same format as Eq. 2.48, here represented by Eq. 2.53:

$$2\pi r_s T \frac{\partial h(r_s + 0, t)}{\partial r} = \pi r_c^2 \frac{\partial H(t)}{\partial t}, \quad (2.52)$$

$$\bar{h} = \frac{r_s S H_o K_o (q r)}{T q [r_s q K_o (q r_s) + 2\alpha K_1 (q r_s)]}, \quad (2.53)$$

where, q and α have the same definition as before, and H_o is the initial head increase in the well. Analogously, the solution at the producing well, i.e., when $r = r_s$, is:

$$H = \frac{8H_o\alpha}{\pi^2} \int_0^\infty \frac{e^{-\beta u^2/\alpha}}{u\Delta(u)} du \quad (2.54)$$

Kipp (1973) presents another theoretical solution for the unconfined flow problem. A homogeneous isotropic aquifer of infinite thickness and infinite lateral extent, partially penetrated by a well of finite radius was assumed. Other important assumptions were single phase flow, an incompressible system and no surface of seepage in the well bore. A set of boundary condition equations in terms of velocity potential was solved using a perturbation technique. The approximations made restricted applications of the semianalytical results to short times after pumping start-up. The line-sink solution by other authors (*Boulton*, 1954-a, and *Dagan*, 1967) was found to have reasonable agreement with the *Kipp* solution, at radial positions greater than 15% of the well depth below the initial free surface. Although some graphical comparisons were shown, no attempt to create dimensionless type curves was made. There are some disadvantages of this solution with respect to analysis of well tests besides limitations imposed by the initial assumptions and the solution method, namely a “tedious trial and error search is necessary to determine the aquifer parameters”.

A simple procedure for estimating aquifer properties by a slug test is shown in a paper by *Bouwer* and *Rice* (1976) using a modified *Thiem* (see reference in the paper) equation and a resistance network analog computer. The method is based on the assumption

that the nature of flow around the well is predominantly radial during a pressure recovery after a fixed volume of water is instantaneously removed from the wellbore:

$$Q = 2\pi K L \frac{y}{\ln(r_e/r_w)} \quad (2.55)$$

where y is the instantaneous position of the water level in the well, with respect to the original water-table and r_e is the effective drainage radius in which the slug volume is dissipated. The drainage radius is obtained by empirical correlations from the electrical model, and is expressed as a function of parameters which depend only on the geometry of the system. The expressions for both a partially and fully penetrating well are given below.

partially penetrating well:

$$\ln \frac{r_e}{r_w} = \left[\frac{1.1}{\ln(H/r_w)} + \frac{A + B \ln[(D - H)/r_w]}{L/r_w} \right]^{-1} \quad (2.56)$$

fully penetrating well:

$$\ln \frac{r_e}{r_w} = \left(\frac{1.1}{\ln(H/r_w)} + \frac{C}{L/r_w} \right)^{-1} \quad (2.57)$$

where A, B and C are empirical functions of L/r_w obtained from Fig. 3 in the *Bouwer and Rice (1976)* paper. To estimate the aquifer conductivity, Eq. 2.58 was derived from integration of Eq. 2.55:

$$t = \frac{r_c^2}{2KL} \ln \left(\frac{r_e}{r_w} \right) \ln \left(\frac{y_o}{y_t} \right) . \quad (2.58)$$

A semilog plot of the ratio y_o/y_t was used to determine K . According to *Bouwer and Rice*, the approach used to determine $\ln(r_e/r_w)$ yielded values of K 10% to 25% in error, depending on the geometric characteristics of the system.

Dagan (1978) presented an approach for interpreting packer, slug and recovery tests in unconfined aquifers in order to determine the permeability of low conductivity aquifers. He did not consider variations of the free surface due to small volumes drained from the system for those type of tests. The method consisted in integrating the source function continuously through the open interval in the well bore:

$$Q = \int_0^L q(z) dz , \quad (2.59)$$

where:

$$q = 2\pi r_w K \frac{\partial \Psi}{\partial \rho}, \quad r = r_w, \quad 0 < z < L \quad (2.60)$$

Using a Green's function to define $d\Psi(r, z, \bar{z})$ for an element along $r = 0$ centered at $z = \bar{z}$ with sources of strength $q(\bar{z})$ distributed along the z axis, the solution produces the integral

$$\Psi(r, z) = \int_0^L \frac{q(\bar{z})}{K} G(r, z, \bar{z}) dz \quad (2.61)$$

The integral in Eq. 2.61 is discretized and a finite sum is used instead:

$$\sum_{i=1}^N \frac{q(\bar{z})}{K} A_{ij} = \beta_{ij} \quad (2.62)$$

The length L is divided in N equal intervals of length $\Delta L = L/N$ and $q(z)$ is replaced by q_i ($i = 1, 2, \dots, N$); β_j ($j = 1, 2, \dots, N$) is started from the boundary condition and stands for $\beta[z = (j - 0.5)\Delta L]$ at the middle of the j^{th} interval. The system composed of Eq. 2.62 is solved for q_i/K by a computer program. Therefore, the result from Eq. 2.55 is:

$$\frac{Q}{K} = \Delta L \sum_{i=1}^N \frac{q_i}{K} \quad (2.63)$$

The hydraulic conductivity is:

$$K = \frac{r_w^2}{2\bar{Q}L(t - t_o)} \ln \left[\frac{(2L - y)/(2L - y_o)}{y/y_o} \right] \quad (2.64)$$

where \bar{Q} is generated from a computer program, and:

$$\bar{Q} = \frac{1}{2\pi Ly[1 - (y/2L)]} \frac{Q}{K} \quad (2.65)$$

The main limitation of this method is that the length of the sandface open to flow must be much larger than the wellbore radius.

Rushton and *Holt* (1981) compared analytical and numerical solutions for gravity drainage for aquifers and concluded that the numerical solutions were accurate in representing the phenomenon of unconfined flow. *Rushton* and *Holt* developed a finite difference form to solve the partial differential equation, Eq. 2.66, and simulated the wellbore storage effect by including a portion of the aquifer that represents the region within the well in the model.

$$\frac{\partial^2 s}{\partial r^2} + \frac{1}{r} \frac{\partial s}{\partial r} = \frac{S}{T} \frac{\partial s}{\partial t} \quad (2.66)$$

The transmissivity was set to a very high value in the well partly to simulate the horizontal water level in the well, while the storage coefficient (equivalent to the effective porosity) was set to unity. The authors explained superficially that liquid level variations in the saturated depth was handled by considering the transmissivity equal to the permeability multiplied by the saturated depth, which was a function of the drawdown. Probably an iterative procedure was used if the drawdown was unknown. Figure 4 in *Rushton* and *Holt* compared the numerical results with field data, and showed good agreement with the early and late time drawdown. However, the intermediate period was poorly matched.

A kernel approach was used by *Patel* and *Mishra* (1983) for unsteady flow to a large diameter well in a confined aquifer taking wellbore storage into consideration. Using $s(r, 0) = 0$ for the initial condition and $s(\infty, 0) = 0$ for an external boundary condition, the solution of Eq. 2.66 when a unit impulse quantity is withdrawn from the aquifer is given by (*Carslaw and Jaeger*, 1959):

$$s(r, t) = \frac{e^{-[r^2/4\beta t]}}{4\pi T t} \quad (2.67)$$

where $\beta = T/\Phi$. By defining a unit kernel function $k(t)$ as the right hand side of Eq. 2.67, the drawdown for a variable pumping rate $Q_A(t)$ is:

$$s(r, t) = \int_0^t Q_A(c) k(t - c) dc \quad (2.68)$$

From discretizing and applying Eq. 2.68 to the sandface:

$$s_A(n) = \sum_{z=1}^n Q_A(z) \delta_{r_w}(n - z + 1) \quad (2.69)$$

where $\delta_{r_w}(m)$ is the discretized kernel coefficient at the sandface. For the wellbore, the discretized drawdown for the time step n is:

$$s_w(n) = \frac{1}{\pi r_c^2} \sum_{p=1}^n Q_A(p) \quad (2.70)$$

The combination of Eqs. 2.69 and 2.70 through a material balance in the well $Q_p = Q_A(n) + Q_w(n)$ and the continuity of the drawdown through the sandface into the well (no seepage face in the confined flow), $S_A(n) = S_w(n)$, the following solution was obtained in a matrix form:

$$\begin{vmatrix} Q_A(n) \\ Q_w(n) \end{vmatrix} = \begin{vmatrix} \delta_{rw} - \frac{1}{\pi r_c^2} & -1 \\ 1 & 1 \end{vmatrix}^{-1} \begin{vmatrix} \frac{1}{\pi r_c^2} \sum_{p=1}^n Q_A(p) - \sum_{z=1}^n Q_A(p) \delta_{rw} (n - z + 1) \\ Q_p \end{vmatrix} \quad (2.71)$$

Dimensionless curves were generated after calculating $s(r, t)$ and s_w from Eqs. 2.68 and 2.70. The results match with those from *Papadopoulos and Cooper (1967)*.

A consideration of the seepage face was presented in a method of interpretation of pumping tests by *Rushton and Singh (1987)*. This method has an important restriction because it assumed that the change in the saturated depth was negligible. Hence, the superposition principle may be applied the same way as in the work of *Patel and Mishra (1983)*. The method also depended on empirical relationships between the height of the seepage face and the withdrawal rate from the aquifer, Q_A . The suggested relationship was of the form:

$$f = G_1 Q_A + G_2 Q_A^2 \quad (2.72)$$

Rushton and Singh based their model on an unpublished report by *Sakthivadivel (1986)*, (see reference in the paper), and the parameters G_1 and G_2 must be deduced from field measurements. Coupling Eq. 2.72 with a material balance in the well (see Eq. 2.47), the resulting set of discretized equations allowed the iterative calculation of the sandface rates, and, consequently, the determination of the drawdown distribution. The paper also presented a comparison of two different approaches: one considering the seepage face and the other neglecting the seepage face. The results showed that the resulting transmissivity could be underestimated by 25%, while the storage coefficient value could be of the order of one-fifth of the true value in the simplified model neglecting the seepage face. The example shown in Figs. 5a and 5b in *Rushton and Singh (1987)* shows Cartesian graphs of drawdown and discharge from the aquifer versus time for both field data and computed results. Our observation of the plot of drawdown versus time identifies a linear relationship for the early time drawdown points. The early time recovery points also show the same behavior, although for a different period of time. Such a linear period is analogous to the

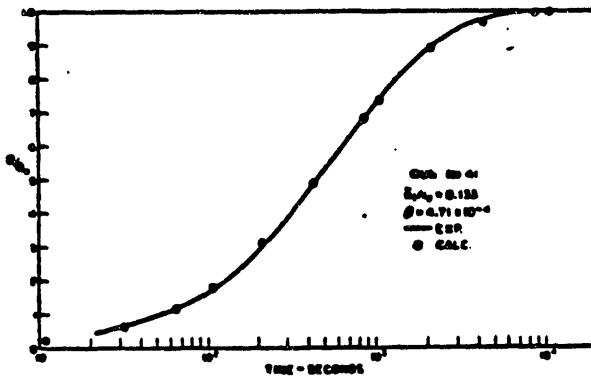


Figure 2.4: Cumulative Production for Small Values of \bar{h}_i/r_e . - from *Matthews and Lefkovits* (1956).

well bore storage effects in confined systems. However, fitting the calculated curves by a theoretical approach is open to discussion.

2.5 Unconfined Flow in the Petroleum Industry

Although *Muskat* (1949) presented a brief discussion of the free surface gravity flow well problem, there have been only a few works concerning gravity drainage well test analysis in the petroleum literature. In 1956, *Matthews and Lefkovits* published a paper presenting some approximate expressions to calculate the production rate decline for homogeneous reservoirs with a free surface pumped at a constant wellbore pressure (liquid level). The flow dynamics were neglected in the derivations which were based on the drainage radius and the *Dupuit* formula to get the following second order hyperbolic expression, not valid in the early stage of production:

$$\frac{q}{q_i} = \frac{1}{(1 + \gamma t)^2}, \quad (2.73)$$

where q_i is the rate of oil production at time $t = 0$ ($t = 0$ may be taken arbitrarily), and γ is given by the following expression, translated into the symbology used in the present dissertation:

$$\gamma = \frac{k_o \Delta \rho g H_o}{\phi (S_o - S_{or}) \mu_o \frac{1}{2} \left[1 + \sqrt{1 - \frac{1}{2 \ln \frac{r_e}{r_w} - 1}} \right] \left[\ln \frac{r_e}{r_w} - \frac{1}{2} \right]} \quad (2.74)$$

A scaled sandbox model was built to verify particular field data and used to validate the theoretical approach. There were two important conclusions from this work: (a) a completely pumped-off well produces at higher rates than a partially pumped well, and higher flow rates may be obtained in completely pumped-off wells if the rates were reached by small increases rather than through sudden well withdrawal, and (b) as shown in Fig. 2.4, the time derivative of the cumulative production indicates flow rates increasing to a maximum value in the early seconds of pumping (out of Eq. 2.73 validity range), probably due to the potential gradient development in the vicinity of the well.

In recent years, there has been interest in evaluating depleted fields for enhanced oil recovery projects. Many of these fields produce under a pure gravity drainage mechanism and available methods of well test interpretation are not appropriate. *Ramey et al.* (1989) was the first paper in many years to consider this sort of problem. Starting from a material balance on an infinitesimal radial flow element, the *Dupuit-Forchheimer* partial differential equation was obtained:

$$\frac{\partial^2 p^2}{\partial r^2} + \frac{1}{r} \frac{\partial p^2}{\partial r} = \left(\frac{\phi \mu (p^{-1})}{0.000264k} \right) \frac{\partial p^2}{\partial t^2} \quad (2.75)$$

where p is the pressure at the bottom of the formation. In the formulation, both the system compressibility and vertical velocity variations along the formation thickness were neglected. By inspection, Eq. 2.75 resembles the ideal gas equation of *Aronofsky and Jenkins* (1953). Since the simplifying assumptions are more appropriate at late times, the approximate solution for Eq. 2.75 (also shown in Ch. 1 as Eq. 1.3) was presented:

$$p_o^2 - p_w^2 = 325.2 \frac{q B \mu}{k(h/p)} \left\{ \log \left(\frac{0.000264kt}{\phi \mu (p_o^{-1}) r_w^2} \right) + 0.351 + \frac{s}{1.151} \right\} \quad (2.76)$$

Since vertical flow in the formulation was neglected, h/p could be considered constant and equal to the liquid gradient for incompressible flow. The *van Everdingen* (1953) skin parameter s was included in the inner boundary condition. From Eq. 2.76, a p^2 versus time semilog graph should produce a straight line, with a slope m that may be used to obtain the permeability - see Section 7.1. The *Ramey et al.* (1989) solution represented an improvement with respect to the *Theis* (1935) approach in that variation in height and the skin effect were considered. Also highlighted in the *Ramey et al.* paper is a discussion of the role of compressible effects in gravity transient flow behavior. Also, the reversibility of

the buildup pressure behavior was questioned because a different physical process occurred during buildup, compared to drawdown. Those subjects are further discussed in Chapter 7.

Equations of the type of Eq. 2.75 were solved numerically in a broader context including wellbore storage effects and a skin by *Fligelman* (1980) and is discussed in Ch. 7 and in Appendix C.2.

2.6 Numerical Solutions

With the development of large, fast computing machines, numerical methods became a powerful source of solutions for nonlinear problems, avoiding simplifications that were used in approximate analytical solutions with relaxed constraints. Numerical solutions have been an important tool to verify analytical methods when necessary, and to test correlations in general.

In the groundwater literature, the main references on numerical methods dedicated to the study of unconfined flow started in the 60's. *Cooley and Donohue* (1969) developed a numerical simulator to solve a finite-difference equation system for unsteady flow in a radial flow problem using two-phase flow in the desaturating region based on classical methods employed in the petroleum industry. This work was advanced by *Cooley* (1971) and presents a radial model geometrically divided into cylindrical blocks in the radial direction. In the vertical direction, constant thickness was allocated to each row. The potentials were the unknowns in each equation, and the flow was divided into radial (horizontal) and vertical directions ruled by the continuity equation coupled with *Darcy's Law*. Multiphase air-liquid flow was governed by capillary pressure. The posed problem did not establish a boundary condition for the free surface. Instead, the model had a fixed grid and calculated the potentials at every block, even those located in the desaturated region. The free surface position could be obtained indirectly by interpolation by the relationship of h (head) and z (vertical coordinate of the node points). The model permitted the existence of two different overlaying beds of sediments with different properties. Unfortunately, the inner boundary condition did not consider wellbore storage, and the wellbore liquid level was obtained by an iterative process. LSOR was the most general method used to solve the nonlinear system, generating tridiagonal matrixes, and the time interpolation factor used was 0.5. The model was verified using the *Boulton* approach applied to synthetic data of two observation locations far from the pumping well. There were no verification runs for the pumping well itself.

A similar method was described in a recent work by *Narasimhan and Zhu* (in press)

who used the *Agarwal et al.* (1970) definition of wellbore storage constant in the inner boundary condition and verified their finite difference model with a series of analytical solutions and numerical experiments. A better verification of the same model applied to radial flow into a gravity drainage well at steady conditions with respect to the inner boundary seepage face was done by *Shamsai and Narasimhan* (1991), by comparing results with the *Hall* (1955) experiments.

Szabo and McCaig (1968) created a general finite difference model to solve the transient gravity flow problem which considered single-phase fluid and a moving boundary free surface whose grid blocks presented adjustable triangular shapes between consecutive time steps. An analog device was used to compare results with the numerical solutions for a two-dimensional linear flow. *Szabo and McCaig* considered a constant head inner boundary condition.

Finite element methods have been widely used to simulate steady or unsteady flow conditions such as by *Neuman* (1975-b), *Neuman and Witherspoon* (1970-a, 1970-b, 1971). In the latter, a general transient gravity drainage model using a variational principle considered wellbore storage and had a Crank-Nicolson time interpolation method. The nodal distribution was variable to adjust to the free surface at the end of each time step. No wellbore solution or verification run was provided in the paper.

Two phase, one-dimensional flow was studied by *Green et al.* (1970) and *Hornberger and Remson* (1970) who developed finite difference models using relative permeability and capillary pressure relationships with liquid saturations to analyze moisture movement in the porous medium. The results in the *Green et al.* paper were compared with experimental measurements of the local saturations using neutron log devices. The importance of hysteresis caused by residual air saturation was highlighted and caused concerns about reversibility in the buildup analysis.

Special procedures in simulation are important to enhance the capability of numerical solutions. As an example, logarithmic and geometric grid spacing have been used in numerical simulation of radial flow into a centered well of accuracy and stability. A handy algorithm to create an adjustable log radial spacing by *Terán and la Garza* (1988) allows one to specify the ratio between the first and last cylindrical block sizes keeping truncation error in the same magnitude all over the grid. An advantage of this algorithm is to provide control of the inner radial block storage capacity reducing the dependence of the number of blocks in the radial direction, and also to provide different radial node distributions as desired with minor consequences on the computational efforts.

Further techniques of numerical simulation in petroleum engineering are presented in

many references, but most of the procedures in the development of the present dissertation are summarized in the classical books by *Peaceman* (1977), *Aziz* and *Setari* (1979) and *Thomas* (1982).

2.7 Contribution of the Present Work

As evidenced in the literature review, the lack of studies about the producing well with a free surface under gravity drainage during drawdown and buildup periods motivated the present dissertation. In the groundwater engineering, evalutation methods were developed only for observation wells. In the case of oil fields, however, the well distributions, producing flow rates and reservoir properties are generally different, and operational time constraints require different procedures for well test evaluations.

In the next chapters, a numerical solution for the radial semiconfined radial flow into a well is developed. The results are compared with some experiments described in this review, and also with analytical approaches.

3. Mathematical Model

In the following, a complete finite difference formulation of a well test in a gravity drainage unconfined reservoir is presented. The groundwater literature presents several attempts to develop analytical solutions to this problem, and all attempts produced equations that are applicable under limitations imposed by restrictive assumptions. One common constraint requires the water table gradient and/or the water table fall to be small. Therefore, high flow-rates should be avoided in pumping test analysis due to the resulting steep gradient around the wellbore.

3.1 Basic Assumptions

The present numerical model consists of a discretization in time and space of a cylindrical reservoir producing at constant flow rate under a gravity drainage drive from a centered, open hole wellbore. The model formulates two-dimensional flow and considers the height variation of the liquid column in the reservoir as a consequence of the free surface position change. Other important effects also considered in this model are wellbore storage, skin effect, and the variation of the direction of the flow at each position (r, z) in an anisotropic medium.

A single-phase flow model was developed using an adaptive grid based on a *stream-layer* concept described next. The following considerations for an unconfined aquifer flow will be assumed until otherwise modified:

- horizontal reservoir of finite, constant thickness
- anisotropy (k_h, k_v)
- Darcy's Law
- conservation of mass and energy
- compressible system containing a single liquid phase with constant properties
- uniform undisturbed free-surface as initial condition
- constant outer boundary pressure

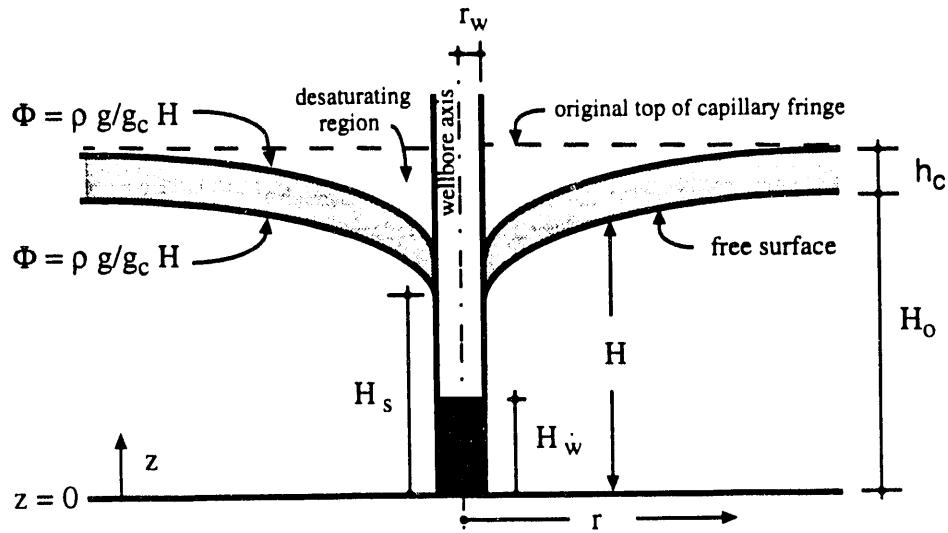


Figure 3.1: Representation of a cross section of an unconfined reservoir and definition of some coordinate variables.

- finite wellbore radius

3.2 Partial Differential Equations and Boundary Values

In a two-dimensional gravity flow system, several distinct regions with appropriate characteristics of flow may be defined: a saturated region where the porous space is completely filled by the liquid; a free liquid surface where the liquid table is in contact with gas at atmospheric pressure; a desaturated region left behind the free surface on drainage during production; the finite radius wellbore inner boundary; an impermeable lower boundary layer; a constant potential outer boundary which is far enough from the producing well to assure an infinite reservoir behavior. Also, a capillary fringe above the free surface is considered in a separate chapter. See Fig. 3.1 for the definition of some coordinate variables. For each one of these regions, a mathematical representation of the physical conditions are posed in the following.

- Saturated Region:

The diffusivity equation is applied to a compressible system completely saturated with a single liquid. From the combination of mass conservation and Darcy's Law principles:

$$\nabla \cdot \left[\frac{k}{\mu} \right] \nabla \Phi = \phi c_t \frac{\partial \Phi}{\partial t} \quad (3.2.1)$$

where the potential Φ is:

$$\Phi = p + \left(\rho \frac{g}{g_c} \right) z \quad (3.2.2)$$

- Free Surface Boundary:

The potential at the free surface is proportional to height. This is equivalent to atmospheric pressure defined as $p = 0$ everywhere on the free surface of height $z = H$:

$$\Phi(r, H) = \left(\rho \frac{g}{g_c} \right) H \quad (3.2.3)$$

If the capillary fringe is to be considered, the following equation is valid:

$$\Phi(r, H_c) = \Phi(r, H) = \left(\rho \frac{g}{g_c} \right) H \quad (3.2.4)$$

- Inner Boundary:

A constant flow rate inner boundary condition considering wellbore storage effects is obtained from a material balance at the well by the following equation:

$$qB_{wb} = -\pi r_w^2 \frac{\partial H_w}{\partial t} + \frac{2\pi r_w}{\mu} \int_0^{H_s} k \frac{\partial \Phi}{\partial r} \Big|_{r=r_w} dz \quad (3.2.5)$$

A complementary condition at the wellbore sandface must be defined. The potential at the sandface is specified by the observation of the existence of a seepage surface, characterizing the discontinuity of the positions of the free surface and the wellbore liquid level. At the inner boundary r_{w+} location, it is assumed that the potential is constant and equal to the potential at r_{w-} , when $z_w \leq H_w$. On the seepage surface ($H_s \geq z_w > H_w$), the potential is proportional to the vertical coordinate, since the point is on a stream-flow line at atmospheric pressure. Hence:

$$H_s \geq z_w > H_w \Rightarrow \Phi_w = \left(\rho \frac{g}{g_c} \right) z_w \quad (3.2.6)$$

$$z_w \leq H_w \Rightarrow \Phi_w = \left(\rho \frac{g}{g_c} \right) H_w \quad (3.2.7)$$

A detailed numerical treatment of the inner boundary condition is presented in Section 4.8.

- Infinite Reservoir Behavior:

Infinitely large reservoir behavior is obtained by assuming a large external radius. Therefore, a constant pressure outer boundary condition is used:

$$\Phi = \Phi_o = \left(\rho \frac{g}{g_c} \right) H_o \quad \Rightarrow \quad r \rightarrow \infty \quad (3.2.8)$$

- Lower Boundary:

A no-flow lower boundary condition is used:

$$\frac{\partial \Phi}{\partial z} = 0 \quad \Rightarrow \quad z = 0 \quad (3.2.9)$$

or

$$k_v = 0 \quad \Rightarrow \quad z = 0 \quad (3.2.10)$$

According to *Muskat* (1937) and several other authors, analytical solution of the system of partial differential equations in this boundary problem is of *insurmountable difficulty*. As seen in Ch. 2, not one of the analytical solutions surveyed was obtained without restrictive assumptions, and none appeared to represent pumping well pressure behavior. The numerical solution developed in the next chapter, although possessing the well known limitations of a finite difference method, is intended to present a tool to solve the gravity drainage well problem completely.

3.3 Dimensionless Parameters

Before modeling the numerical problem, a set of dimensionless constants and variables is required. The relationships presented use Darcy units for the reservoir parameters.

(a) Dimensionless length or thickness:

$$\begin{aligned} z_D &= \frac{z}{r_w}, & H_D &= \frac{H}{r_w}, & H_{wD} &= \frac{H_w}{r_w}, & H_{oD} &= \frac{H_o}{r_w} \\ r_D &= \frac{r}{r_w}, & r_{eD} &= \frac{r_e}{r_w} \end{aligned} \quad (3.3.11)$$

(b) Dimensionless pressure and potential:

$$p_D = \frac{p}{\rho \frac{g}{g_c} H_o} \quad (3.3.12)$$

$$\Phi_D = W = \frac{\Phi}{\rho \frac{g}{g_c} H_o} \quad (3.3.13)$$

(c) Dimensionless static liquid gradient:

$$\gamma_o = \frac{1}{H_{oD}} \quad (3.3.14)$$

(d) Dimensionless production rate:

$$q_D = \frac{q \mu}{2 \pi r_w k_h \rho \frac{g}{g_c} H_o} \quad (3.3.15)$$

(e) Dimensionless time:

$$t_D = \frac{k_h \rho \frac{g}{g_c} H_o}{\phi \mu r_w^2} t \quad (3.3.16)$$

(f) Dimensionless liquid and total compressibilities:

$$\begin{aligned} c_{L_D} &= \rho \frac{g}{g_c} H_o c_L \\ c_{t_D} &= \rho \frac{g}{g_c} H_o c_t \end{aligned} \quad (3.3.17)$$

In the next chapter, a finite difference model is considered in detail.

4. Finite Difference Model

The partial differential equation that represents radial liquid flow in the reservoir may be generated from a material balance in an infinitesimal element of porous medium, based on the conservation of mass and *Darcys* flow law. The complexities of the boundary conditions in the free surface gravity flow problem drive the present solution to a numerical approach, making it possible to linearize the boundary equations. The discretized form of these equations can be formulated from the same material balance criteria using a finite-difference formulation described in this chapter.

Figure 4.1 represents a grid configuration of the problem, depicting a half cross-section of a cylindrical reservoir with a free-surface developed from the original position H_o . The horizontal dimensions of the grid blocks can be scaled to provide accuracy around the well bore, where the most important flow effects occur. The radial grid spacing used the *Multimodal* method [Terán and *de la Garza*, 1988], in which logarithmic spacing is a particular case.

In order to simulate the variable free surface position, an adaptive vertical grid was conceived to group a set of streamlines, and referred to as the *Stream Layer Model* (SLM) in the following. During a discrete time step, each block but the one containing the free surface will keep its geometrical dimensions unchanged while the liquid flows through the block boundaries in the vertical and axial directions. In this cross sectional model, no angular flow normal to the cross section is allowed, and the axial directions are not necessarily horizontal, as can be seen in Fig. 4.1. The dimensions normal to the axial flow are corrected by the average slopes of the block boundaries, according to Fig. 4.2.

In a separate section of this chapter, the capillary fringe is added to the model by incorporating an additional layer with constant height on the top of the free surface layer, or phreatic layer. Capillary flow is important for small scale sand-box experiments where capillarity has a sensitive role in the results.

In the present model, the free surface layer is the only one that changes position during a production or buildup time step period, because the free surface layer contains the movable boundary. In the case of neglecting the capillary region above the free surface, the free surface downward movement leaves behind a liquid saturation in the desaturating zone. In the present model, no later flow from that region is allowed, and the liquid moves entirely

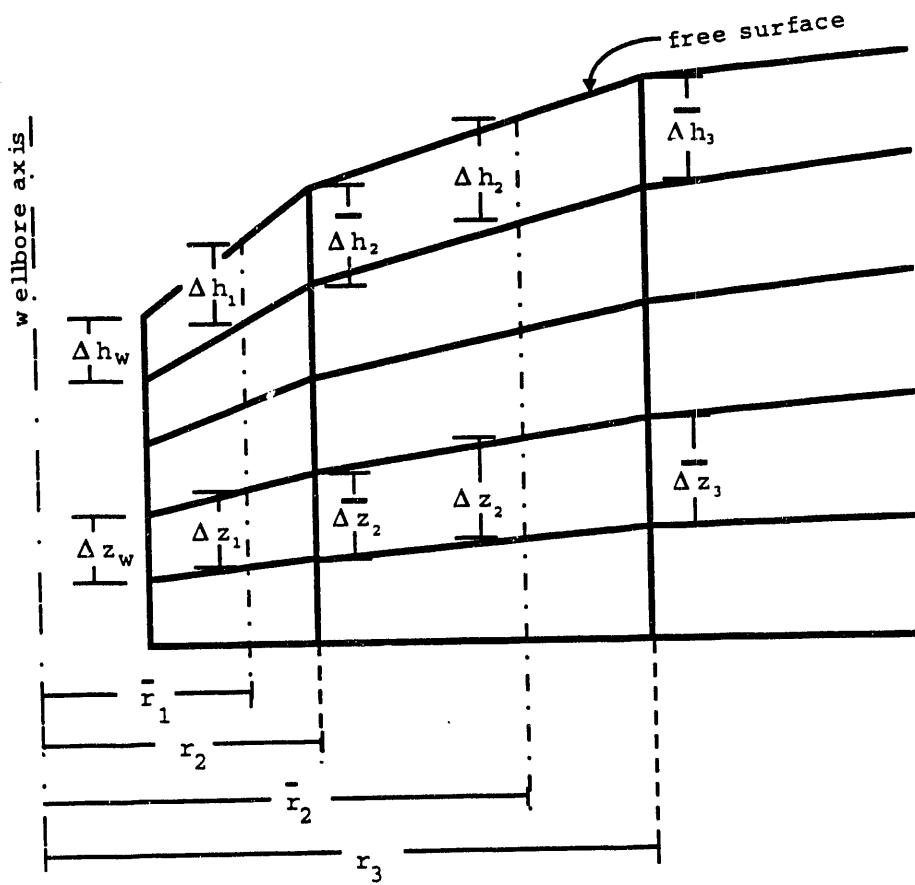


Figure 4.1: The *Stream Layer Model* grid

within a specific period. The liquid saturation behind the free surface boundary is considered residual. Based on groundwater studies - see *Bouwer and Rice* (1978) and *Neuman* (1979), the air entry effects in the desaturating region (two-phase flow) are neglected.

The technique used in calculations was the iterative solution of the block potentials, concentrating the nonlinearities in the free surface and the wellbore potentials where the boundary values are estimated in advance based on the previous time step. The convergence method was the Newton-Raphson procedure which generates a diagonal Jacobian matrix with special characteristics due to the non-zero coefficients corresponding to the inner boundary blocks, as shown in Fig. 4.3.

A single step solution follows the sequence:

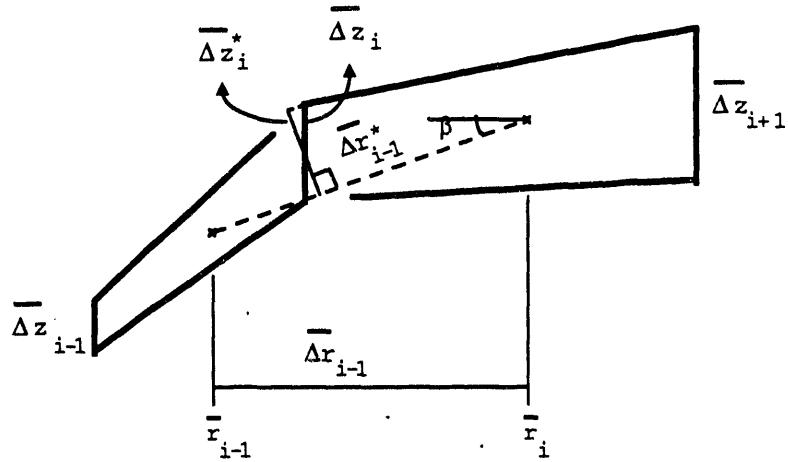


Figure 4.2: Geometric corrections based in the slope of the flow axis.

- (a) Calculate the inner boundary potential as a first approach
- (b) Calculate the equation coefficients
- (c) Calculate the Jacobian matrix elements
- (d) Solve the matrix for the potentials
- (e) Recalculate the inner boundary potential
- (f) Check the results for convergence and iterate if necessary

Since the layer containing the free surface is the only one moving during a discrete time period, the block thicknesses are dependent on the free surface position, or dependent on the block potentials. This loop dependency is responsible for the strong nonlinearity of the gravity drainage problem.

Skin effect was considered in the model by assuming a cylindrical damaged region around the well with an altered permeability, using *Hawkins* (1956) approach:

$$s = \left(\frac{k}{k_d} - 1 \right) \ln \left(\frac{r_d}{r_w} \right) \quad (4.1)$$

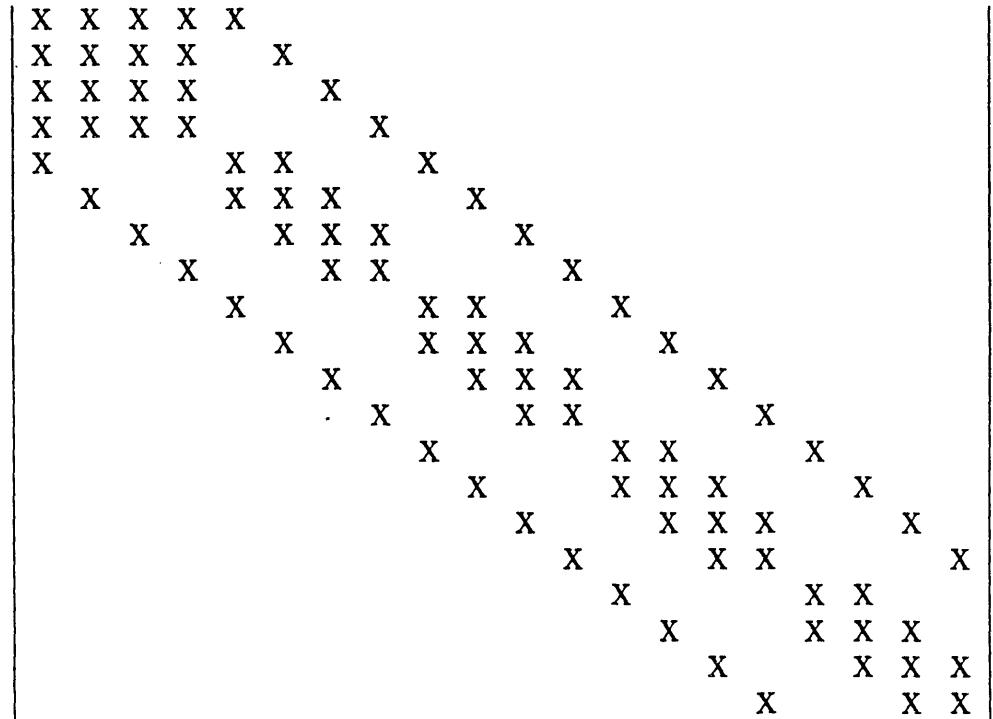


Figure 4.3: Matrix representation of a 5x4 cross sectional grid

The finite difference treatment of the equations and the boundary conditions is presented in the sections that follow. A detailed description of the computer model is given in the Sec. 4.9.

4.1 Grid Distribution

The cross-sectional model is composed of a grid of $M \times N$ radial and vertical blocks, numbered from the left to the right, and from the top to the bottom. The uppermost layer corresponding to the position $j = 1$ contains the liquid-free surface. The inner blocks are set at $i = 1$. Thus, the number of radial blocks is M , not considering the wellbore and the external outer blocks with constant potential.

The radial grid distribution is a special kind of logarithmic spacing using the *multimodal* method of discretization adopting the the block-centered scheme. In this method,

the problems of stability are reduced by controlling the radial size of the blocks by fixing a ratio R between the inner and the outer blocks. Hence, the critical conditions of potential changes in the inner blocks, those containing the sandface, are decreased, reducing the stability problems that commonly happen in such radial flow models at late times. The usual logarithmic scale is a particular case of the *multimodal* method.

In the cross-sectional view, the blocks embrace sets of streamlines with variable slope depending on the block position and the time. However, the vertical flow between consecutive layers allows some flow lines to move from a layer set to an adjacent layer set during a period of time Δt . The stream layer is a geometric location containing the streamlines moving through it. The only moving boundary within that time period will exactly be the top of the first (uppermost) layer which contains the free surface, while all other layers have the same uniform thickness over all time steps.

The time discretization uses a variable parameter θ between 0 and 1, assigning a variable degree of implicitness to the model. The nature of the gravity problem requires analysis of ideal θ values, which is done in the model sensitivity analysis.

The coordinates are named in the following way, according to the figures presented in this chapter:

r_i = radial distance of the inner limit of the block i, j to the wellbore axis

\bar{r}_i = radial distance of the block node to the wellbore axis ¹

$z_{i,j}$ = vertical coordinate of the lower block i, j lower boundary at \bar{r}_i , taken from the base of the formation

Δz_i = vertically uniform thickness of the block i, j at position \bar{r}_i

$\overline{\Delta z_i}$ = thickness of the block i, j at the position r_i

H_i = height of the free surface at the position \bar{r}_i

Δh_i = equivalent to Δz_i , referred to the free surface layer blocks

$\overline{\Delta h_i}$ = equivalent to $\overline{\Delta z_i}$, referred to the position r_i

H_o = initial position of the free surface at rest, from the impermeable base

H_w = liquid level in the well

¹In the *MULTIMODAL* method, both r_i and \bar{r}_i are calculated implicitly - see [Terán and de la Garza, 1988].

H_s = liquid level in the sandface

$z_{w,j}$ = lower inner block boundary position j at the sandface

$\Delta z_{w,j}$ = inner block boundary thickness at the sandface

$\Delta h_{w,j}$ = equivalent to $\Delta z_{w,j}$ referred to the free surface layer inner block

Figure 4.2 shows a special representation of a grid block (i, j) at a time t , containing a set of streamlines and an adjacent block. Due to increased slopes of the streamlines around the well, the geometric projections of the lengths and heights are based on angles calculated on the following basis:

$$\beta_{i,j} = \operatorname{tg}^{-1} \left(\frac{z_{i,j} + \frac{1}{2}\Delta z_{i,j} - (z_{i-1,j} + \frac{1}{2}\Delta z_{i-1,j})}{\bar{r}_i - \bar{r}_{i-1}} \right), \quad (4.1.1)$$

and the average slope of an individual block:

$$\bar{\beta}_{i,j} = \operatorname{tg}^{-1} \left(\frac{z_{i+1,j} + \frac{1}{2}\Delta z_{i+1,j} - (z_{i-1,j} + \frac{1}{2}\Delta z_{i-1,j})}{\bar{r}_{i+1} - \bar{r}_{i-1}} \right). \quad (4.1.2)$$

The flow domain is divided into two main directions of flow, namely the vertical (subscript v), and the axial (quasi-radial) (subscript x). The mathematical description of such a flow can be formulated through the general finite difference equation for a single liquid phase flowing in a low compressibility system:

$$\Delta_x(T_x \Delta_x \Phi) + \Delta_v(T_v \Delta_v \Phi) = \frac{1}{\Delta t} \Delta_t(\phi V) \quad (4.1.3)$$

In gravity drainage flow, the direction of the streamlines change continuously, and for that reason, the true transmissivity considers the geometry and also the permeability tensor. In this work where anisotropy is considered, the pseudoaxial permeability (in the streamline direction) is taken as an elliptic function of the horizontal (radial) and vertical permeabilities, according to the rule:

$$k_{x_{i,j}} = \frac{k_h k_v}{k_h \sin \beta_{i,j} + k_v \cos \beta_{i,j}} \quad (4.1.4)$$

Each of the boundary conditions will define individual treatment for the transmissivity terms, as well as the storage variation. In the next sections of this chapter, Eq. 4.1.3 will be expanded and applied to the saturated region and the boundaries.

4.2 Balance of Unknowns and Equations

Number of unknowns:

Reservoir block potentials	$M \times N$
Free surface positions	M
Position of the fluid level in wellbore	1
Total unknowns:	$M(N + 1) + 1$

Number of equations:

Material balance for the blocks	$M \times N$
Free surface position	M
Wellbore material balance	1
Total equations:	$M(N + 1) + 1$

4.3 Saturated Region

Equation 4.1.3 can be written for the saturated region and the inner and outer boundaries.

4.3.1 Discretization of the Saturated Region Equations

Discretization of a block i, j , for $M > i > 1$, and $N > j > 2$:

$$\Delta_x (T_x \Delta_x \Phi) = T_{x_{i+1,j}} (\Phi_{i+1,j} - \Phi_{i,j}) - T_{x_{i,j}} (\Phi_{i,j} - \Phi_{i-1,j}), \quad (4.3.1)$$

$$\Delta_v (T_v \Delta_v \Phi) = T_{v_{i,j-1}} (\Phi_{i,j-1} - \Phi_{i,j}) - T_{v_{i,j}} (\Phi_{i,j} - \Phi_{i,j+1}), \quad (4.3.2)$$

and:

$$\frac{1}{\Delta t} \Delta_t (\phi V) = \frac{A_{v_i} \Delta z_i \phi c_t}{\Delta t \cos \bar{\beta}_{i,j}} \Delta_t \Phi_{i,j} \quad (4.3.3)$$

where the typical transmissivity terms are:

$$T_{x_{i,j}} = \frac{2\pi}{\mu} \frac{k_{x_{i,j}}}{\ln \left(\frac{\bar{r}_i}{\bar{r}_{i-1}} \right)} \overline{\Delta z}_i \cos \beta_{i,j} \quad (4.3.4)$$

$$T_{v_{i,j}} = \frac{A_{v_i}}{\mu} \frac{k_{v_{i,j}}}{\Delta z_i} \quad (4.3.5)$$

The time interpolation used a parameter θ to weight the old unknowns and the unknowns within a time step. Thus, using the dimensionless parameters defined in the Ch. 3.3, re-organizing, and collecting terms, the resulting equation may be summarized in the following way:

$$A_{i,j} W_{i,j} + B_{i,j} W_{i,j-1} + C_{i,j} W_{i,j+1} + D_{i,j} W_{i-1,j} + E_{i,j} W_{i+1,j} = F_{i,j} \quad (4.3.6)$$

where the coefficients of the unknown dimensionless potentials and the independent term are:

$$A_{i,j} = -\theta \left[\overline{\Delta z}_{D_{i+1}} + f_{x_{i,j}} f_{r_i} \overline{\Delta z}_{D_i} + \frac{2k_v}{k_{x_{i+1,j}} \cos \beta_{i+1,j}} \frac{A_{1_i}}{\Delta z_{D_i}} \right. \\ \left. + \frac{k_h}{k_{x_{i+1,j}} \cos \beta_{i+1,j}} \frac{A_{1_i} \Delta z_{D_i}}{\theta \cos \bar{\beta}_{i,j} \Delta t_D} c_{t_D} \right] \quad (4.3.7)$$

$$B_{i,j} = \frac{\theta k_v A_{1_i}}{k_{x_{i+1,j}} \cos \beta_{i+1,j} \Delta z_{D_i}} \quad (4.3.8)$$

$$C_{i,j} = \frac{\theta k_v A_{1_i}}{k_{x_{i+1,j}} \cos \beta_{i+1,j} \Delta z_{D_i}} \quad (4.3.9)$$

$$D_{i,j} = \theta f_{x_{i,j}} f_{r_i} \overline{\Delta z}_i \quad (4.3.10)$$

$$E_{i,j} = \theta \overline{\Delta z}_{i+1} \quad (4.3.11)$$

$$\begin{aligned}
F_{i,j} = & - \frac{k_h}{k_{x_{i+1,j}} \cos \beta_{i+1,j}} \frac{A_{1,i} \Delta z_{D_i}}{\cos \bar{\beta}_{i,j} \Delta t_D} c_{t_D} W_{i,j}^k - (1-\theta) \times \\
& \left\{ \overline{\Delta z}_{D,i+1} [W_{i+1,j}^k - W_{i,j}^k] - f_{x_{i,j}} f_{r_i} \overline{\Delta z}_{D_i} [W_{i,j}^k - W_{i-1,j}^k] \right. \\
& \left. + \frac{k_v A_{1,i}^*}{k_{x_{i+1,j}} \cos \beta_{i+1,j} \Delta z_{D_i}} [W_{i,j-1}^k - 2W_{i,j}^k + W_{i,j+1}^k] \right\} \quad (4.3.12)
\end{aligned}$$

In these coefficients, the superscript k refers to the beginning of the present time step values. Other parameters are:

$$A_{1,i} = \frac{A_{v_i} \ln \left(\frac{\bar{r}_{i+1}}{\bar{r}_i} \right)}{2\pi r_w^2} \quad (4.3.13)$$

$$f_{x_{i,j}} = \frac{k_{x_{i,j}} \cos \beta_{i,j}}{k_{x_{i+1,j}} \cos \beta_{i+1,j}} \quad (4.3.14)$$

$$f_{r_i} = \frac{\ln \left(\frac{\bar{r}_{i+1}}{\bar{r}_i} \right)}{\ln \left(\frac{\bar{r}_i}{\bar{r}_{i-1}} \right)} \quad (4.3.15)$$

The values of $\overline{\Delta z}_D$ are estimated by interpolation between nodes.

The Jacobian matrix elements for the block i,j were obtained using a Newton-Raphson procedure.

The residue Ψ is:

$$\Psi_{i,j} = A_{i,j} W_{i,j} + B_{i,j} W_{i,j-1} + C_{i,j} W_{i,j+1} + D_{i,j} W_{i-1,j} + E_{i,j} W_{i+1,j} - F_{i,j} \quad (4.3.16)$$

Differentiating Eq. 4.3.16 with respect to each variable W :

$$\begin{aligned}
DA_{i,j} = \frac{\partial \Psi_{i,j}}{\partial W_{i,j}} = & A_{i,j} + \frac{\partial A_{i,j}}{\partial W_{i,j}} W_{i,j} + \frac{\partial B_{i,j}}{\partial W_{i,j}} W_{i,j-1} + \frac{\partial C_{i,j}}{\partial W_{i,j}} W_{i,j+1} \\
& \frac{\partial D_{i,j}}{\partial W_{i,j}} W_{i-1,j} + \frac{\partial E_{i,j}}{\partial W_{i,j}} W_{i+1,j} - \frac{\partial F_{i,j}}{\partial W_{i,j}} \quad (4.3.17)
\end{aligned}$$

Since all the partial differentials are zero:

$$DA_{i,j} = A_{i,j} \quad (4.3.18)$$

Similarly, other Jacobian coefficients are:

$$DB_{i,j} = \frac{\partial \Psi_{i,j}}{\partial W_{i,j-1}} = B_{i,j} \quad (4.3.19)$$

$$DC_{i,j} = \frac{\partial \Psi_{i,j}}{\partial W_{i,j+1}} = C_{i,j} \quad (4.3.20)$$

$$DD_{i,j} = \frac{\partial \Psi_{i,j}}{\partial W_{i-1,j}} = D_{i,j} \quad (4.3.21)$$

$$DE_{i,j} = \frac{\partial \Psi_{i,j}}{\partial W_{i+1,j}} = E_{i,j} \quad (4.3.22)$$

4.3.2 Saturated Region Inner Boundary Equations

The blocks containing the sandface require a special treatment because of their location. The transmissivity connected to the sandface can be estimated by geometric extrapolation of thicknesses and slopes of the layers. The inner boundary potentials are determined by the wellbore condition, as shown in the Section 4.8. Figure 4.1 presents a schematic of the inner blocks around the wellbore and the extrapolated thicknesses. The average thickness of a block flowing to the wellbore is taken half-way between the the node and the inner face. If small radial spacing is used near the well, the geometric extrapolation can be relaxed, and a constant thickness can be adopted for the whole block and its inner boundary. To be general, the equations presented below used extrapolated thicknesses.

We now consider discretization of block $1,j$ for $N > j > 2$. Following the same procedures as before and considering the proper symbology for the inner boundary parameters, the resulting equation is:

$$A_{1,j} W_{1,j} + B_{1,j} W_{1,j-1} + C_{1,j} W_{1,j+1} + E_{1,j} W_{2,j} = F_{1,j} \quad (4.3.23)$$

where the coefficients of the unknown dimensionless potentials and the independent term are:

$$\begin{aligned}
A_{1,j} &= -\theta \left[\overline{\Delta z}_{D_2} + f_{x_{1,j}} f_{r_1} \overline{\Delta z}_{D_1} + \frac{2k_v}{k_{x_{2,j}} \cos \beta_{2,j}} \frac{A_{11}}{\Delta z_{D_1}} \right. \\
&\quad \left. + \frac{k_h}{k_{x_{2,j}} \cos \beta_{2,j}} \frac{A_{11} \Delta z_{D_1}}{\theta \cos \bar{\beta}_{1,j} \Delta t_D} c_{t_D} \right] \quad (4.3.24)
\end{aligned}$$

$$B_{1,j} = \frac{\theta k_v A_{11}}{k_{x_{2,j}} \cos \beta_{2,j} \Delta z_{D_1}} \quad (4.3.25)$$

$$C_{1,j} = \frac{\theta k_v A_{11}}{k_{x_{2,j}} \cos \beta_{2,j} \Delta z_{D_1}} \quad (4.3.26)$$

$$D_{1,j} = 0 \quad (4.3.27)$$

$$E_{1,j} = \theta \overline{\Delta z}_2 \quad (4.3.28)$$

$$\begin{aligned}
F_{1,j} &= -f_{x_{1,j}} f_{r_1} \overline{\Delta z}_{D_1} \left\{ \theta W_{w,j} - (1-\theta) \left[W_{1,j}^k - W_{w,j}^k \right] \right\} \\
&\quad - \frac{k_h}{k_{x_{2,j}} \cos \beta_{2,j}} \frac{A_{11} \Delta z_{D_1}}{\Delta t_D \cos \bar{\beta}_{1,j}} c_{t_D} W_{1,j}^k - (1-\theta) \left\{ \overline{\Delta z}_{D_2} (W_{2,j}^k - W_{1,j}^k) \right. \\
&\quad \left. + \frac{k_v}{k_{x_{2,j}} \cos \beta_{2,j}} \frac{A_{11}}{\Delta z_{D_1}} \left[-2W_{1,j}^k + W_{1,j-1}^k + W_{1,j+1}^k \right] \right\}. \quad (4.3.29)
\end{aligned}$$

In these coefficients, the superscript k refers to the previous time step value. Other parameters are:

$$A_{11} = \frac{A_{v_1} \ln \left(\frac{\bar{r}_2}{\bar{r}_1} \right)}{2\pi r_w^2} \quad (4.3.30)$$

$$f_{x_{1,j}} = \frac{k_{x_{1,j}} \cos \beta_{1,j}}{k_{x_{2,j}} \cos \beta_{2,j}} \quad (4.3.31)$$

$$f_{r_1} = \frac{\ln \left(\frac{\bar{r}_2}{\bar{r}_1} \right)}{\ln \left(\frac{\bar{r}_1}{\bar{r}_w} \right)} \quad (4.3.32)$$

Determine Jacobian matrix elements for the block $1, j$ using a Newton-Raphson procedure. The residue Ψ is:

$$\Psi_{1,j} = A_{1,j} W_{1,j} + B_{1,j} W_{1,j-1} + C_{1,j} W_{1,j+1} + E_{1,j} W_{2,j} - F_{1,j} \quad (4.3.33)$$

Differentiating Eq. 4.3.33 with respect to each variable $W_{i,j}$:

$$\begin{aligned} DA_{1,j} = \frac{\partial \Psi_{1,j}}{\partial W_{1,j}} &= A_{1,j} + \frac{\partial A_{1,j}}{\partial W_{1,j}} W_{1,j} + \frac{\partial B_{1,j}}{\partial W_{1,j}} W_{1,j-1} + \frac{\partial C_{1,j}}{\partial W_{1,j}} W_{1,j+1} \\ &+ \frac{\partial E_{1,j}}{\partial W_{1,j}} W_{2,j} - \frac{\partial F_{1,j}}{\partial W_{1,j}} \end{aligned} \quad (4.3.34)$$

Except the last one, all the partial differentials in Eq. 4.3.34 are zero. Thus:

$$DA_{i,j} = A_{i,j} - \frac{\partial F_{1,j}}{\partial W_{1,j}} = A_{i,j} + \theta f_{x_{1,j}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_j}}{\partial W_{1,j}} \quad (4.3.35)$$

Similarly, the other Jacobian coefficients are:

$$DB_{1,j} = \frac{\partial \Psi_{1,j}}{\partial W_{1,j-1}} = B_{1,j} + \theta f_{x_{1,j}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_j}}{\partial W_{1,j-1}} \quad (4.3.36)$$

$$DC_{1,j} = \frac{\partial \Psi_{1,j}}{\partial W_{1,j+1}} = C_{1,j} + \theta f_{x_{1,j}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_j}}{\partial W_{1,j+1}} \quad (4.3.37)$$

$$DD_{1,j} = 0 \quad (4.3.38)$$

$$DE_{1,j} = \frac{\partial \Psi_{1,j}}{\partial W_{2,j}} = E_{1,j} \quad (4.3.39)$$

The partial differentiations of the inner boundary potentials (see App. A.1), calculated individually, may be substituted above.

4.3.3 Saturated Region Outer Boundary Blocks

The outer boundary condition is a constant potential at the external radius. The index m is used to refer to the M^{th} block, and $m + 1$ is a dummy external block with constant potential and geometry.

Now we consider discretization of the block m, j , for $N > j > 2$. Following the same procedures and considering the proper symbology for the outer boundary parameters, the resulting equation is:

$$A_{m,j} W_{m,j} + B_{m,j} W_{m,j-1} + C_{m,j} W_{m,j+1} + \dots + D_{m,j} W_{m-1,j} = F_{m,j} \quad (4.3.40)$$

where the coefficients of the unknown dimensionless potentials and the independent term are:

$$\begin{aligned} A_{m,j} &= -\theta \left[\overline{\Delta z}_{D_{m+1}} + f_{x_{m,j}} f_{r_m} \overline{\Delta z}_{D_m} + \frac{2k_v}{k_h} \frac{A_{1m}}{\Delta z_{D_m}} \right. \\ &\quad \left. + \frac{A_{1m} \Delta z_{D_m}}{\theta \cos \beta_{m,j} \Delta t_D} c_{t_D} \right] \end{aligned} \quad (4.3.41)$$

$$B_{m,j} = \frac{\theta k_v A_{1m}}{k_h \Delta z_{D_m}} \quad (4.3.42)$$

$$C_{m,j} = \frac{\theta k_v A_{1m}}{k_h \Delta z_{D_m}} \quad (4.3.43)$$

$$D_{m,j} = \theta f_{x_{m,j}} f_{r_m} \overline{\Delta z}_{D_m} \quad (4.3.44)$$

$$\begin{aligned} F_{m,j} &= -\frac{A_{1m} \Delta z_{D_m}}{\Delta t_D \cos \beta_{m,j}} c_{t_D} W_{m,j}^k - \overline{\Delta z}_{D_{m+1}} \left[1 - (1 - \theta) W_{m,j}^k \right] - \\ &\quad (1 - \theta) \left\{ -f_{x_{m,j}} f_{r_m} \overline{\Delta z}_{D_m} \left[W_{m,j}^k - W_{m-1,j}^k \right] \right. \\ &\quad \left. + \frac{k_v A_{1m}}{k_h \Delta z_{D_m}} \left[W_{m,j-1}^k - 2W_{m,j}^k + W_{m,j-1}^k \right] \right\} \end{aligned} \quad (4.3.45)$$

At the external boundary, the flux is always normal to the external face, and so, $\cos \beta_{m+1,j} =$

1. Thus, other parameters are:

$$A_{1m} = \frac{A_{vm} \ln \left(\frac{\bar{r}_{m+1}}{\bar{r}_m} \right)}{2\pi r_w^2} \quad (4.3.46)$$

$$f_{x_{m,j}} = \frac{k_{x_{m,j}} \cos \beta_{m,j}}{k_h} \quad (4.3.47)$$

$$f_{r_m} = \frac{\ln \left(\frac{\bar{r}_{m+1}}{\bar{r}_m} \right)}{\ln \left(\frac{\bar{r}_m}{\bar{r}_{m-1}} \right)} \quad (4.3.48)$$

Determine the Jacobian matrix elements for the block m, j using a Newton-Raphson procedure. The residue Ψ is:

$$\Psi_{m,j} = A_{m,j} W_{m,j} + B_{m,j} W_{m,j-1} + C_{m,j} W_{m,j+1} + D_{m,j} W_{m-1,j} - F_{m,j} \quad (4.3.49)$$

Differentiating Eq. 4.3.49 with respect to each variable $W_{m,j}$:

$$DA_{m,j} = \frac{\partial \Psi_{m,j}}{\partial W_{m,j}} = A_{m,j} \quad (4.3.50)$$

$$DB_{m,j} = \frac{\partial \Psi_{m,j}}{\partial W_{m,j-1}} = B_{m,j} \quad (4.3.51)$$

$$DC_{m,j} = \frac{\partial \Psi_{m,j}}{\partial W_{m,j+1}} = C_{m,j} \quad (4.3.52)$$

$$DD_{m,j} = \frac{\partial \Psi_{m,j}}{\partial W_{m-1,j+1}} = D_{m,j} \quad (4.3.53)$$

$$DE_{m,j} = 0 \quad (4.3.54)$$

4.4 Free Surface Blocks

The free surface blocks are those containing the free moving boundary on top of them. In the present discretization, the capillary fringe is neglected and the no-flow upper condition is applied. The free surface block thicknesses are the unknowns that cause the nonlinearity in the problem and are calculated by an iterative process.

4.4.1 Free Surface Layer Equations

The principle of mass conservation associated with Darcy's Law applied to an element of reservoir containing the free surface was written in Darcy units according to the conventions adopted in Fig. 4.4, which shows a representation of a block element containing the free surface. In this study, the directions of flow and the transmissivities are ruled by the streamline slopes. That characteristic of the model enables one to consider the axial transmissivity terms as a function of the the direction of flow normal to the block boundaries, similarly to the saturated region equations.

The condition that the components of velocity normal to the free surface are null means there is no external flow through the free surface, neglecting the unsaturated flow. Taking that observation and the condition given by Eq. 3.2.3, the expansion of Eq. 4.1.3 is described in the next sections.

Discretization of a block $i, 1$, for $M > i > 1$:

$$\Delta_x (T_x \Delta_x \Phi) = T_{x_{i+1,1}} (\Phi_{i+1,1} - \Phi_{i,1}) - T_{x_{i,1}} (\Phi_{i,1} - \Phi_{i-1,1}) \quad (4.4.1)$$

$$\Delta_v (T_v \Delta_v \Phi) = - T_{v_{i,1}} (\Phi_{i,1} - \Phi_{i,2}) \quad (4.4.2)$$

$$\frac{1}{\Delta t} \Delta_t (\phi V) = \frac{A_{v_i} \phi}{\Delta t \cos \beta_{i,1}} \left[S_g (\Delta h_i - \Delta h_i^k) + \left(\frac{\Delta h_i + \Delta h_i^k}{2} \right) c_t \Delta_t \Phi \right] \quad (4.4.3)$$

Typical transmissivity terms are:

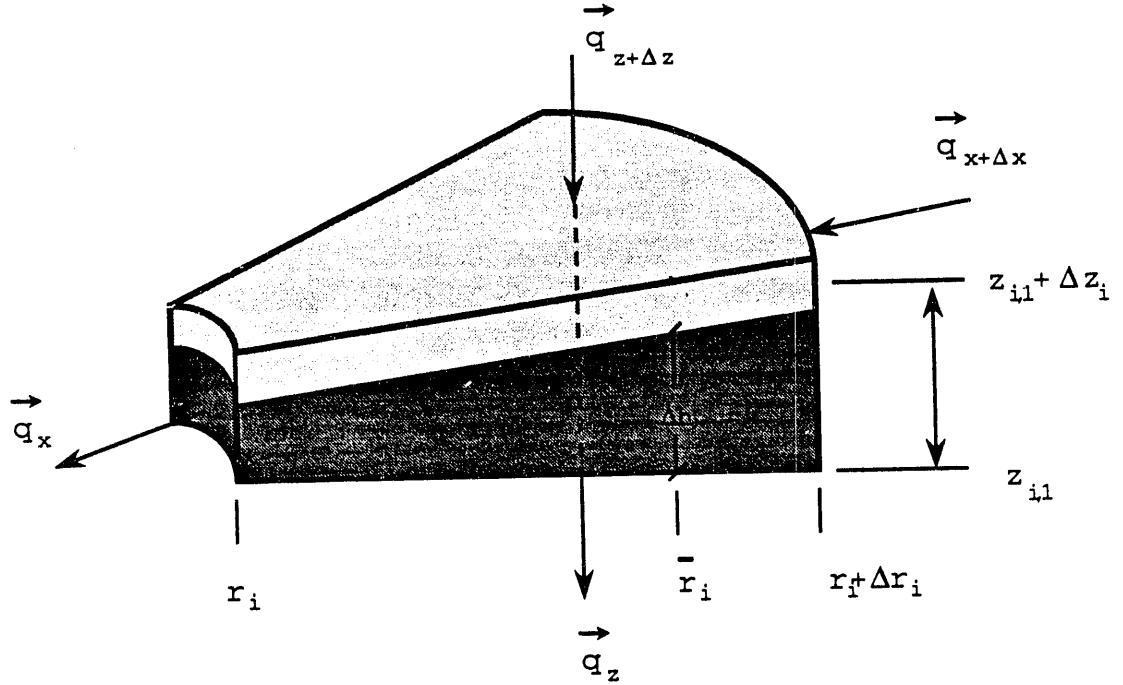


Figure 4.4: Representation of a block containing the free surface. Convention of flow and coordinate directions.

$$T_{x_{i,1}} = \frac{2\pi}{\mu} \frac{k_{x_{i,1}}}{\ln\left(\frac{\bar{r}_i}{\bar{r}_{i-1}}\right)} \overline{\Delta z_i} \cos \beta_{i,1} \quad (4.4.4)$$

$$T_{v_{i,1}} = \frac{A_{v_i}}{\mu} \frac{2k_v}{\Delta z_i + \Delta h_i} \quad (4.4.5)$$

The usual procedures to develop the discretized equations taking the time relaxation parameter θ is synthesized by the general equation:

$$A_{i,1} W_{i,1} + C_{i,1} W_{i,2} + D_{i,1} W_{i-1,1} + E_{i,1} W_{i+1,1} = F_{i,1} \quad (4.4.6)$$

where the coefficients of the unknown dimensionless potentials and the independent term are:

$$A_{i,1} = -\theta \left[\overline{\Delta h}_{D_{i+1}} + f_{x_{i,1}} f_{r_i} \overline{\Delta h}_{D_i} + \frac{2k_v}{k_{x_{i+1,1}} \cos \beta_{i+1,1}} \frac{A_{1,i}}{\Delta z_{D_i} + \Delta h_{D_i}} \right]$$

$$+ \frac{k_h A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1}} \frac{\Delta h_{D_i} + \Delta z_{D_i}}{2\theta \cos \bar{\beta}_{i,1} \Delta t_D} c_{t_D} \Big] \quad (4.4.7)$$

$$B_{i,1} = 0 \quad (4.4.8)$$

$$C_{i,1} = \frac{2\theta k_v A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1} (\Delta z_{D_i} + \Delta h_{D_i})} \quad (4.4.9)$$

$$D_{i,1} = \theta f_{x_{i,1}} f_{r_i} \bar{\Delta h}_{D_i} \quad (4.4.10)$$

$$E_{i,1} = \theta \bar{\Delta h}_{D_{i+1}} \quad (4.4.11)$$

$$\begin{aligned} F_{i,j} = & - \frac{k_h}{k_{x_{i+1,1}} \cos \beta_{i+1,1} \cos \bar{\beta}_{i,1} \Delta t_D} \left[(\Delta h_{D_i} - \Delta h_{D_i}^k) S_g - \left(\frac{\Delta h_{D_i} + \Delta h_{D_i}^k}{2} \right) c_{t_D} W_{i,1}^k \right] \\ & - (1 - \theta) \left\{ \bar{\Delta h}_{D_{i+1}}^k [W_{i+1,1}^k - W_{i,1}^k] - f_{x_{i,1}} f_{r_i} \bar{\Delta h}_{D_i} [W_{i,1}^k - W_{i-1,1}^k] \right. \\ & \left. + \frac{2 k_v A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1} (\Delta z_{D_i} + \Delta h_{D_i})} [-W_{i,1}^k + W_{i,2}^k] \right\} \quad (4.4.12) \end{aligned}$$

These definitions of $A_{1,i}$, $f_{x_{i,1}}$, and f_{r_i} are the same as given by Eqs. 4.3.13 through 4.3.15. Determine the Jacobian matrix elements for the block $i,1$ using a Newton-Raphson procedure. Let Ψ be defined as the residue:

$$\Psi_{i,1} = A_{i,1} W_{i,1} + C_{i,1} W_{i,2} + D_{i,1} W_{i-1,1} + E_{i,1} W_{i+1,1} - F_{i,1} \quad (4.4.13)$$

The Jacobian elements are summarized in the following:

$$\begin{aligned} DA_{i,1} = \frac{\partial \Psi_{i,1}}{\partial W_{i,2}} = & A_{i,1} - \left[\frac{\theta (1 - \alpha_{i+1} + \alpha_i f_{x_{i,1}} f_{r_i})}{\bar{\gamma}_i} \right. \\ & - \frac{C_{i,1}}{(\Delta h_{D_i} + \Delta z_{D_i}) \bar{\gamma}_i} + \frac{k_h A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1}} \frac{c_{t_D}}{2\bar{\gamma}_i \Delta t_D \cos \bar{\beta}_{i,1}} \Big] W_{i,1} \\ & - \frac{C_{i,1}}{(\Delta h_{D_i} + \Delta z_{D_i}) \bar{\gamma}_i} W_{i,2} + \frac{\theta (f_{x_{i,1}} f_{r_i}) \alpha_i}{\bar{\gamma}_i} W_{i-1,1} + \frac{\theta (1 - \alpha_{i+1})}{\bar{\gamma}_i} W_{i+1,1} \\ & \left. - \frac{k_h}{k_{x_{i+1,1}} \cos \beta_{i,1}} \frac{A_{1,i}}{\Delta t_D \cos \bar{\beta}_{i,1}} \frac{c_{t_D} W_{i,1}^k}{2\bar{\gamma}_i} \right] \quad (4.4.14) \end{aligned}$$

$$DB_{i,1} = 0 \quad (4.4.15)$$

$$DC_{i,1} = \frac{\partial \Psi_{i,1}}{\partial W_{i,2}} = C_{i,1} \quad (4.4.16)$$

$$DD_{i,1} = \frac{\partial \Psi_{i,1}}{\partial W_{i-1,1}} = D_{i,1} + \frac{\theta(1-\alpha_i) f_{x_{i,1}} f_{r_i}}{\bar{\gamma}_{i-1}} [W_{i-1,1} - W_{i,1}] \quad (4.4.17)$$

$$DE_{i,1} = \frac{\partial \Psi_{i,1}}{\partial W_{i+1,1}} = E_{i,1} + \frac{\theta \alpha_{i+1}}{\bar{\gamma}_{i+1}} [W_{i+1,1} - W_{i,1}] \quad (4.4.18)$$

In these expressions, some parameters were introduced such as the dimensionless dynamic average pressure gradient $\bar{\gamma}_i$ which can be approximated by the dimensionless static liquid gradient, since the vertical grid sizes of the blocks are small.

$$\bar{\gamma}_i \cong \gamma_o = \frac{1}{H_o}. \quad (4.4.19)$$

The average dimensionless gradient is used to find the dimensionless potential at the free surface, according to the boundary condition expressed by the Eq. 3.2.2, discretized below:

$$W_{i,1} = p_{D_{i,1}} + \gamma_o \left(z_{D_{i,1}} + \frac{\Delta h_{D_i}}{2} \right), \quad (4.4.20)$$

Since $p_{D_{i,1}}$ is the pressure at the block node center, and considering that the free surface is at atmospheric pressure, the above expression can simplified to:

$$W_{i,1} = \frac{h_{D_i}}{H_{o_D}}. \quad (4.4.21)$$

Thus, the uppermost layer properties become congruent to the free surface properties in the discretized process.

Between nodes, the free surface elevation is interpolated by:

$$\bar{h}_{D_i} = (1 - \alpha_i) h_{D_{i-1}} + \alpha_i h_{D_i} \quad (4.4.22)$$

where the parameter α_i is a linear function of the radial distance:

$$\alpha_i = \frac{r_{D_i} - \bar{r}_{D_{i-1}}}{\bar{r}_{D_i} - \bar{r}_{D_{i-1}}}. \quad (4.4.23)$$

Since we define:

$$h_{D_i} = z_{D_{i,1}} + \Delta h_{D_i} \quad (4.4.24)$$

$$\bar{h}_{D_i} = \bar{z}_{D_{i,1}} + \overline{\Delta h}_{D_i} \quad (4.4.25)$$

and:

$$\bar{z}_{D_i} = (1 - \alpha_i) z_{D_{i-1}} + \alpha_i z_{D_i} \quad (4.4.26)$$

then:

$$\overline{\Delta h}_{D_i} = (1 - \alpha_i) \Delta h_{D_{i-1}} + \alpha_i \Delta h_{D_i}, \quad (4.4.27)$$

the overbar indicating the values taken between two consecutive nodes.

4.4.2 Free Surface Inner Boundary Equation

Similar procedures are used to obtain the equation for the inner boundary block containing the free surface, and the results are summarized below. A discretization of the continuity equation gives the following material balance residue equation:

$$\Psi_{1,1} = A_{1,1} W_{1,1} + C_{1,1} W_{1,2} + E_{1,1} W_{2,1} - F_{1,1} \quad (4.4.28)$$

where:

$$\begin{aligned} A_{1,1} &= -\theta \left[\overline{\Delta h}_{D_2} + f_{x_{1,1}} f_{r_1} \overline{\Delta h}_{D_1} + \frac{2k_v A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1} (\Delta h_{D_1} + \Delta z_{D_1})} \right. \\ &\quad \left. + \frac{k_h A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1}} \frac{\Delta h_{D_1} + \Delta h_{D_1}^k}{2\theta \Delta t_D} c_{t_D} \right] \end{aligned} \quad (4.4.29)$$

$$C_{1,1} = \frac{2\theta k_v A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1} (\Delta h_{D_1} + \Delta z_{D_1})} \quad (4.4.30)$$

$$E_{1,1} = \theta \overline{\Delta h}_{D_2} \quad (4.4.31)$$

$$F_{1,1} = \frac{k_h}{k_{x_{2,1}} \cos \beta_{2,1}} \frac{A_{1,1}}{\Delta t_D \cos \beta_{1,1}} \left[(\Delta h_{D_1} - \Delta h_{D_1}^k) S_g - \left(\frac{\Delta h_{D_1} + \Delta h_{D_1}^k}{2} \right) c_{t_D} W_{1,1}^k \right]$$

$$\begin{aligned}
& -f_{x_{1,1}} f_{r_1} \left[\theta \overline{\Delta h}_{D_1} W_{w_1} - (1 - \theta) \overline{\Delta h}_{D_1}^k \left(W_{1,1}^k - W_{w_1}^k \right) \right] - (1 - \theta) \times \\
& \left\{ \overline{\Delta h}_{D_2}^k \left[W_{2,1}^k - W_{1,1}^k \right] - \frac{2 k_v A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1} (\Delta h_{D_1}^k + \Delta z_{D_1})} \left[W_{1,1}^k - W_{1,2}^k \right] \right\} \\
\end{aligned} \tag{4.4.32}$$

Consider Δh_{D_w} the extrapolated thickness at the sandface, and the mean thickness:

$$\overline{\Delta h}_{D_1} = \frac{\Delta h_{D_1} + \Delta h_{D_w}}{2} \tag{4.4.33}$$

Also:

$$\alpha_2 = \frac{r_{D_2} - \bar{r}_{D_1}}{\bar{r}_{D_2} - \bar{r}_{D_1}}, \tag{4.4.34}$$

$$\alpha_w = \frac{\bar{r}_{D_2} - 1}{\bar{r}_{D_2} - \bar{r}_{D_1}}, \tag{4.4.35}$$

$$\alpha_1 = \frac{1 + \alpha_{D_w}}{2}. \tag{4.4.36}$$

The differentiations of the residue equation give the Jacobian elements, as follows:

$$DA_{1,1} = \frac{\partial \Psi_{1,1}}{\partial W_{1,1}} = A_{1,1} + \frac{\partial A_{1,1}}{\partial W_{1,1}} W_{1,1} + \frac{\partial C_{1,1}}{\partial W_{1,1}} W_{1,2} + \frac{\partial E_{1,1}}{\partial W_{1,1}} W_{2,1} - \frac{\partial F_{1,1}}{\partial W_{1,1}} \tag{4.4.37}$$

The resulting differentiation produces:

$$\begin{aligned}
DA_{1,1} = A_{1,1} - & \left[\frac{\theta (1 - \alpha_2 + \alpha_1 f_{x_{1,1}} f_{r_1})}{\bar{\gamma}_1} - \frac{C_{1,1}}{(\Delta h_{D_1} + \Delta z_{D_1}) \bar{\gamma}_1} + \frac{k_h A_{1,1}}{k_{x_{2,1}} 2 \bar{\gamma}_1 \Delta t_D} \frac{c_{t_D}}{2} \right] W_{1,1} \\
& - \frac{C_{1,1}}{(\Delta h_{D_1} + \Delta z_{D_1}) \bar{\gamma}_1} W_{1,2} + \frac{\theta (1 - \alpha_2)}{\bar{\gamma}_1} W_{2,1} - \frac{k_h}{k_{x_{2,1}}} \frac{A_{1,1}}{\Delta t_D \bar{\gamma}_1} \left[S_g - \frac{c_{t_D} W_{1,1}^k}{2} \right] \\
& + \frac{\theta \alpha_1 f_{x_{1,1}} f_{r_1}}{\bar{\gamma}_1} W_{w_1} + \theta f_{x_{1,1}} f_{r_1} \overline{\Delta h}_1 \frac{\partial W_{w_1}}{\partial W_{1,1}} \\
\end{aligned} \tag{4.4.38}$$

α_1 and α_2 above were obtained from the differentiations of $\bar{\Delta}h_D$. Similarly, the other Jacobian coefficients are:

$$DB_{1,1} = 0 \quad (4.4.39)$$

$$DC_{1,1} = \frac{\partial \Psi_{1,1}}{\partial W_{1,2}} = C_{1,1} + \theta f_{x_{1,1}} f_{r_1} \bar{\Delta}h_1 \frac{\partial W_{w_1}}{\partial W_{1,2}} \quad (4.4.40)$$

$$DD_{1,1} = 0 \quad (4.4.41)$$

$$\begin{aligned} DE_{i,1} = \frac{\partial \Psi_{1,1}}{\partial W_{2,1}} = E_{1,1} - \frac{\theta [\alpha_2 + f_{x_{1,1}} f_{r_1} (1 - \alpha_1)]}{\bar{\gamma}_2} W_{1,1} + \frac{\theta \alpha_2}{\bar{\gamma}_2} \dot{W}_{2,1} \\ + f_{x_{1,1}} f_{r_1} \theta \left(\frac{1 - \alpha_1}{\bar{\gamma}_2} W_{w_1} + \Delta \bar{h}_{D_1} \frac{\partial W_{w_1}}{\partial W_{2,1}} \right) \end{aligned} \quad (4.4.42)$$

Values of the derivatives of the inner boundary potentials can be obtained separately, and the equations are presented in App. A.1.

4.4.3 Free Surface Outer Boundary Blocks

Following the same procedures, the final coefficients are:

$$A_{m,1} = -\theta \left[\bar{\Delta}h_{D_{m+1}} + f_{x_{m,1}} f_{r_m} \bar{\Delta}h_{D_m} + \frac{2k_v A_{1,m}}{k_h (\Delta h_{D_m} + \Delta z_{D_m})} \right] \quad (4.4.43)$$

$$+ \frac{A_{1,m} (\Delta h_{D_m} + \Delta h_{D_m}^k)}{2\theta \cos \bar{\beta}_{1,m} \Delta t_D} c_{t_D} \quad (4.4.44)$$

$$C_{m,1} = + \frac{2\theta k_v A_{1,m}}{k_h (\Delta h_{D_m} + \Delta z_{D_m})} \quad (4.4.45)$$

$$D_{m,1} = \theta f_{x_{m,1}} f_{r_m} \bar{\Delta}h_{D_m} W_{m-1,1} \quad (4.4.46)$$

$$F_{m,1} = \frac{A_{1,m}}{\Delta t_D \cos \bar{\beta}_{1,m}} \left[(\Delta h_{D_m} - \Delta h_{D_m}^k) S_g - \left(\frac{\Delta h_{D_m} + \Delta h_{D_m}^k}{2} \right) c_{t_D} W_{m,1}^k \right]$$

$$\begin{aligned}
& -\overline{\Delta h}_{D_{m+1}}^k \left[1 - (1 - \theta) W_{m,1}^k \right] - (1 - \theta) \times \\
& \left\{ -f_{x_{m,1}} f_{r_m} \overline{\Delta h}_{D_m}^k \left[W_{m,1}^k - W_{m-1,1}^k \right] - \right. \\
& \left. \frac{2 k_v A_{1_m}}{k_h (\Delta h_{D_m}^k + \Delta z_{D_m})} \left[W_{m,1}^k - W_{m,2}^k \right] \right\} \tag{4.4.47}
\end{aligned}$$

The Jacobian matrix elements for the block $m, 1$ are determined using a Newton-Raphson procedure:

Let:

$$\Psi_{m,1} = A_{m,1} W_{m,1} + C_{m,1} W_{m,2} + D_{m,1} W_{m-1,1} - F_{m,1} \tag{4.4.48}$$

Differentiating Eq. 4.4.48 with respect to each variable $W_{m,n}$:

$$\begin{aligned}
DA_{m,1} = \frac{\partial \Psi_{m,1}}{\partial W_{m,1}} = A_{m,1} + \frac{\partial A_{m,1}}{\partial W_{m,1}} W_{m,1} + \frac{\partial C_{m,1}}{\partial W_{m,1}} W_{m,2} + \\
\frac{\partial D_{m,1}}{\partial W_{m,1}} W_{m-1,1} - \frac{\partial F_{m,1}}{\partial W_{m,1}}. \tag{4.4.49}
\end{aligned}$$

The resulting differentiation produces:

$$\begin{aligned}
DA_{m,1} = A_{m,1} - \left[\frac{\theta (\alpha_m f_{x_{m,1}} f_{r_m})}{\bar{\gamma}_m} - \frac{C_{m,1}}{(\Delta h_{D_m} + \Delta z_{D_m}) \bar{\gamma}_m} + \frac{A_{1_m} c_{t_D}}{2 \bar{\gamma}_m \Delta t_D \cos \bar{\beta}_{m,1}} \right] W_{m,1} \\
- \frac{C_{m,1}}{(\Delta h_{D_m} + \Delta z_{D_m}) \bar{\gamma}_m} W_{m,2} + \frac{\theta \alpha_m f_{x_{m,1}} f_{r_m}}{\bar{\gamma}_m} W_{m-1,1} + \\
- \frac{A_{1_m}}{\Delta t_D \cos \bar{\beta}_{m,1} \bar{\gamma}_m} \left[S_g - \frac{c_{t_D} W_{m,1}^k}{2} \right] \tag{4.4.50}
\end{aligned}$$

In Eq. 4.4.50 the same definitions as those used so far are applied to the parameters. It is helpful to define the outer boundary average thickness:

$$\overline{\Delta h}_{D_{m+1}} = (1 - \alpha_{m+1}) \Delta h_{D_m} + \alpha_{m+1} \Delta h_{D_{m+1}} = (1 - \alpha_{m+1}) \Delta h_{D_m} + \frac{\alpha_{m+1}}{n} \quad (4.4.51)$$

where:

$$\alpha_{m+1} = \frac{r_{e_D} - \bar{r}_m}{\bar{r}_{m+1} - \bar{r}_m} \quad (4.4.52)$$

Similarly, the other Jacobian coefficients are:

$$DB_{m,1} = 0 \quad (4.4.53)$$

$$DC_{m,1} = \frac{\partial \Psi_{m,1}}{\partial W_{m,2}} = C_{m,1} \quad (4.4.54)$$

$$DD_{m,1} = \frac{\partial \Psi_{m,1}}{\partial W_{m-1,1}} = D_{m,1} + \frac{\theta (1 - \alpha_m) f_{x_{m,1}} f_{r_m}}{\bar{\gamma}_{m-1}} [W_{m-1,1} - W_{m,1}] \quad (4.4.55)$$

$$DE_{m,1} = \frac{\partial \Psi_{m,1}}{\partial W_{m+1,1}} = 0 \quad (4.4.56)$$

4.5 Lower Boundary Blocks

The lower boundary is defined by the no-flow condition defined by Eq. 3.2.10. The resulting discretization of Eq. 4.1.3 for each region, namely the inner boundary, the outer boundary, and the main reservoir follows.

4.5.1 Saturated Region Lower Boundary Equations

The residue in the block i, n is:

$$\Psi_{i,n} = A_{i,n} W_{i,n} + B_{i,n} W_{i,n-1} + D_{i,n} W_{i-1,n} + E_{i,n} W_{i+1,n} - F_{i,n} \quad (4.5.1)$$

where the coefficients are:

$$\begin{aligned} A_{i,n} &= -\theta \left[\overline{\Delta z}_{D_{i+1}} + f_{x_{i,n}} f_{r_i} \overline{\Delta z}_{D_i} + \frac{k_v}{k_{x_{i+1,n}} \cos \beta_{i,n}} \frac{A_{1_i}}{\Delta z_{D_i}} \right. \\ &\quad \left. + \frac{k_h}{k_{x_{i+1,n}} \cos \beta_{i,n}} \frac{A_{1_i} \Delta z_{D_i}}{\theta \Delta t_D \cos \beta_{i,n}} c_{t_D} \right] \end{aligned} \quad (4.5.2)$$

$$B_{i,n} = \frac{\theta k_v A_{1_i}}{k_{x_{i+1,n}} \cos \beta_{i,n} \Delta z_{D_i}} \quad (4.5.3)$$

$$D_{i,n} = \theta f_{x_{i,n}} f_{r_i} \overline{\Delta z}_{D_i} \quad (4.5.4)$$

$$E_{i,n} = \theta \overline{\Delta z}_{D_{i+1}} \quad (4.5.5)$$

$$\begin{aligned} F_{i,n} &= -\frac{k_h}{k_{x_{i+1,n}} \cos \beta_{i,n}} \frac{A_{1_i} \Delta z_{D_i}}{\Delta t_D \cos \beta_{i,n}} c_{t_D} W_{i,n}^k - (1 - \theta) \left\{ \overline{\Delta z}_{D_{i+1}} [W_{i+1,n}^k - W_{i,n}^k] \right. \\ &\quad \left. - f_{x_{i,n}} f_{r_i} \overline{\Delta z}_{D_i} [W_{i,n}^k - W_{i-1,n}^k] + \frac{k_v}{k_{x_{i+1,n}} \cos \beta_{i,n}} \frac{A_{1_i}}{\Delta z_{D_i}} [W_{i,n-1}^k - W_{i,n}^k] \right\}. \end{aligned} \quad (4.5.6)$$

Jacobian matrix elements for the block i, n were determined using a Newton-Raphson procedure. Differentiating Eq. 4.5.1 with respect to each variable W :

$$DA_{i,n} = \frac{\partial \Psi_{i,n}}{\partial W_{i,n}} = A_{i,n} \quad (4.5.7)$$

$$DB_{i,n} = \frac{\partial \Psi_{i,n}}{\partial W_{i,n-1}} = B_{i,n} \quad (4.5.8)$$

$$DC_{i,n} = 0 \quad (4.5.9)$$

$$DD_{i,n} = \frac{\partial \Psi_{i,n}}{\partial W_{i-1,n}} = D_{i,n} \quad (4.5.10)$$

$$DE_{i,n} = \frac{\partial \Psi_{i,n}}{\partial W_{i+1,n}} = E_{i,n} \quad (4.5.11)$$

4.5.2 Lower Inner Boundary Equation

The residue in the block i, n is:

$$\Psi_{1,n} = A_{1,n} W_{1,n} + B_{1,n} W_{1,n-1} + E_{1,n} W_{2,n} - F_{1,n} \quad (4.5.12)$$

in which the coefficients are:

$$\begin{aligned} A_{1,n} &= -\theta \left[\overline{\Delta z}_{D_2} + f_{x_{1,n}} f_{r_1} \overline{\Delta z}_{D_1} + \frac{k_v}{k_{x_{2,n}} \cos \beta_{2,n}} \frac{A_{11}}{\Delta z_{D_1}} \right. \\ &\quad \left. + \frac{k_h}{k_{x_{2,n}} \cos \beta_{2,n}} \frac{A_{11} \Delta z_{D_1}}{\theta \Delta t_D \cos \beta_{1,n}} c_{tD} \right] \end{aligned} \quad (4.5.13)$$

$$B_{1,n} = \frac{\theta k_v A_{11}}{k_{x_{2,n}} \cos \beta_{2,n} \Delta z_{D_1}} \quad (4.5.14)$$

$$E_{1,n} = \theta \overline{\Delta z}_{D_2} \quad (4.5.15)$$

$$\begin{aligned} F_{1,n} &= -\frac{k_h}{k_{x_{2,n}} \cos \beta_{2,n}} \frac{A_{11} \Delta z_{D_1}}{\Delta t_D \cos \beta_{1,n}} c_{tD} W_{1,n}^k - \theta f_{x_{1,n}} f_{r_1} \overline{\Delta z}_{D_1} W_{w_n} \\ &\quad - (1 - \theta) \left\{ \overline{\Delta z}_{D_2} [W_{2,n}^k - W_{1,n}^k] - f_{x_{1,n}} f_{r_1} \overline{\Delta z}_{D_1} [W_{1,n}^k - W_{w_n}^k] \right. \\ &\quad \left. + \frac{k_v}{k_{x_{2,n}} \cos \beta_{2,n}} \frac{A_{11}}{\Delta z_{D_1}} [W_{1,n-1}^k - W_{1,n}^k] \right\} \end{aligned} \quad (4.5.16)$$

Find the Jacobian matrix elements for the block $1, n$ using a Newton-Raphson procedure. Differentiating Eq. 4.5.12 with respect to each variable W :

$$DA_{1,n} = \frac{\partial \Psi_{1,n}}{\partial W_{1,n}} = A_{1,n} - \frac{\partial F_{1,n}}{\partial W_{1,n}} = A_{1,n} + \theta f_{x_{1,n}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_n}}{\partial W_{1,n}} \quad (4.5.17)$$

$$DB_{1,n} = \frac{\partial \Psi_{1,n}}{\partial W_{1,n-1}} = B_{1,n} + \theta f_{x_{1,n}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_n}}{\partial W_{1,n-1}} \quad (4.5.18)$$

$$DC_{1,n} = 0 \quad (4.5.19)$$

$$DD_{1,n} = 0 \quad (4.5.20)$$

$$DE_{1,n} = \frac{\partial \Psi_{1,n}}{\partial W_{2,n}} = E_{1,n} \quad (4.5.21)$$

The partial differentiations of the inner boundary potentials (see App. A.1), calculated separately may be substituted above.

4.5.3 Lower Outer Boundary Equation

The residue in the block m, n is:

$$\Psi_{m,n} = A_{m,n} W_{m,n} + B_{m,n} W_{m,n-1} + D_{m,n} W_{m-1,n} - F_{m,n} \quad (4.5.22)$$

in which the coefficients are:

$$A_{m,n} = -\theta \left[\overline{\Delta z}_{D_{m+1}} + f_{x_{m,n}} f_{r_m} \overline{\Delta z}_{D_m} + \frac{k_v}{k_h} \frac{A_{1m}}{\Delta z_{D_m}} + \frac{A_{1m} \Delta z_{D_m}}{\theta \Delta t_D \cos \bar{\beta}_{m,n}} c_{t_D} \right]$$

(4.5.23)

$$B_{m,n} = \frac{\theta k_v A_{1m}}{k_h \Delta z_{D_m}} \quad (4.5.24)$$

$$D_{m,n} = \theta f_{x_{m,n}} f_{r_m} \overline{\Delta z}_{D_m} \quad (4.5.25)$$

$$\begin{aligned} F_{m,n} = & - \frac{A_{1m} \Delta z_{D_m}}{\Delta t_D \cos \bar{\beta}_{m,n}} c_{t_D} W_{m,n}^k - \Delta z_{D_{m+1}} \\ & - (1 - \theta) \left\{ -\overline{\Delta z}_{D_{m+1}} W_{m,n}^k - f_{x_{m,n}} f_{r_m} \overline{\Delta z}_{D_m} [W_{m,n}^k - W_{m-1,n}^k] \right. \\ & \left. + \frac{k_v}{k_h} \frac{A_{1m}}{\Delta z_{D_m}} [W_{m,n-1}^k - W_{m,n}^k] \right\} \end{aligned} \quad (4.5.26)$$

Determine Jacobian matrix elements for the block m, n using a Newton-Raphson procedure.

Differentiating Eq. 4.5.22 with respect to each variable W :

$$DA_{m,n} = \frac{\partial \Psi_{m,n}}{\partial W_{m,n}} = A_{m,n} \quad (4.5.27)$$

$$DB_{m,n} = \frac{\partial \Psi_{m,n}}{\partial W_{m,n-1}} = B_{m,n} \quad (4.5.28)$$

$$DC_{m,n} = 0 \quad (4.5.29)$$

$$DD_{m,n} = \frac{\partial \Psi_{m,n}}{\partial W_{m-1,n}} = D_{m,n} \quad (4.5.30)$$

$$DE_{m,n} = 0 \quad (4.5.31)$$

4.6 Second Layer Region Blocks

The second layer, the one immediately below the free surface layer, is a saturated block. However, the vertical component of the discretized equation contains a variable

corresponding to the thickness of the free surface layer block above it. For that reason, the equations are summarized below.

4.6.1 Second Layer Equations

The residue in the block $i, 2$ ($1 < i < m$) is:

$$\Psi_{i,2} = A_{i,2} W_{i,2} + B_{i,2} W_{i,1} + C_{i,2} W_{i,3} + D_{i,2} W_{i-1,2} + E_{i,2} W_{i+1,2} - F_{i,1} \quad (4.6.1)$$

in which the coefficients are:

$$\begin{aligned} A_{i,2} &= -\theta \left[\overline{\Delta z}_{D_{i+1}} + f_{x_{i,2}} f_{r_i} \overline{\Delta z}_{D_i} + \frac{k_v A_{1,i}}{k_{x_{i+1,2}} \cos \beta_{i+1,2}} \left(\frac{1}{\Delta z_{D_i}} + \frac{2}{\Delta z_{D_i} + \Delta h_{D_i}} \right) \right. \\ &\quad \left. + \frac{k_h}{k_{x_{i+1,2}} \cos \beta_{i+1,2}} \frac{A_{1,i} \Delta z_{D_i}}{\theta \Delta t_D \cos \beta_{i,2}} c_{t_D} \right] \end{aligned} \quad (4.6.2)$$

$$B_{i,2} = + \frac{2\theta k_v A_{1,i}}{k_{x_{i+1,2}} \cos \beta_{i+1,2} (\Delta z_{D_i} + \Delta h_{D_i})} \quad (4.6.3)$$

$$C_{i,2} = + \frac{\theta k_v A_{1,i}}{k_{x_{i+1,2}} \cos \beta_{i+1,2} \Delta z_{D_i}} \quad (4.6.4)$$

$$D_{i,2} = + \theta f_{x_{i,2}} f_{r_i} \overline{\Delta z}_{D_i} \quad (4.6.5)$$

$$E_{i,2} = + \theta \overline{\Delta z}_{D_{i+1}} \quad (4.6.6)$$

$$\begin{aligned} F_{i,2} &= - \frac{k_h}{k_{x_{i+1,2}} \cos \beta_{i+1,2}} \frac{A_{1,i} \Delta z_{D_i}}{\Delta t_D \cos \beta_{i,2}} c_{t_D} W_{i,2}^k - (1-\theta) \left\{ \overline{\Delta z}_{D_{i+1}} [W_{i+1,2}^k - W_{i,2}^k] \right. \\ &\quad - f_{x_{i,2}} f_{r_i} \overline{\Delta z}_{D_i} [W_{i,2}^k - W_{i-1,2}^k] + \frac{2k_v A_{1,i}}{k_{x_{i+1,2}} \cos \beta_{i+1,2} (\Delta z_{D_i} + \Delta h_{D_i}^k)} [W_{i,1}^k - W_{i,2}^k] \\ &\quad \left. - \frac{k_v A_{1,i}}{k_{x_{i+1,2}} \cos \beta_{i+1,2} \Delta z_{D_i}} [W_{i,2}^k - W_{i,3}^k] \right\} \end{aligned} \quad (4.6.7)$$

Find the Jacobian matrix elements for the block $i, 2$ using a Newton-Raphson procedure.

Differentiating Eq. 4.6.1 with respect to each variable W :

$$DA_{i,2} = \frac{\partial \Psi_{i,2}}{\partial W_{i,2}} = A_{i,2} \quad (4.6.8)$$

$$DB_{i,2} = \frac{\partial \Psi_{i,2}}{\partial W_{i,1}} = B_{i,2} \left[1 - \frac{W_{i,1} - W_{i,2}}{(\Delta z_{D_i} + \Delta h_{D_i}) \bar{\gamma}_i} \right] \quad (4.6.9)$$

$$DC_{i,2} = \frac{\partial \Psi_{i,2}}{\partial W_{i,3}} = C_{i,2} \quad (4.6.10)$$

$$DD_{i,2} = \frac{\partial \Psi_{i,2}}{\partial W_{i-1,2}} = D_{i,2} \quad (4.6.11)$$

$$DE_{i,2} = \frac{\partial \Psi_{i,2}}{\partial W_{i+1,2}} = E_{i,2} \quad (4.6.12)$$

4.6.2 Second Layer Inner Boundary Equation

The residue in the block 1,2 is:

$$\Psi_{1,2} = A_{1,2} W_{1,2} + B_{1,2} W_{1,1} + C_{1,2} W_{1,3} + E_{1,2} W_{2,2} - F_{1,2} \quad (4.6.13)$$

in which the coefficients are:

$$\begin{aligned} A_{1,2} = & -\theta \left[\overline{\Delta z}_{D_2} + f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} + \frac{k_v A_{11}}{k_{x_{2,2}} \cos \beta_{2,2}} \left(\frac{1}{\Delta z_{D_1}} + \frac{2}{\Delta z_{D_1} + \Delta h_{D_1}} \right) \right. \\ & \left. + \frac{k_h}{k_{x_{2,2}} \cos \beta_{2,2}} \frac{A_{11} \Delta z_{D_1}}{\theta \Delta t_D \cos \bar{\beta}_{1,2}} c_{t_D} \right] \end{aligned} \quad (4.6.14)$$

$$B_{1,2} = + \frac{2\theta k_v A_{11}}{k_{x_{2,2}} \cos \beta_{2,2} (\Delta z_{D_1} + \Delta h_{D_1})} \quad (4.6.15)$$

$$C_{1,2} = + \frac{\theta k_v A_{11}}{k_{x_{2,2}} \cos \beta_{2,2} \Delta z_{D_1}} \quad (4.6.16)$$

$$E_{1,2} = + \theta \overline{\Delta z}_{D_2} \quad (4.6.17)$$

$$\begin{aligned}
F_{1,2} = & -\theta f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} W_{w_2} - \frac{k_h}{k_{x_{2,2}} \cos \beta_{2,2}} \frac{A_{1,1} \Delta z_{D_1}}{\Delta t_D \cos \bar{\beta}_{1,2}} c_{t_D} W_{1,2}^k \\
& - (1 - \theta) \left\{ \overline{\Delta z}_{D_2} \left[W_{2,2}^k - W_{1,2}^k \right] - f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} \left[W_{1,2}^k - W_{w_2}^k \right] \right. \\
& \left. \frac{2 k_v A_{1,1}}{k_{x_{2,2}} \cos \beta_{2,2}} \left[\frac{2 (W_{2,2}^k - W_{1,2}^k)}{\Delta z_{D_1} + \Delta h_{D_1}^k} - \frac{W_{1,2}^k - W_{1,3}^k}{\Delta z_{D_1}} \right] \right\} \quad (4.6.18)
\end{aligned}$$

Find the Jacobian matrix elements for the block $i, 2$ using a Newton-Raphson procedure.

Differentiating Eq. 4.6.18 with respect to each variable W :

$$DA_{1,2} = \frac{\partial \Psi_{1,2}}{\partial W_{1,2}} = A_{1,2} - \frac{\partial F_{1,2}}{\partial W_{1,2}} = A_{1,2} + \theta f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_2}}{\partial W_{1,2}} \quad (4.6.19)$$

$$DB_{1,2} = \frac{\partial \Psi_{1,2}}{\partial W_{1,1}} = B_{1,2} \left[1 - \frac{W_{1,1} - W_{1,2}}{(\Delta z_{D_1} + \Delta h_{D_1}) \bar{\gamma}_1} \right] + \theta f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_2}}{\partial W_{1,1}} \quad (4.6.20)$$

$$DC_{1,2} = \frac{\partial \Psi_{1,2}}{\partial W_{1,3}} = C_{1,2} + \theta f_{x_{1,2}} f_{r_1} \overline{\Delta z}_{D_1} \frac{\partial W_{w_2}}{\partial W_{1,3}} \quad (4.6.21)$$

$$DD_{1,2} = 0 \quad (4.6.22)$$

$$DE_{1,2} = \frac{\partial \Psi_{1,2}}{\partial W_{2,2}} = E_{1,2} \quad (4.6.23)$$

The partial differentiations of the inner boundary potentials (see App. A), calculated individually, may be substituted above.

4.6.3 Second Layer Outer Boundary Equation

The residue in the block $m, 2$ is:

$$\Psi_{m,2} = A_{m,2} W_{m,2} + B_{m,2} W_{m,1} + C_{m,2} W_{m,3} + D_{m,2} W_{m-1,2} - F_{m,2} \quad (4.6.24)$$

where the coefficients are:

$$\begin{aligned}
A_{m,2} &= -\theta \left[\overline{\Delta z}_{D_{m+1}} + f_{x_{m,2}} f_{r_m} \overline{\Delta z}_{D_m} + \frac{k_v A_{1m}}{k_h} \left(\frac{1}{\Delta z_{D_m}} + \frac{2}{\Delta z_{D_m} + \Delta h_{D_m}} \right) \right. \\
&\quad \left. + \frac{A_{1m} \Delta z_{D_m}}{\Delta t_D \cos \bar{\beta}_{m,2}} c_{t_D} \right] \tag{4.6.25}
\end{aligned}$$

$$B_{m,2} = + \frac{2\theta k_v A_{1m}}{k_h (\Delta z_{D_m} + \Delta h_{D_m})} \tag{4.6.26}$$

$$C_{m,2} = + \frac{\theta k_v A_{1m}}{k_h \Delta z_{D_m}} \tag{4.6.27}$$

$$D_{m,2} = + \theta f_{x_{m,2}} f_{r_m} \overline{\Delta z}_{D_m} \tag{4.6.28}$$

$$\begin{aligned}
F_{m,2} &= - \frac{A_{1m} \Delta z_{D_m}}{\Delta t_D \cos \bar{\beta}_{m,2}} c_{t_D} W_{m,2}^k - \theta \overline{\Delta z}_{D_{m+1}} - (1 - \theta) \left\{ \overline{\Delta z}_{D_{m+1}} [1 - W_{m,2}^k] \right. \\
&\quad - f_{x_{m,2}} f_{r_m} \overline{\Delta z}_{D_m} [W_{m,2}^k - W_{m-1,2}^k] + \frac{2k_v A_{1m}}{k_h (\Delta z_{D_m} + \Delta h_{D_m}^k)} [W_{m,1}^k - W_{m,2}^k] \\
&\quad \left. - \frac{k_v A_{1m}}{k_h \Delta z_{D_m}} [W_{m,2}^k - W_{m,3}^k] \right\} \tag{4.6.29}
\end{aligned}$$

Find the Jacobian matrix elements for the block $m, 2$ using a Newton-Raphson procedure.

Differentiating Eq. 4.6.24 with respect to each variable W :

$$DA_{m,2} = \frac{\partial \Psi_{m,2}}{\partial W_{m,2}} = A_{m,2} \tag{4.6.30}$$

$$DB_{m,2} = \frac{\partial \Psi_{m,2}}{\partial W_{m,1}} = B_{m,2} \left[1 - \frac{W_{m,1} - W_{m,2}}{(\Delta z_{D_m} + \Delta h_{D_m}) \bar{\gamma}_m} \right] \tag{4.6.31}$$

$$DC_{m,2} = \frac{\partial \Psi_{m,2}}{\partial W_{m,1}} = C_{m,2} \tag{4.6.32}$$

$$DD_{m,2} = \frac{\partial \Psi_{m,2}}{\partial W_{m-1,2}} = D_{m,2} \tag{4.6.33}$$

$$DE_{m,2} = 0 \tag{4.6.34}$$

4.7 Capillary Fringe

So far in this chapter, the existence of the capillary fringe in the model has been neglected. The importance of the capillary fringe appears to be a matter of scale, because in most field cases, the capillary layer is negligible compared to the total liquid height. To verify the model against sandbox experiments, however, it is necessary to introduce capillary effects. Due to this modification, the free surface concept must be reconsidered. Thus, we will use the terminology of groundwater literature to denote as phreatic surface the interface domain corresponding to atmospheric pressure in the saturated region below the capillary fringe.

The capillary fringe is a layer immediately above the phreatic surface, partially or completely saturated, possessing the particular characteristic of pressures below atmospheric as a function of the capillary pressure. The uniformity of thickness and saturation is essentially dependent on the uniformity of grain and pore throat size distributions, and on the surface tension of the gas-liquid-rock system. At rest, the potentials in the fringe have the same value as in the saturated region, but at dynamic conditions, the potentials have the same value as in the phreatic line. According to the *Wyckoff et al.* (1932) and *Hall* (1955) sandbox experiments, the capillary fringe is not a dead region, and downward movement of the phreatic surface causes liquid to flow by the potential reduction generating pressure gradients. In the radial model sandbox experiments, it is possible to observe the tracks of the stream lines in the capillary fringe leaving the outer boundary in an upward direction, then following the phreatic surface, and bending downward in the vicinity of the wellbore.

4.7.1 Description of the Physical Mechanism of the Capillary Flow

A description of the physical behavior and the causes of the capillary flow is presented next, with the purpose of creating a mathematical representation to be added to the Stream Layer Model. We start by assuming an anisotropic formation saturated with a single liquid, presenting a uniform capillary layer of thickness h_c at rest, above the original static level H_o . The potential and pressure distributions at any radial position in the formation is represented by Fig. 4.6. The pressures above H_o are negative due to the capillary pressure γh_c , which pulls the gas-liquid contact out, normal to the surface. The phreatic line at rest corresponds to atmospheric pressure.

When pumping starts, the potentials below the phreatic surface decrease, and the

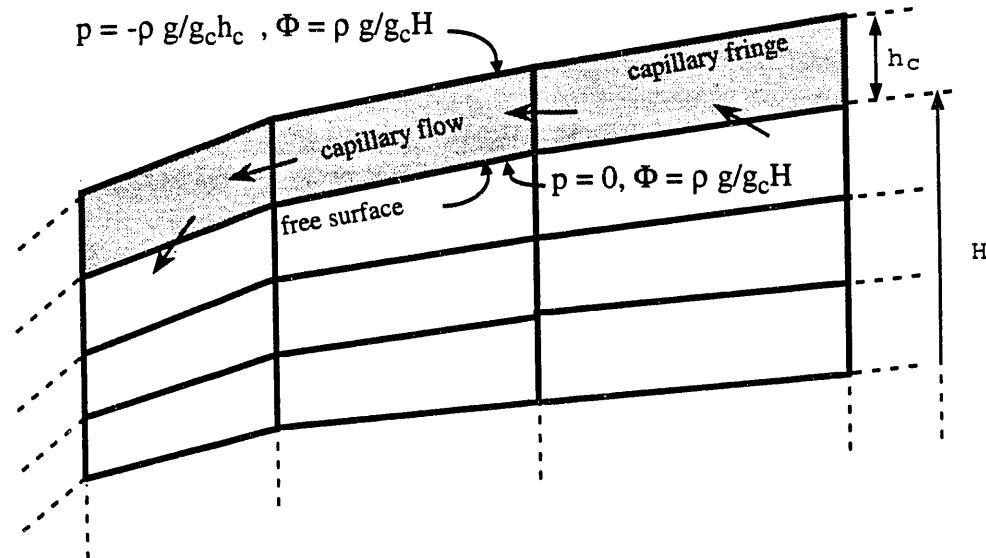


Figure 4.5: Representation of the capillary fringe layer.

surface travels downward producing a pressure reduction in the capillary layer. Since there is a variable potential distribution along the phreatic surface, an equivalent distribution should exist above it. The liquid in the fringe starts moving accordingly. In the gas liquid contact, the pressure reduction forces a movement of the meniscus and gas entry in a typical two-phase flow leaving a decreasing liquid saturation above the gas-liquid contact decreasing to the irreducible value S_{w_r} . The capillary fringe tends to maintain the thickness h_c , but, in a transient flow it may be possible that h_c will become variable due to the resistance to the air entry by the lower relative permeability to gas, at high liquid saturations in the desaturating region. However, it is also possible that the capillary fringe presents a residual gas saturation in cases of sandbox experiments, and two-phase flow should be considered in this region. It is reasonable to expect expansion of the capillary fringe mainly around the wellbore at early times where the vertical velocities of the free surface drawdown are higher than those of the air entry before reaching steady conditions.

The potential variation in the capillary layer generates flow in that region, which, neglecting the temporary expansion described before, may be considered of constant storativity. So, the capillary fringe acts as an additional source of flow into the phreatic surface, close to the well. However, due to the constant storativity, the liquid change through the phreatic interface varies in direction and intensity. It was observed experimentally that there were stream lines moving upward across the phreatic surface feeding the capillary layer. *Hall (1955)* pointed out that “the phreatic line is not a flow line in a case where a

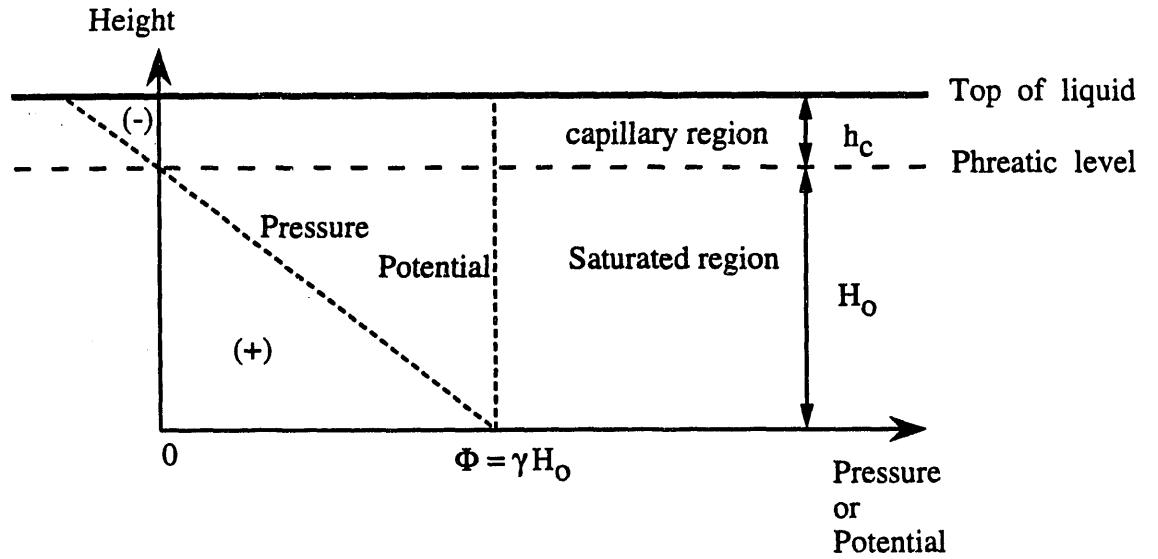


Figure 4.6: Potential and pressure profiles at a radial position r , under static conditions.

capillary layer exists." Hence, there must be a flux profile along the radial direction showing maximum absolute values at the inner and outer boundaries, though with different directions. Somewhere in between the flux is zero.

4.7.2 Mathematical Representation of the Capillary Layer Boundary

Based on the preceding flow concept, we could rewrite the free surface layer equation incorporating a source term, which is the contribution of the capillary layer. Some simplifications are often made in the capillary flow such as to consider constant dynamic vertical gradient approaching the static gradient in the capillary fringe, as well as neglecting the transient behavior of delayed air entry. The following mathematical relationships are valid in the capillary layer where the pressures are negative ($H_o \leq z \leq H_o + h_c$):

$$\Phi_c(r, t) = \Phi_{ph}(r, t) \quad (4.7.1)$$

$$\Phi_{ph}(r, t) = \left(\rho \frac{g}{g_c} \right) h(r, t) \quad (4.7.2)$$

$$\Phi(r, h + h_c, t) = \left(\rho \frac{g}{g_c}\right) h(r, t) \quad (4.7.3)$$

where the subscripts c and ph refer to the capillary layer and to the phreatic surface, and h represents the phreatic level. Equation 4.7.3 is a consequence of Eq. 4.7.1 and Eq. 4.7.2, and does not constitute an additional condition. Eq. 4.7.3 limits the upper geometric boundary of the system. A material balance for an incompressible system is translated by the Laplace equation if we think of the capillary layer changing its position in space, but not the mean pressure:

$$\nabla^2 \Phi_c = 0 \quad (4.7.4)$$

Assuming the discretization principles used before, Eq. 4.7.4 could be represented in the following way, considering the entire capillary fringe integrated into a single layer:

$$(q_{in} - q_{out})|_x + (q_{in} - q_{out})|_v = 0 \quad (4.7.5)$$

For convenience, Eq. 4.7.5 can be expressed as function of the potential in the phreatic layer (the one immediately below the capillary layer). The notation used here is the same as for the free surface layer developed in the previous chapter. In fact, it is only a matter of nomenclature.

$$q_{in_x} = \frac{2\pi}{\mu} k_{x_{i+1,1}}^c \cos \beta_{i+1,1} h_c \frac{\Phi_{i+1,1} - \Phi_{i,1}}{\ln \left(\frac{\bar{r}_{i+1}}{\bar{r}_i} \right)} \quad (4.7.6)$$

and

$$q_{out_x} = \frac{2\pi}{\mu} k_{x_{i,1}}^c \cos \beta_{i,1} h_c \frac{\Phi_{i,1} - \Phi_{i-1,1}}{\ln \left(\frac{\bar{r}_i}{\bar{r}_{i-1}} \right)} \quad (4.7.7)$$

In the vertical flow direction, the downward movement of the entire liquid column leaves behind a volume equivalent to an average liquid saturation in the desaturated region. We may consider that the liquid volume going out of the capillary layer through the top boundary must be the same as the equivalent volume of liquid going into the capillary layer through the lower boundary. This means that the capillary layer has no role on the liquid

balance. Hence, the only vertical capillary layer in or out flow to be considered, named q_c , is the net flow through the lower boundary, feeding the top of the phreatic surface. Since there is no cross flow between stream lines, flow is not produced by potential differentials between layers, but merely added to the top of the phreatic layer. Following the same sign convention for flow direction, Eq. 4.7.5 becomes:

$$(q_{in} - q_{out})|_x - q_c = 0 \quad (4.7.8)$$

After substitutions, we may express q_c for a discrete capillary layer block i :

$$q_{ci} = \frac{2\pi h_c}{\mu} \left[k_{x_{i+1,1}}^c \cos \beta_{i+1,1} \frac{\Phi_{i+1,1} - \Phi_{i,1}}{\ln \left(\frac{\bar{r}_{i+1}}{\bar{r}_i} \right)} - k_{x_{i,1}}^c \cos \beta_{i,1} \frac{\Phi_{i,1} - \Phi_{i-1,1}}{\ln \left(\frac{\bar{r}_i}{\bar{r}_{i-1}} \right)} \right] \quad (4.7.9)$$

4.7.3 Mathematical Representation of the Phreatic Layer

The discretization of the phreatic layer will be handled the same way as the free surface layer was handled previously. The difference in the material balance equation is that the flow in is the source term q_c . The same principle of variable thickness will be employed to the phreatic layer as well, and the Eq. 4.1.3 becomes:

$$\Delta_x (T_x \Delta_x \Phi) + \Delta_v (T_v \Delta_v \Phi) + q_{ci} = \frac{1}{\Delta t} \Delta_t (\phi V) \quad (4.7.10)$$

Discretization of a block $i, 1$, ($M > i > 1$):

Following the same exact procedures as in section 4.4, the equivalent residue equation coefficients are:

$$A_{i,1} = -\theta \left[h_{c_D} k_r^o + \bar{\Delta h}_{D_{i+1}} + f_{x_{i,1}} f_{r_i} (h_{c_D} k_r^o + \bar{\Delta h}_{D_i}) + \frac{2k_v A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1} (\Delta h_{D_i} + \Delta z_{D_i})} + \frac{k_h A_{1,i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1}} \frac{\Delta h_{D_i} + \Delta h_{D_i}^k}{2\theta \Delta t_D \cos \bar{\beta}_{i,1}} c_{t_D} \right] \quad (4.7.11)$$

$$C_{i,1} = \frac{2\theta k_v A_{1i}}{k_{x_{i+1,1}} \cos \beta_{i+1,1} (\Delta h_{D_i} + \Delta z_{D_i})} \quad (4.7.12)$$

$$D_{i,1} = +\theta f_{x_{i,1}} f_{r_i} (h_{c_D} k_r^o + \overline{\Delta h}_{D_i}) \quad (4.7.13)$$

$$E_{i,1} = \theta (h_{c_D} k_r^o + \overline{\Delta h}_{D_{i+1}}) \quad (4.7.14)$$

$$\begin{aligned} F_{i,1} = & \frac{k_h}{k_{x_{i+1,1}} \cos \beta_{i+1,1} \cos \bar{\beta}_{i,1}} - \frac{A_{1i}}{\Delta t_D} \left[(\Delta h_{D_i} - \Delta h_{D_i}^k) S_g - \right. \\ & \left. \left(\frac{\Delta h_{D_i} + \Delta h_{D_i}^k}{2} \right) c_{t_D} W_{i,1}^k \right] - (1-\theta) \left\{ (h_{c_D} k_r^o + \overline{\Delta h}_{D_i}^k) [W_{i+1,1}^k - W_{i,1}^k] \right. \\ & + f_{x_{i,1}} f_{r_i} (h_{c_D} k_r^o + \overline{\Delta h}_{D_i}^k) [W_{i-1,1}^k - W_{i,1}^k] \\ & \left. + \frac{2 k_v A_{1i} [W_{i,2}^k - W_{i,1}^k]}{k_{x_{i+1,1}} \cos \beta_{i+1,1} (\Delta h_{D_i}^k + \Delta z_{D_i})} \right\} \end{aligned} \quad (4.7.15)$$

where $k_r^o = k_{x_{i+1,1}}^c / k_{x_{i+1,1}} = k_{x_{i,1}}^c / k_{x_{i,1}}$ is the relative permeability of the liquid in the capillary fringe, at an assumed residual gas saturation.

We find the Jacobian matrix elements for the block $i, 1$ using a Newton-Raphson procedure. The coefficients determined for the Jacobian matrix are those of Eqs. 4.4.14 through 4.4.18, because all the parameters related to the capillary fringe are constant in the present work.

4.7.4 Inner Boundary Phreatic Layer Equation

The capillary layer above the free surface does not produce to the wellbore because of the capillary forces in the sandface. In terms of a discretized block, the equation for the inner boundary blocks is developed using the same concept described in the previous sections for the capillary layer. Skipping the tedious algebraic manipulations, and by analogy with the procedures in section 4.3.2, the resulting residue equation coefficients:

$$A_{1,1} = -\theta \left[h_{c_D} k_r^o + \overline{\Delta h}_{D_2} + f_{x_{1,1}} f_{r_1} \overline{\Delta h}_{D_1} + \frac{2 k_v A_{1,1}}{K_{x_{2,1}} (\Delta h_{D_1} + \Delta z_{D_1})} \right. \\ \left. \frac{k_h A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1} \cos \beta_1} \frac{\Delta h_{D_1} + \Delta h_{D_1}^k}{2 \theta \Delta t_D} c_{t_D} \right] \quad (4.7.16)$$

$$C_{1,1} = \frac{2 \theta k_v A_{1,1}}{K_{x_{2,1}} \cos \beta_{2,1} (\Delta h_{D_1} + \Delta z_{D_1})} \quad (4.7.17)$$

$$E_{1,1} = \theta (h_{c_D} k_r^o + \overline{\Delta h}_{D_2}) \quad (4.7.18)$$

$$F_{1,1} = -\theta f_{x_{1,1}} f_{r_1} \overline{\Delta h}_{D_1} W_{w_1} + \\ \frac{k_h}{k_{x_{2,1}} \cos \beta_{2,1} \cos \beta_1} \frac{A_{1,1}}{\Delta t_D} \left[(\Delta h_{D_1} - \Delta h_{D_1}^k) S_g - \left(\frac{\Delta h_{D_1} + \Delta h_{D_1}^k}{2} \right) c_{t_D} W_{1,1}^k \right] \\ - (1 - \theta) \left\{ (h_{c_D} k_r^o + \overline{\Delta h}_{D_2}^k) [W_{2,1}^k - W_{1,1}^k] \right. \\ \left. + f_{x_{1,1}} f_{r_1} [\overline{\Delta h}_{D_1}^k W_{w_1}^k - (h_{c_D} k_r^o + \overline{\Delta h}_{D_1}^k) W_{1,1}^k] \right. \\ \left. + \frac{2 k_v A_{1,1}}{k_{x_{2,1}} \cos \beta_{2,1} (\Delta h_{D_1}^k + \Delta z_{D_1})} [W_{1,2}^k - W_{1,1}^k] \right\} \quad (4.7.19)$$

We find the Jacobian matrix elements for the block $i, 1$ using a Newton-Raphson procedure.

The coefficients determined for the Jacobian matrix are those of Eqs. 4.4.38 through 4.4.42, because the parameters related to the capillary fringe are constant in the present work.

4.7.5 Outer Boundary Phreatic Layer Equation

At the outer boundary, the capillary layer is passive, and does not have an external source feeding the capillary layer throughout. All liquid feeding the capillary layer comes vertically from the free surface due to capillary forces. If the impermeable top of the

formation allows for the existence of the capillary fringe at that position, that is the origin of the uppermost stream flow lines. In terms of discretized block, the equation for the outer boundary blocks is developed using the same concept described in the previous sections for the capillary layer. Skipping again the tedious algebraic manipulations, by analogy with the procedures in section 4.4.3, the resulting dimensionless equation coefficients are:

$$A_{m,1} = -\theta \left[\overline{\Delta h}_{m+1} + f_{x_{m,1}} f_{r_m} (h_{c_D} k_r^o + \overline{\Delta h}_{D_m}) + \frac{2 k_h A_{1_m}}{k_h (\Delta h_m + \Delta z_{D_m})} \right. \\ \left. + \frac{A_{1_m} (\Delta h_{D_m} + \Delta h_{D_m}^k)}{2 \theta \Delta t_D \cos \bar{\beta}_{m,1}} c_{t_D} \right] \quad (4.7.20)$$

$$C_{m,1} = \frac{2 k_h A_{1_m}}{k_h (\Delta h_m + \Delta z_{D_m})} \quad (4.7.21)$$

$$D_{m,1} = \theta f_{x_{m,1}} f_{r_m} (h_{c_D} k_r^o + \overline{\Delta h}_{D_m}) \quad (4.7.22)$$

$$F_{m,1} = \frac{A_{1_m}}{\Delta t_D \cos \bar{\beta}_{m,1}} \left[(\Delta h_{D_m} - \Delta h_{D_m}^k) S_g - \frac{\Delta h_{D_m} - \Delta h_{D_m}^k}{2} c_{t_D} W_{m,1}^k \right] - \Delta h_{D_m}^k \\ - \theta \left\{ - \left[\overline{\Delta h}_{D_{m+1}}^k + f_{x_{m,1}} f_{r_m} (h_{c_D} k_r^o + \overline{\Delta h}_{D_m}^k) + \frac{2 k_h A_{1_m}}{k_h (\Delta z_{D_m} + \Delta h_{D_m}^k)} \right] W_{m,1}^k \right. \\ \left. + \frac{2 k_h A_{1_m}}{k_h (\Delta z_{D_m} + \Delta h_{D_m}^k)} W_{m,2}^k + f_{x_{m,1}} f_{r_m} (h_{c_D} k_r^o + \overline{\Delta h}_{D_m}^k) W_{m-1}^k \right\} \quad (4.7.23)$$

Again, the Jacobian coefficients are obtained from the partial derivatives. Since the capillary height was the only additional term in Eqs. 4.7.20 - 4.7.23, and because capillary height is considered constant in this study, the results for the derivatives are the same as presented by Eqs. 4.4.50 to 4.4.56.

4.8 Sandface and Wellbore Conditions

The inner boundary conditions of the gravity drive problem are difficult to handle due to the changes occurring on the free surface and stream-lines entering the wellbore. A basic condition presented is the potential along the sandface in a position r_{w+} which depends on the vertical location of the liquid level in the wellbore:

$$\Phi_w = H_w \rho (g/g_c) \Rightarrow z_{r_w+} \leq H_w \quad (4.8.1)$$

$$\Phi_w = z_{r_w+} \rho (g/g_c) \Rightarrow H_w < z_{r_w+} \leq H_s \quad (4.8.2)$$

Using dimensionless parameters, the conditions are:

$$W_w = \gamma_o H_{wD} \Rightarrow z_{wD} \leq H_{wD} \quad (4.8.3)$$

$$W_w = \gamma_o z_{wD} \Rightarrow H_{wD} < z_{wD} \leq H_{sD} \quad (4.8.4)$$

In App. A.1, a discretization in dimensionless form of the preceding conditions is developed, considering all possible relative positions of the liquid level in the wellbore and the grid blocks adjacent to it.

A second relevant condition is given by a material balance on the well. Considering the instantaneous mass flow rate of liquid measured at stock tank conditions as the result of the mass of liquid flowing across the sandface minus the change of mass in the wellbore due to storage, a general equation may be written:

$$q B_{wb} = -\frac{\pi r_w^2 \Delta H_w}{\Delta t} + \frac{2\pi r_w}{\mu} \int_0^{H_s} k_x \frac{\partial p}{\partial r} dz \quad (4.8.5)$$

where k_x is the permeability in the direction tangent to a stream-line at the sandface, and B_{wb} is the volume-formation factor corresponding to the average pressure of the liquid column in the wellbore. Next, the material balance analytical equation Eq. 4.8.5 is converted into dimensionless form. Multiplying both sides of Eq. 4.8.5 by $\frac{\mu}{2\pi r_w k_h h_o \rho (g/g_c)}$, yields:

$$\frac{q \mu B_{wb}}{2\pi r_w k_h h_o \rho (g/g_c)} = -\frac{\mu r_w}{2k_h h_o \rho (g/g_c)} \frac{\Delta H_w}{\Delta t} + \frac{r_w}{k_h} \int_0^{H_s} \frac{k_x}{\partial r} \frac{\partial \left[\frac{p}{h_o \rho (g/g_c)} \right]}{\partial r} dz \quad (4.8.6)$$

Using dimensionless parameters defined previously:

$$q_D B_{wb} = -\frac{\Delta H_{wD}}{2\phi \Delta t_D} + r_{D1} \int_0^{H_{sD}} \frac{k_x}{k_h} \frac{\partial p_D}{\partial r_D} dz_D \quad (4.8.7)$$

Equation 4.8.7 is discretized using a time-relaxation parameter θ . This process follows the geometric characteristics of the stream-layer model. The integral representing flow through the sandface is replaced by a summation of radial steady-state flow through each layer cylindrical region from the geometric center to the sandface.

$$q_D B_{wb} = - \frac{\Delta H_{w_D}}{2\phi \Delta t_D} + \frac{\Delta \bar{z}_{D_1}}{\ln(\bar{r}_1)} \left\{ \frac{k_{x1,1}}{k_h} (1-\theta) (W_{1,1}^k - W_{w1}^k) + \sum_{j=2}^n \frac{k_{x1,j}}{k_h} [\theta (W_{1,j} - W_{w,j}) + (1-\theta) (W_{1,j}^k - W_{w,j}^k)] \right\} + \frac{\Delta \bar{h}_{D_1}}{\ln(\bar{r}_1)} \frac{k_{x1,1}}{k_h} \theta (W_{1,1} - W_{w1}). \quad (4.8.8)$$

It is necessary to define some parameters introduced above. First, the formation-volume factor is calculated based on a time average wellbore pressure taken at half the distance from the top to the bottom of the liquid column:

$$B_{wb} = \frac{V_{wb}}{V_{std}} = \frac{1}{1 + \frac{H_{w*}}{2} \rho (g/g_c) c_L} = \frac{2}{2 + \gamma_o H_{w_D}^* c_{L_D}} = \frac{4}{4 + \gamma_o (H_{w_D} + H_{w_D}^k) c_{L_D}} \quad (4.8.9)$$

Most of the time, B_{wb} may be considered 1 due to the low liquid compressibilities. However, B_{wb} is considered here for complete theoretical formulation. The asterisk superscript in Eq. 4.8.9 represents a time average liquid height. The average block thickness at the wellbore may be taken as the mean of the block thickness at the center (Δh_{D_1}) and the extrapolated thickness at the sandface (Δh_{D_w}):

$$\Delta \bar{h}_{D_1} = \frac{\Delta h_{D_1} + \Delta h_{D_w}}{2}, \quad \text{and} \quad \Delta \bar{z}_{D_1} = \frac{\Delta z_{D_1} + \Delta z_{D_w}}{2} \quad (4.8.10)$$

Also, the dimensionless radial distance \bar{r}_{D_1} from the inner block geometric center to the sandface is used. The drop of the liquid column in the wellbore is:

$$\Delta H_{w_D} = H_{w_D} - H_{w_D}^k \quad (4.8.11)$$

The superscript k stands for the value at the beginning of the time-step.

The discretization of the inner boundary potentials requires previous knowledge of the liquid position in the wellbore. Figure 4.1 shows that the base position of each block is an extrapolation of the coordinates expressed in dimensionless form as:

$$z_{wD_j} = (n - j) \Delta z_{D_w} \quad (4.8.12)$$

where the extrapolated thickness is obtained by:

$$\Delta z_{D_w} = \alpha_w \Delta z_{D_1} + (1 - \alpha_w) \Delta z_{D_2} \quad (4.8.13)$$

and:

$$\alpha_w = \frac{\ln(\bar{r}_2)}{\ln(\bar{r}_2 / \bar{r}_1)} \quad (4.8.14)$$

A similar extrapolation is done to the layer containing the free surface:

$$\Delta h_{D_w} = \alpha_w \Delta h_{D_1} + (1 - \alpha_w) \Delta h_{D_2} \quad (4.8.15)$$

Let j_1 be the position of the inner block adjacent to the liquid surface in the wellbore. The j_1 value is initially unknown, and must be estimated before the calculations. Fortunately, the variation of the liquid level in the wellbore is easy to follow, and this procedure becomes a minor iterative problem. Due to the dependence of the potentials at the sandface on the liquid level in the wellbore, there is a particular way to calculate the average potential at the sandface of each inner block according to the respective stream-layer position. Every case is detailed in App. A.1. It is possible to evaluate the liquid level H_{wD} in the material balance equation by the appropriate substitution of the sandface potentials using an iterative method, or a direct method by solving a quadratic equation.

The partial derivatives of the sandface potentials must be determined in order to be substituted into the Jacobian equations required by the Newton-Raphson method. In view of the changes in the inner boundary condition along the sandface, there is a need for evaluating each case individually. In other words, due to the fact that in the material balance equation all the inner block potentials and sandface potentials are present, a variety of criteria must be considered in the differentiations. These procedures are summarized in App. A.1.

4.9 Computer Model

A general description of approximation of the numerical model described in the preceding is presented in this section. The nonlinear nature of the problem of gravity flow into a wellbore requires an iterative procedure to compute the potentials and the free surface positions in the reservoir, as well as the wellbore potential. Some criteria used in the simulator for a typical time-step calculation are described next:

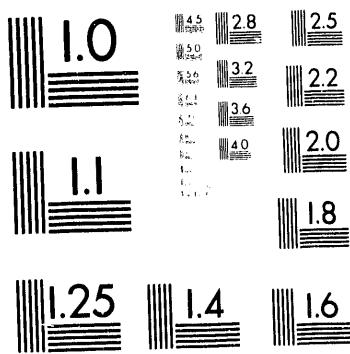
1. At the beginning of a time-step, the model assumes the dimensionless free surface distribution as being the same as that at the end of the previous time-step. At each radial position i , the same number N of vertical blocks is allocated which have an initial uniform thickness calculated by dividing the total height by the constant number of blocks N :

$$\Delta z_{D_i} = \frac{H_{D_i}}{N}. \quad (4.9.1)$$

As an intial guess, the thickness of a block containing the free surface, Δh_{D_i} , takes the same value as Δz_{D_i} :

$$\Delta h_{D_i} = \Delta h_{D_i}^k. \quad (4.9.2)$$

2. During a time-step, the variation of the liquid column H_{D_i} will be the same as the free surface displacement in the correspondent block in the first layer ($j = 1$). All other blocks in the saturated region will not have a change in thickness during this period. In other words, during the time-step loop calculations, Δz_{D_i} remains constant while Δh_{D_i} varies.
3. The wellbore liquid height H_{w_D} is initially estimated from the wellbore material balance equation developed in Appendix A.2, considering the initial potential distribution around the wellbore for the present time-step. The quadratic equation approach was the one preferred for accuracy and speed in the calculations.
4. After calculating H_{w_D} , the sandface potentials W_w , are determined according to the inner boundary condition.
5. The equation and Jacobian matrix coefficients are calculated.
6. The matrix system is solved for the dimensionless normalized potential $w_{i,j}^{v+1}$. The matrix solver used in the model was developed by *Brand et al.* (1992) at Stanford University. The iterative method of solution of the matrix allowed for a maximum of 30 iterations.



2 of 3

7. Convergence is verified for every i, j dimensionless potential and the wellbore liquid height normalized by the initial height:

$$\left| W_{i,j}^{v+1} - W_{i,j}^v \right| \leq \varepsilon, \quad \text{and} \quad \frac{\left| H_{w_D}^{v+1} - H_{w_D}^v \right|}{H_{o_D}} \leq \varepsilon \quad (4.9.3)$$

where ε is an arbitrary small number that may be chosen as an input in the program.

8. If convergence is not achieved, H_{D_i} is recalculated based on potential calculated in the free surface layer:

$$H_{D_i} = W_{i,1} \times H_{o_D} \quad (4.9.4)$$

In this case, the liquid gradient is assumed to be static.

9. The wellbore liquid level H_{w_D} and the dimensionless potential at the sandface W_w , are updated and all other matrix and Jacobian coefficients are re-calculated.
10. The procedures from 2 to 9 are repeated until convergence is obtained. If convergence is not achieved within a given maximum number of iterations, or if the matrix calculation algorithm requires more than the prescribed number of iterations, the program reduces the time-step size and the calculations restart using the values calculated in the previous time-step.
11. After convergence, new block thicknesses are calculated as follows:

$$\Delta z_{D_i} = \frac{H_{D_i}}{n} \quad (4.9.5)$$

12. Interpolate the potential distribution for the new blocks, and update the geometrically dependent parameters such as permeabilities in the direction of flow, average thicknesses, geometrical factors based on the slope corrections for transmissivity and inter-layer distances.
13. The radial number of blocks included in the computation is checked for the necessity of incorporating one more column of blocks. The blocks located at the last column considered in that time step are verified to have a potential drop greater than an arbitrary small number. An additional column of blocks is added if at least one outer boundary block starts to be sensitive to the potential drops.
14. Restart a new time step following the procedures from 1.

In App. E, a detailed description of the computer program is presented, as well as the code list. We turn now to operation of the *Stream Layer Model* and verification of results in Chapter 5.

5. Verification of the Numerical Model

The Stream Layer Model was compared with classical solutions in the ground-water literature, and the results were consistent with available data and traditional methods found in the literature. An appropriate way to verify the wellbore response was to compare the simulator results with steady-state experimental model results performed by *Wyckoff et al.* (1932) and *Hall* (1955). The transient response at observation wells far from the producing wellbore were checked by comparing the simulator results with a classical groundwater method by *Neuman* (1972, 1974), and a numerical solution by *Cooley* (1971). Also, simulator results were compared with the *Theis* (1935) compressible flow line source solution and the p^2 transient solution derived by *Ramey et al.* (1989) to approximate the gravity drainage problem with a free surface. The analyses were done for an isotropic reservoir ($k_h = k_v$) using original data from the references above.

5.1 Verification with Wyckoff, Botset and Muskat Experiments

Wyckoff et al. (1932) investigated the *Dupuit* solution for steady-state gravity drainage flow into a well by means of an experimental sandbox apparatus which is briefly described in Ch. 2. The experiments consisted of several runs at different steady state flow conditions, in which measurements of the head potentials used manometers at the bottom of the sand, and visual inspections of the liquid levels. In the experiments where the sand levels were much above the original liquid levels at the outer boundary, the free surface position was masked by the presence of the capillary fringe. The flow in the capillary fringe was detected by tracing streamlines with ink.

When the capillary fringe was absent, the experimental results verified the *Dupuit* equation to approximate the head distribution at the bottom of the sand, although there were some errors involved in the following assumptions: (a) the liquid moves towards the well in cylindrical shells, and (b) the slope of the free surface was so small that the square of dh/dr could be neglected. Those errors seemed to compensate each other in such a way that Eq. 5.1.1, intended to give the free surface position distribution, was shown to approximate the liquid head at the bottom of the sand:

point	r (cm)	well head (cm)	A
well	6.4	32.50	0.0700
r_1	10.2	33.50	0.0691
r_2	15.3	34.20	0.0663
r_3	20.3	34.80	0.0654
r_4	25.3	35.30	0.0651
r_5	30.5	35.85	0.0674
r_6	35.6	36.20	0.0676
r_7	40.6	36.60	0.0703
r_8	45.7	36.80	0.0691
r_9	50.8	37.00	0.0684
r_{10}	55.8	37.20	0.0685
r_{11}	66.0	37.60	0.0704
r_{12}	76.2	37.80	0.0663
r_{13}	86.4	38.10	0.0681
r_{14}	96.5	38.30	0.0662
r_{15}	106.7	38.40	0.0581
r_{16}	116.8	38.70	0.0662
r_{17}	127.0	39.10	0.1403
r_{18}	137.2	39.10	0.0876
r_{19}	147.3	39.30	
r_{20}	156.0	39.30	

Table 5.1: Determination of the average constant “A” from the *Wyckoff et al.* (1932) experimental Run No. 11, using the Dupuit approach.

$$h_o^2 - h^2 = \left(\frac{q}{\pi k \rho \frac{g}{g_c}} \right) \ln \frac{r}{r_e} \quad (5.1.1)$$

To compare with the *Wyckoff et al.* results, the Stream Layer Model was run from an undisturbed initial condition until reaching steady-state. The permeability for the sandbox experimental Run No. 11 was determined using Eq. 5.1.1 and a plot of $(h_o^2 - h_n^2)$ versus $\ln r/r_e$. To avoid capillary fringe effects in Run No. 11, the top of the liquid was set at the top of the sand level in the sandbox, and the pumping flow rate was kept small enough to avoid a large cone of depression above the free surface. Based on the *Wyckoff et al.* observations, the *Dupuit* approach may be used to calculate the permeability by averaging the values of a constant “A” (see Tab. 5.1) defined as:

$$A = \frac{q \ln \left(\frac{r_o}{r_n} \right)}{h_o^2 - h_n^2} \quad (5.1.2)$$

from which the permeability k was calculated in mD with respect to the 15° sector ($1/24^{th}$ of a complete 360° cylinder):

$$k = \frac{(1000)(24)\mu}{\pi \rho \frac{g}{g_c}} A \quad (5.1.3)$$

For the units used in the experiments, and assuming a liquid viscosity of 1 cp in view of the lack of information about the temperature of the system, Eq. 5.1.3 simplifies to $k(mD) = 7,875,709A$. From the average computations of Tab. 5.1, a permeability of approximately 530,000 mD was calculated for the experiment of run No. 11.

To match Run No. 11, no capillary fringe ($h_c = 0$) was considered in the *SLM*. The results presented in Tab. 5.2 show good agreement of the liquid heads measured at the base of the sand at a steady-state condition with the head potentials calculated by the simulator.

Runs No. 18 and No. 19 were selected to further verify the *SLM*, because of the conditions of flow, where both capillary effects and high vertical velocities were present. The permeability used in the calculations was the same obtained from the experiment Run No. 11. In addition, the only informations available in the paper about the liquid levels observed in the sandbox were those correspondent to Runs No. 18 and No. 19, shown in Fig. 3 in *Wyckoff et al.*. There was no information about the free surface position in *Wyckoff et al.*, but only about the liquid top level (including the capillary fringe) and the liquid heads at the sand bottom. Table 5.2 shows the results of the experiments compared to the results of the simulator. There are acceptable differences of no more of 8 % in the worst case. A plot in Fig. 5.1 shows the observed liquid top position measured directly from Fig. 3 in *Wyckoff et al.* paper, compared with the results from *SLM*.

In the runs presented in Table 5.2, the grid dimensions were those that provided the best fit with the experiments within a reasonable computing time. Also, a no-flow sandface region estimated at about 1 cm, located at the free surface intersection, was considered, due to surface tension effects theoretically expected and confirmed by the *Hall* (1955) work. The surface tension is responsible for this effect, and it happens where the pressure is not enough for the liquid flowing into the wellbore. This reduction in the upper region of the seepage face becomes important when the vertical scale is small, as in Run No. 19. In Run No. 19, the no-flow region could represent 10% of the well liquid level in the reservoir at the sandface.

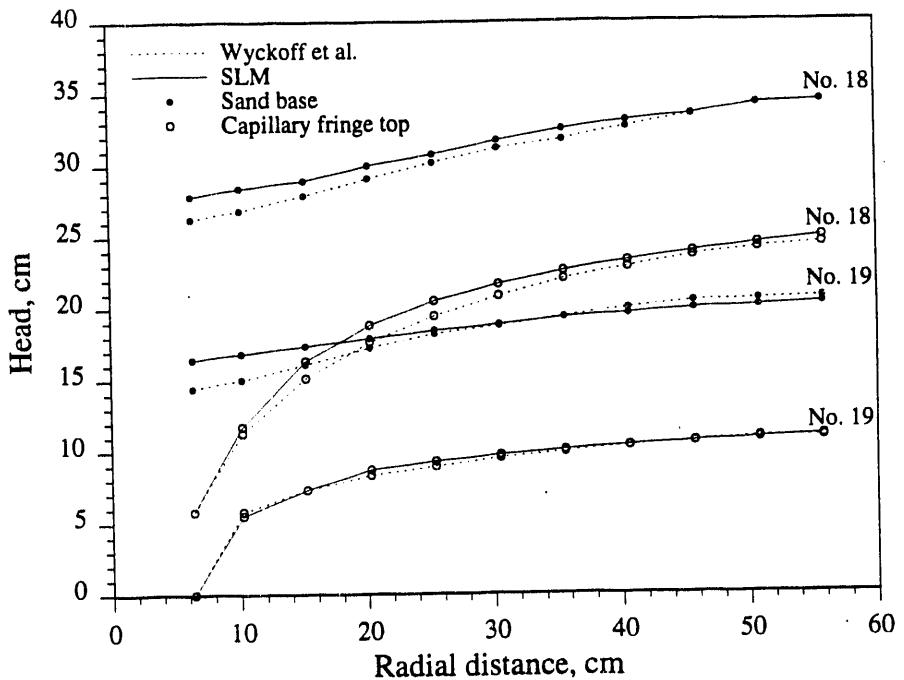


Figure 5.1: Liquid levels and heads in the sand model by *Wyckoff et al.* compared to the *SLM* results of runs 18 and 19.

5.2 Verification with the Cooley and Neuman Solutions

Cooley (1971) solved a problem of gravity drainage dedicated to interference testing numerically neglecting wellbore effects. He applied a finite difference method to solve an integral of the nonlinear partial differential equation obtained by the divergence theorem. Also, the *Cooley* model considered the influence of capillary pressure in the unsaturated region in the capillary fringe. His results, compared with the *Boulton* (1963) convolution integral solution, show that the influence of the capillary fringe is negligible at field scale. The *Cooley* results used in the present comparison were obtained directly from the figures in the paper.

Neuman (1972) used the same synthetic data as *Cooley* to verify his analytical approach not considering the influence of the unsaturated region, and concluded that capillarity "... has very little influence on the time-drawdown response of unconfined aquifers during gravitational drainage". Later, *Neuman* (1974) extended his work to partially penetrating wells. A computer program related to the prior solution, kindly furnished to us by *Neuman*, was applied to generate the present results for comparison. The *Cooley* data, summarized in the following, were used to calculate the head drawdown response at two

Point	r (cm)	Run No. 11			Run No. 18			Run No. 19		
		Wyc.	SLM	%	Wyc.	SLM	%	Wyc.	SLM	%
well	6.4	32.50	32.38	-0.34	5.80	5.80	0	+0.00	0.00	0.00
1	10.2	33.50	33.48	-0.00	11.30	11.72	+3.70	5.80	5.33	-8.10
2	15.2	34.20	34.41	+0.61	15.10	16.19	+7.22	7.30	7.47	+2.33
3	20.3	34.80	35.05	+0.71	17.60	18.65	+5.97	8.30	8.39	+1.08
4	25.4	35.30	35.54	+0.68	19.40	20.28	+4.53	8.90	8.99	+1.01
5	30.5	35.85	35.94	+0.25	20.80	21.47	+3.22	9.50	9.45	-0.53
6	35.6	36.20	36.27	+0.19	22.00	22.38	+1.72	9.90	9.83	-0.71
7	40.6	36.60	36.55	-0.14	22.80	23.11	+1.36	10.30	10.14	-1.55
8	45.7	36.80	36.80	0	23.60	23.74	+0.59	10.60	10.42	-1.70
9	50.8	37.00	37.02	+0.00	24.10	24.28	+0.75	10.90	10.66	-2.25
10	55.8	37.20	37.22	+0.00	24.40	24.74	+1.39	10.90	10.87	-0.27
q (bbl/D)		139.6	136.10	-2.48	284.3	283.9	-0.14	75.5	75.4	-0.13

Table 5.2: Heads (cm) at sand base from *Wyckoff et al.* experimental results compared with the *Stream Layer Model*.

constant pumping rate, q	302 gpm (10,354 BPD)
original height, H_o	59.5 ft
pumping wellbore radius, r_w	0.53 ft
storage coefficient, S	0.034 (dimensionless)
transmissivity, T	89,250 gpd/ft
specific yield, S_y	0.23 (dimensionless)

Table 5.3: Data from *Cooley* (1971)

different radial positions far from the wellbore center. In the example, the two observation points were selected at radial distances of 44.9 and 101.1 ft respectively, to avoid wellbore storage effects, not considered in the *Cooley, Boulton and Neuman* solutions.

The reservoir parameters were estimated from this data - see App. B:

(a) Reservoir porosity ϕ :

$$S_y \cong \phi_e \Rightarrow S_y = 0.23 \Rightarrow \phi = \phi_e = 0.23, \text{ for } S_{w_r} = 0$$

(b) Total Compressibility c_t :

$$S_s = \rho g \phi (c_w + c_f) = \rho g \phi c_t \quad [\text{Cooley (1971), Eq.2}]$$

$$S_s = \frac{S}{h_o} = \frac{0.034}{59.5} = 0.0006 \text{ ft}^{-1}$$

Using appropriate units: $c_t = 0.006 \text{ psi}^{-1}$

(c) Reservoir permeability $k_h = k_v = k$ (single phase isotropic reservoir):

$$T = \frac{k h_o}{\mu} \left(\rho \frac{g}{g_c} \right) = 89,250 \text{ gpd/ft} \quad \Rightarrow \quad k = 73 \text{ D}$$

The Stream Layer model (SLM) was run using the above set of data translated from ground-water parameters to equivalent reservoir properties used in petroleum engineering. The mean radial drawdown $\bar{s}(r)$ was computed by averaging the head potentials along the entire thickness at each radial position of the observation wells, while $s_b(r)$ and $s_h(r)$ were the head drawdowns at the lower boundary and free surface, respectively. The results presented good fit and coherence as observed in the figures below. Also, the applications of the *Ramey et al.* simplified analytical solution showed about the same results as the *Theis* line source solutions at early times, but became closer to the Stream Layer Model as time increased. At early times, both the p-squared and the line source solutions are not able to reproduce the desaturating pressure behavior, as verified in the figures from 5.2 to 5.5. However, since both *Cooley (1971)* and *Neuman (1972)* calculated the average potential for a radial position, the best agreement was found with $\bar{s}(r)$ from the SLM model. At late times, a log-log plot correlates the points closely such that it is difficult for one to observe any difference between all these methods. See Fig. 5.2 and 5.4. The semilog graphs in Figs. 5.3 and 5.5 are appropriate for an accurate view of the late time potentials.

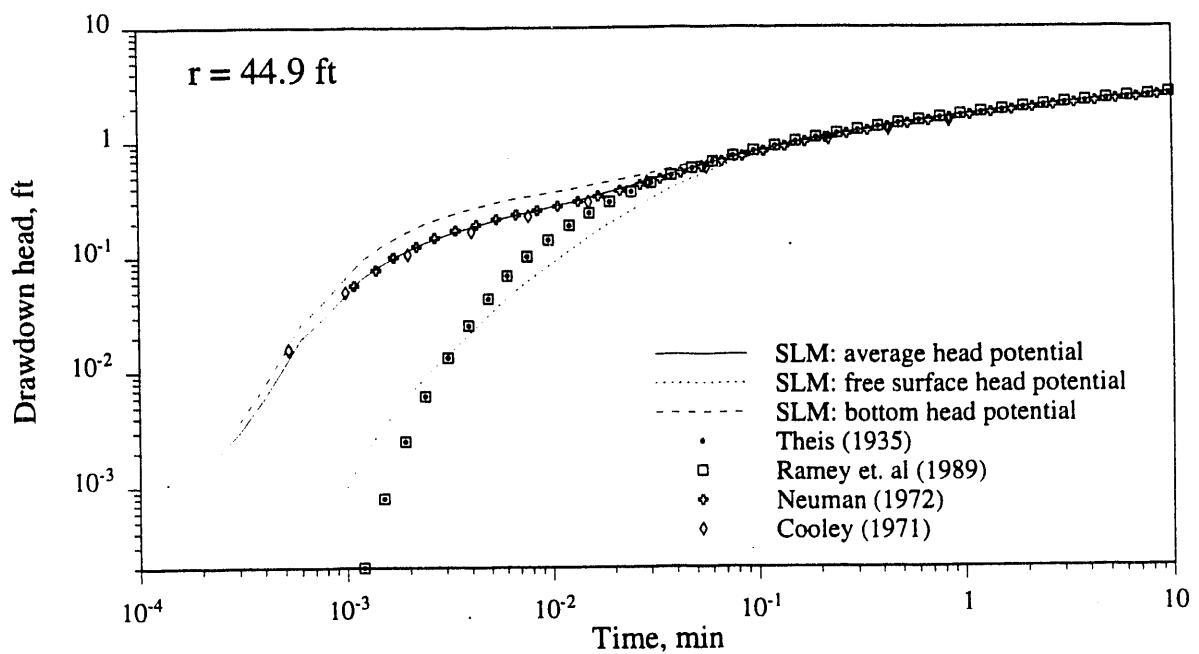


Figure 5.2: Stream Layer Model compared to *Cooley*, *Theis*, *Ramey et al.* and *Neuman*. Observation well at 44.9 ft. Log-log plot.

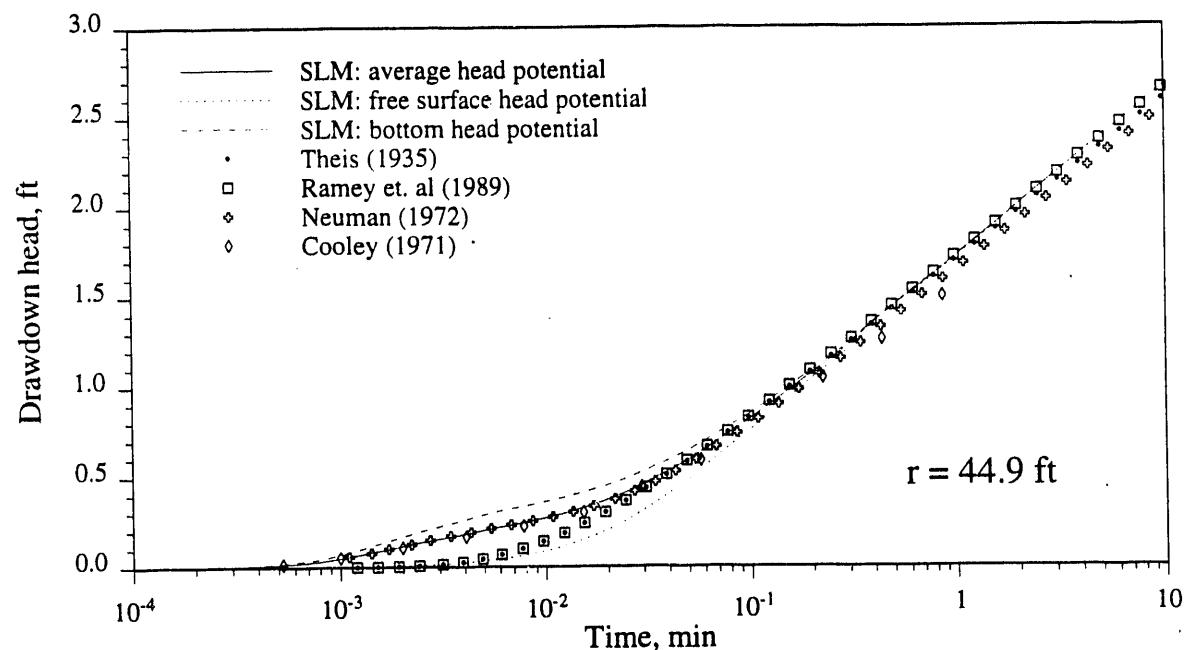


Figure 5.3: Stream Layer Model compared to *Cooley*, *Theis*, *Ramey et al.* and *Neuman*. Observation well at 44.9 ft. Semilog plot.

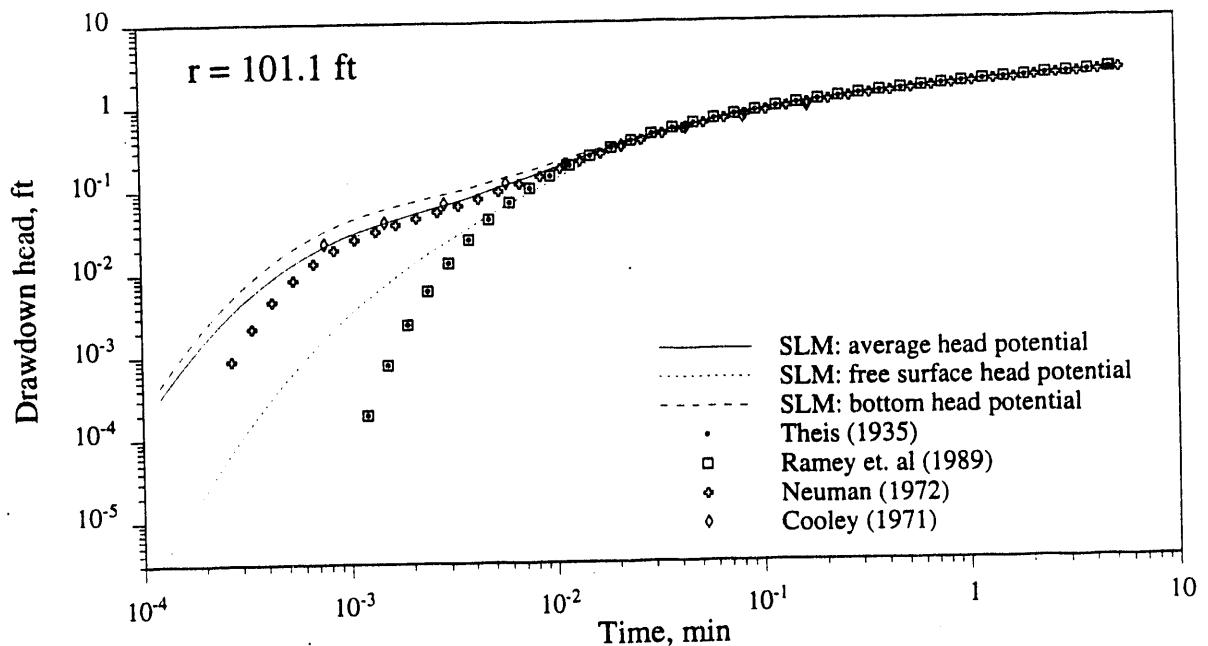


Figure 5.4: Stream Layer Model compared to *Cooley*, *Theis* , *Ramey et al.* and *Neuman*. Observation well at 101.1 ft. Log-log plot.

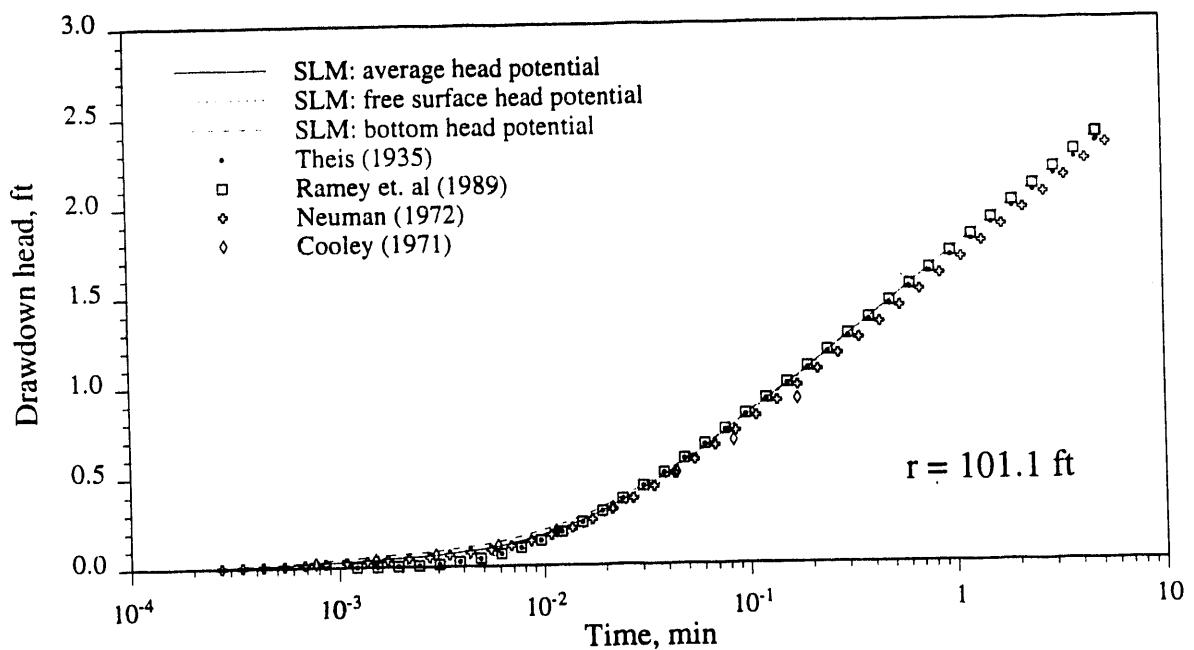


Figure 5.5: Stream Layer Model compared to *Cooley*, *Theis*, *Ramey et al.* and *Neuman*. Observation well at 101.1 ft. Semilog plot.

5.3 Verification with Hall Sandbox Experiments

Another verification compared the *SLM* with the experiments by *Hall* (1955). He performed measurements using a specially designed sandbox apparatus according to Fig. 2.1 in the Literature Review chapter reproduced from the *Hall* (1955) paper. As referenced in Ch. 2, there were four series of tests executed under different conditions, summarized below:

- Series A:
Greatest degree of saturation, close to an ideal condition; increasing flow rate, steady-state steps.
- Series B:
Nonuniform distribution of air voids in the sand by natural resaturation; decreasing flow rates at each steady-state step.
- Series C:
Tests executed in reverse order of series B; better saturation conditions.
- Series D:
Same conditions as in series C, but using reduced surface tension water by chemical treatment.

The experiments of series A were selected to be compared with the *SLM*. The basic data were:

(a) Permeability from direct measurements:

$$k = 0.460 \text{ cm/s} \Rightarrow 453,000 \text{ mD}$$

(b) Water viscosity ¹ at the experiment conditions:

Test No.	Temperature (°C)	Viscosity (cp)
1	26.8	0.86
2	28.5	0.82
3	29.0	0.81
4	29.1	0.81
5	29.5	0.81
6	29.2	0.81
7	29.5	0.81
8	29.6	0.80

¹Calculated by interpolation in Table 6-4 from *Amyx et. al* (1960).

	A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8
grid M x N	12x20							
H_w (<i>Hall</i>), cm	106.4	91.2	76.0	60.8	45.6	30.4	15.2	0
H_w (<i>SLM</i>), cm	107.7	91.4	78.3	62.8	45.5	30.3	16.7	0
% Diff.	+1.32	+0.22	+3.02	+3.29	-0.21	-0.33	+9.93	0
Q (<i>Hall</i>), BPD	1082	2100	2960	3613	4160	4500	4773	-
Q (<i>SLM</i>), BPD	1082	2100	2960	3613	4160	4500	4773	4849
% Diff.	0	0	0	0	0	0	0	-
H_s (<i>Hall</i>), cm	110.7	104.3	94.5	90.2	86.4	84.8	83.3	83.3
H_s (<i>SLM</i>), cm	111.8	100.7	93.8	87.9	83.4	81.0	79.75	79.6
% Diff.	+0.10	-3.45	-0.74	-5.10	-3.47	-4.48	-4.26	-0.84

Table 5.4: *SLM* results compared to the *Hall* series A experiments.

- (c) External radius: $r_e = 197.6$ cm
- (d) Wellbore radius: $r_w = 11.4$ cm
- (e) Porosity: $\phi = 0.40$
- (f) Constant external liquid head: $H_o = 122$ cm
- (g) Capillary fringe thickness: $h_c = 8.9$ cm
- (h) Residual liquid saturation (assumed): 20%

The *SLM* steady-state runs were obtained by establishing both a constant rate, and the corresponding measured wellbore liquid height. The vertical blocks were sized into 20 blocks, keeping the number of radial cylindrical blocks constant at 12. Each run started from a rest (flat liquid table) condition and produced at a constant rate until reaching a steady-state.

From the results of Table 5.4, acceptable agreement was found between the numerical and the experimental methods. All the compared heads and flow rates were fitted with less

Series A-6		Bottom			Free Surface		
Node	Dist., (cm)	Hall	SLM	Diff.,(%)	Hall	SLM	Diff.,(%)
w	11.68	30.48	30.27	-0.68	84.8	81.00	-4.48
1	14.02	38.10	38.07	-0.79	86.9	81.15	-6.62
2	19.20	52.83	51.39	-2.73	90.0	82.44	-8.40
3	26.52	66.29	64.63	-2.50	93.5	85.30	-8.77
4	36.58	78.74	76.89	-2.30	97.3	89.72	-7.79
5	51.51	89.92	88.78	-1.23	101.4	95.52	-5.80
6	71.63	99.06	98.75	-0.56	105.8	101.92	-2.08
7	96.62	106.43	106.76	+0.31	110.7	108.04	-2.40
8	121.92	111.76	112.42	+0.59	114.2	112.83	-1.20
9	146.61	115.57	116.63	+0.92	116.7	116.51	-0.16
10	171.6	119.13	120.10	+0.81	120.1	119.77	-0.33

Table 5.5: Comparison of potential heads between the *Hall* sandbox experiments and *SLM* ($M = 12, N = 24$), series A, Test No.6

than 10% error in the worst case. The greatest differences were observed in calculated and measured free surface positions along the radial distance, as shown in Table 5.5, for the series A-6 experiment. Those positions in the experiment were obtained by interpolation between the values given by gauges located in the sandbox. The best fit for the sink potentials were obtained by including a restriction into the flow along the free surface, probably due to the surface tension at the outlet. Those restrictions were reduced in the tests which the free surface was more developed. As a conclusion, since the discrepancies were within acceptable errors, the Stream Layer Model can be considered verified with the results contained in the present chapter.

In the next chapter, the sensitivity of the *Stream Layer Model* to the grid and time-step dimensions are studied.

6. Sensitivity

In this chapter, some practical aspects of the *SLM* program regarding the simulation approach in wellbore drawdown or buildup pressure behavior will be analyzed, specially the model dimensioning of grid blocks and time-steps. A major difficulty is to handle nonlinearities that occur throughout the reservoir, but are stronger around and at the inner boundary sandface. In some previous models found in the literature, simplifications were generally made to avoid this difficulty which limited applications to regions sufficiently far from the well, or to small flow rates to assure compatibility with some assumptions.

The *SLM* is a finite-difference model that is able to simulate both pumping and observation wells, even those cases that generate large free surface drops. Nevertheless, widely-known restrictions of radial numerical models are still present and amplified by the nonlinear characteristic of the mathematical problem. Stability, convergence and accuracy criteria follow characteristics similar to those of standard confined two-dimensional radial numerical models, tempered by difficulties introduced by the seepage face boundary condition.

6.1 Grid Block Dimensioning

The effects of grid sizing were determined by runs of several configurations of grid dimensions, fixing one direction, and varying the other. A large number of runs was performed from 20 to 60 radial blocks, and 60 to 100 vertical blocks, some results of which are presented in Figs. 6.1 and 6.2. The reservoir and production characteristics were kept constant throughout, and were selected for the present investigation from a field example summarized in Table 6.1. For the radial grid spacing, the *MULTIMODAL* method (*Terán and de la Garza - 1988*) was adopted, using a ratio of 4×10^5 between the first and last radial block dimensions. This ratio was obtained by successive experimental runs in order to get large storage for the first column of blocks (containing the sandface) and keep a logarithmic spacing. An infinite reservoir was simulated by setting the external radius far enough that drawdown effects were not evident.

The numerical treatment of the partial differential equations in the gravity flow problem introduces discretization errors in the potential solutions that normally tend to decrease with grid refinement. Figures 6.1 and 6.2 show semilog plots of the wellbore

pressure drawdown comparing the effect of grid sizing on the simulator response for a range of grid sizes which produced acceptable computing times. In the range of meshes studied, grid dimensions slightly affected the potential response only after the desaturating period when unconfined flow behavior was no longer effective. In the examples of Fig. 6.1 and Fig. 6.2, the potentials in the wellbore show no considerable differences for gross or fine meshes.

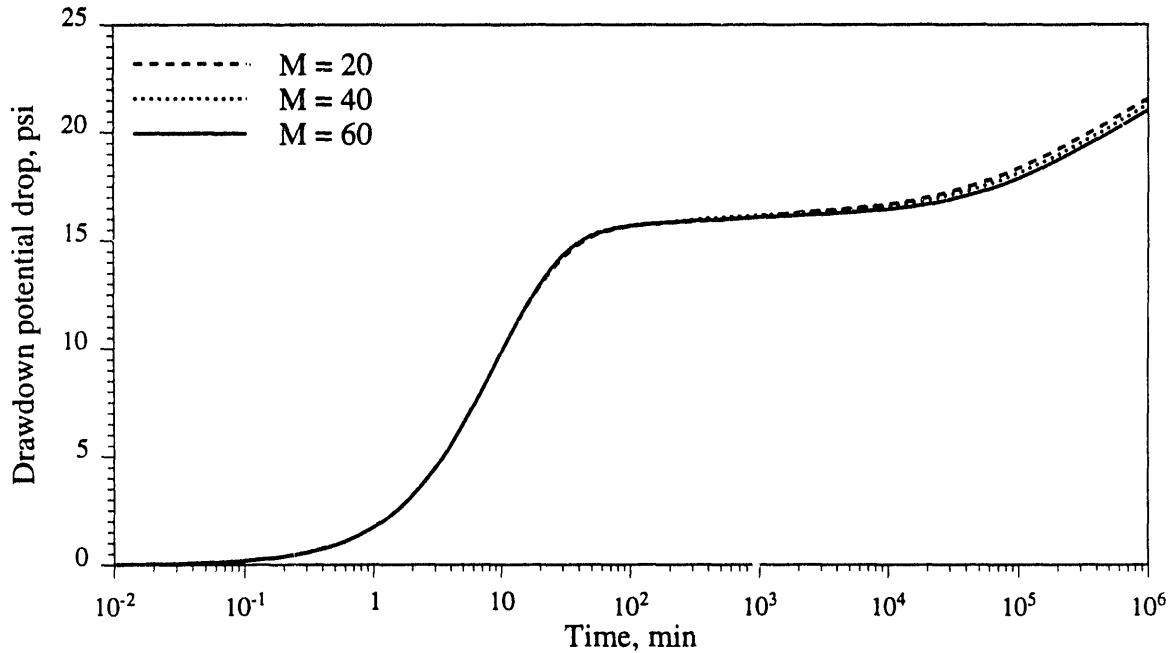


Figure 6.1: Radial mesh variation, 100 vertical grid blocks

The computing time for the drawdown period observed in Fig. 6.1 and Fig. 6.2 varied in the following way: for the horizontal mesh variation, the CPU time was increased by a factor of 3.3 when the grid number varied from 20 to 60 in the radial direction; for the vertical gridding, a refinement from 60 to 100 blocks increased the computation time by a factor of 2.7. An obvious but important limitation to grid refinement is computing time. For a very fine grid mesh, the CPU time becomes prohibitive to reach long drawdown or buildup times. Since we are concerned with the wellbore response, an additional difficulty is the fact that the desaturating period is sometimes long requiring that wells reach pseudoconfined behavior after months or years of pumping.

Some reservoir property relationships in the groundwater literature, mainly the *Boulton* parameter $\beta = k_v/k_h(r/h_o)^2$, are related to the time when the unconfined period is followed by the pseudoconfined period, as observed from the *Boulton* (1963) and *Neuman*

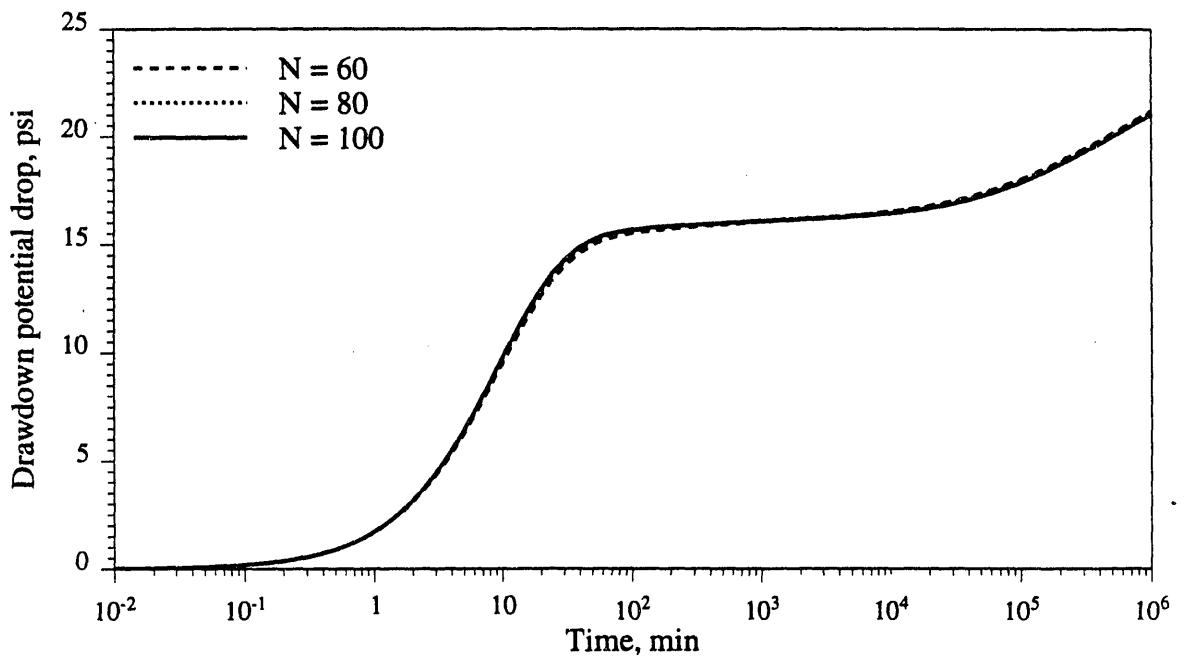


Figure 6.2: Vertical mesh variation, 60 radial grid blocks

(1975-a) type curves. According to these approaches, when the potential drawdown starts following a pseudoradial solution, the dimensionless time increases as β decreases. In the wellbore ($r = r_w$) for example, practical values of β are much smaller than 0.001, the lowest value shown in the type-curves. As a consequence of the delay of the pseudoradial response, simulation of the flowing period requires long production time, sometimes taking months of producing time. However, at late times when the potentials near the well have large changes, computations require long computing time for the reduced time-step required by the numerical solution. Since the grid blocks around the well start having large variations in the potentials, the combined effects of geometry and time-step sizes influence computation performance.

To recall some calculation procedures of the program, the Jacobian matrix is solved iteratively in the Newton method. Only part of the blocks are included in the simulation at the beginning: those located in the reservoir region affected by the production. Since the potential perturbation far from the wellbore is less than an arbitrarily small value, it is not necessary to include distant grid blocks in the calculations, thus saving computing time. The matrix solver requires approximately 80 to 93% of the computing time to calculate a complete iteration loop depending on the number of vertical and horizontal grid blocks. The number of vertical grid blocks is the most important factor due to the correspondence with the bandwidth of the matrix, according to the formulation of the matrix solver.

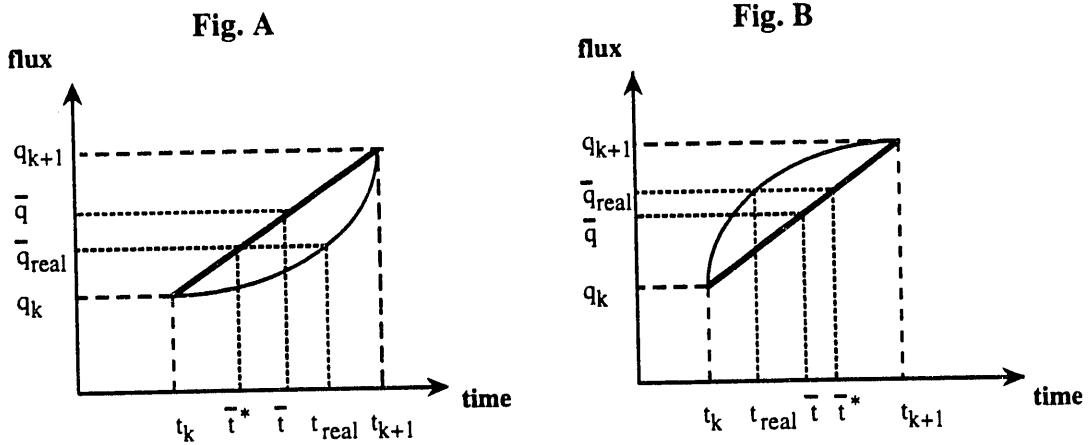


Figure 6.3: Nonlinear flux variation in a block boundary during a time-step. Acceleration and deceleration periods.

6.2 Time Weighting and Time Step Length

A major reason for errors in the numerical model is the linearization imposed on the nonlinear problem. As a consequence, accuracy is related to the time-step size and the time-interpolating parameter, besides the grid mesh size. The time-interpolating parameter is important, and gives the degree of implicitness of the unknowns. In the mathematical formulation of the present numerical approach, the potential variation during a time-step is assumed linear, and the calculation of the material balance for an element considers a constant time weighting factor θ :

$$\theta [q_{in} - q_{out}] + (1 - \theta) [q_{in}^k - q_{out}^k] = \frac{\Delta V}{\Delta t} \quad (6.2.1)$$

where k refers to the previous time-step.

Figure 6.3 shows the flux in or out of a grid block element during a time-step from t_k to t_{k+1} . Depending on the true shape of the curve and the time-step size, the flux approximated by linear interpolation may be under or over-estimated. For example, since during the drainage process the difference $q_{in} - q_{out}$ is almost zero (low compressibility systems - saturated region) or negative (free surface region), the material balance error introduced by the linearization may gradually increase when the flux through a block boundary is accelerating, and decrease when the flux is decelerating, as seen in Fig 6.3. At long times, the errors in both directions may compensate or attenuate the apparent deviations from the correct solution.

In the verification runs of the sandbox experiments, as steady-state conditions were

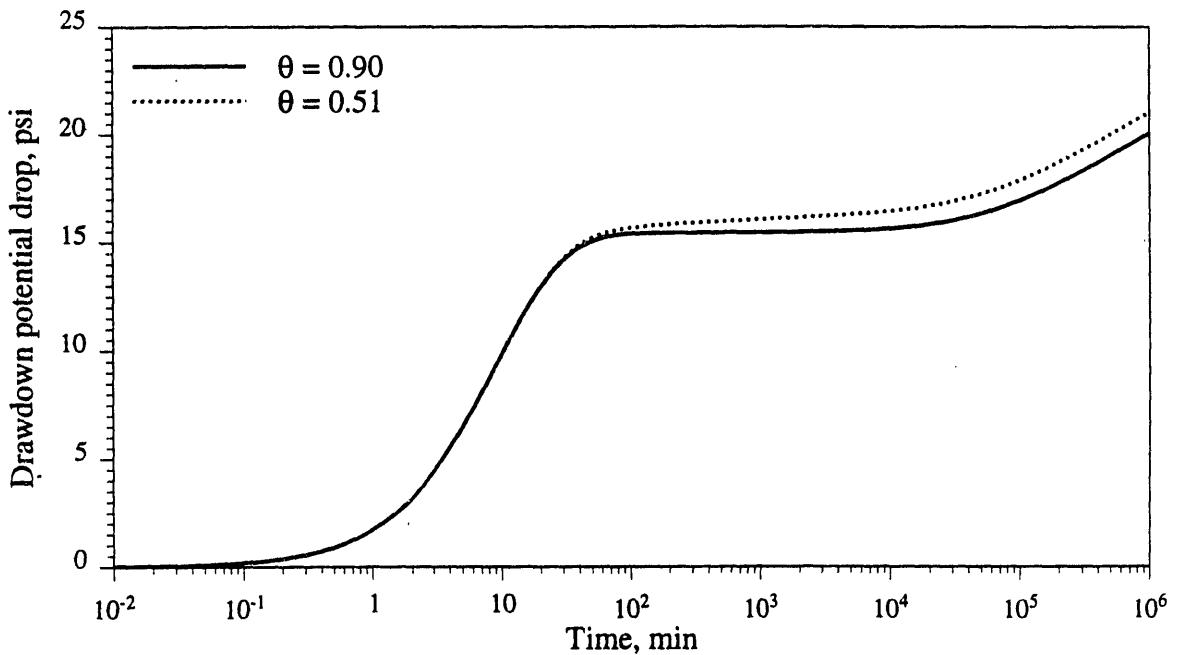


Figure 6.4: Effects of time interpolation parameter in the SLM results

reached, the Crank-Nicolson method ($\theta = 0.5$) developed oscillating problems, and a very large number of time-steps was required to avoid oscillations and a large number of low convergence iterations. The fully-implicit scheme in time calculations ($\theta = 1$) was then tested, and after reaching steady-state conditions, stability and convergence were obtained with any time-step size. In this case, the drawdown results were a little underestimated, compared with the Crank-Nicolson scheme. Figure 6.4 shows a graph containing the same data input run with different interpolation parameters θ . For $\theta = 0.5$, the computing CPU time was as high as 900 minutes to reach 10^6 minutes of production time due to the exaggerated number of iterations generated by the oscillation in the results. These oscillations were avoided automatically by reducing the time-step sizes as necessary.

Based on the observations reported in the preceding, practical conclusions are that time-step dimensioning is important for convergence and accuracy, and that the interpolation parameter θ may vary from 0.50 to 1. However, for improving the computing time, the Crank-Nicolson scheme should sometimes be avoided.

Bearing in mind that at a given time each particular region of the reservoir behaves according to one of three possible flow regimes, namely wellbore storage, desaturating cone, and pseudoconfined, the time-step sizes are dimensioned based on the most unfavorable conditions caused by the rate of potential changes with time. Since small time-steps make

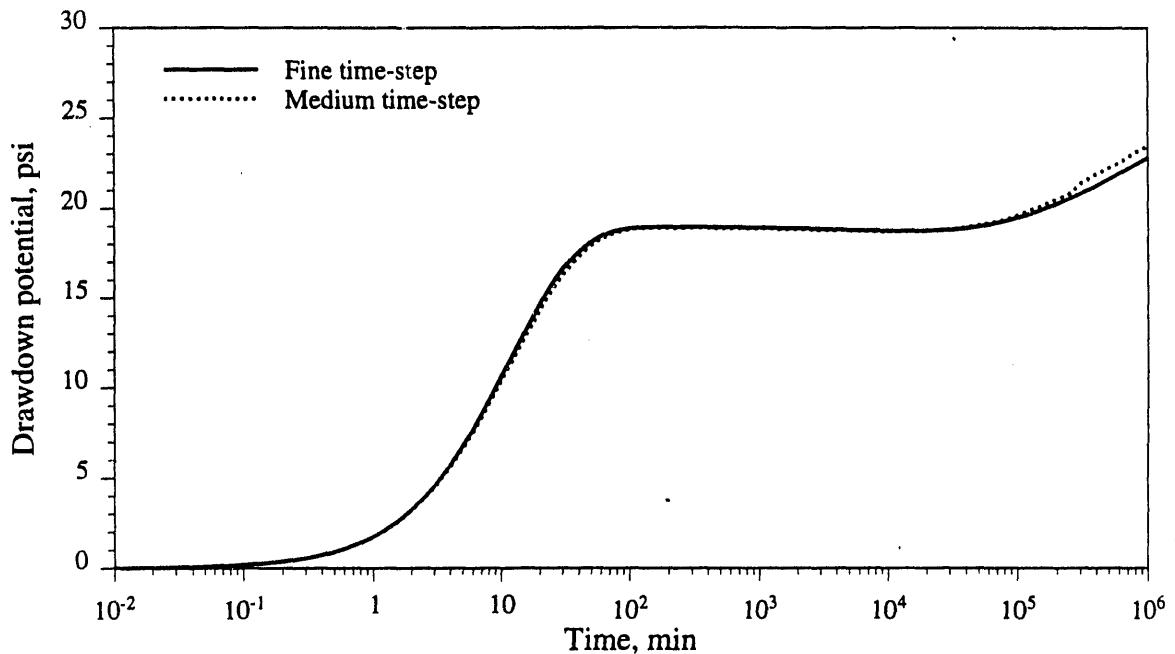


Figure 6.5: Effects of time-step size in the wellbore potential drawdown

the computations long and expensive, some criteria were adopted in the program to set time step sizes in a range such that time steps were maximum under an accuracy point of view, and minimum to achieve a reasonable computing time.

At early time during the wellbore storage dominated period, the flow rate variations through the blocks and into the wellbore were similar to that represented in Fig. 6.3-A. That is, the linearization over-estimates the fluxes in the boundaries ($\bar{q} > \bar{q}_{real}$). Noticeably q_{out} is affected more than q_{in} . As a consequence, the calculated drawdown pressures have a tendency to increase further, due to a material balance error. As time increases, if the deceleration process (Fig. 6.3-B) takes over, or a less nonlinear behavior starts, the error is reduced. Inside the reservoir, as we go far from the wellbore, the process explained before is repeated at different times and places, but it is really near the well where this phenomenon is most important.

Figure 6.5 is a graph showing the effects of the time-step size in the results during the drawdown period. Three different time-step sizes were compared. A very large time-step size case did not converge after $t_D = 100$ (not on Fig. 6.5) when the wellbore storage period had finished and the desaturating surface (free-surface downward movement) started to accelerate. A medium time-step size was used in a second run, and a fine time-step size was used as a reference. The results observed in the plots of Fig. 6.5 were consistent with

expectations.

The most critical problem with time-step dimensioning in the given examples starts in the transition between the desaturating and the “pseudoconfined” periods. We define “pseudoconfined” as the drawdown behavior that follows the period in which the free surface downward movement becomes approximately uniform, after the formation of the desaturating cone around the wellbore. In that transition, the pressure drawdown curve inflects upward, and the linearization tends to over-estimate the fluxes (see Fig. 6.3), increasing the pressure drawdown by material balance error accumulation as observed in Fig. 6.5. Conservative time-steps are recommended during the pseudoconfined period during drawdown or buildup because of the grid block sizes related to the rate of potential change.

An empirical rule of thumb derived from an extensive number of runs is to start runs with a $t_D = 0.01$ for simulating field cases, and $t_D = 0.0001$ for laboratory experiments, and to increase the number of time-steps per log cycle by 10 or 100 at each new time log-cycle. A key for this choice was to observe the sharpness of the vertical potential gradient in the formation near the well, and the dimensionless block sizes compared to the fluxes. The lower the mobility k/μ , or the higher the production flow rate, the smaller the required time-step sizes may be.

The time-step size can not be separated from the block sizes since large increases in the time interval may cause stability problems. The blocks located around the wellbore are those of concern, because they have low storage capacity. The *SLM* simulator has an automatic time-step control that helps to guarantee a time-step size reduction at any time when it is necessary. In other words, after a maximum number of iterations, because of problems with matrix solver convergence, lack of convergence in the wellbore pressure calculation, or exaggerated free surface movement, the time-step size control algorithm changes time step sizes gradually until acceptable values dictated by the built-in convergence criteria result. The maximum number of iterations, *itermx*, is generally input data, but can also be a variable in the program. A variable number departing from 3 and increasing by 1 at each new time log cycling has proven to be efficient, avoiding exaggerated reductions in the time-step sizes.

In terms of convergence, buildup behavior is poorer than drawdown behavior. As seen in Ch. 7, the late time potential buildup behavior is more nonlinear than drawdown behavior. Thus, the time-steps must be reduced. For example, computing time for 10^6 minutes of a fine 80x100 grid drawdown followed by a same buildup period, as in Fig. 7.9, took more than 1000 CPU minutes in an IBM RISC 6000/550, 50% of the time consumed in the last buildup logarithmic time cycle.

In the next chapter, a discussion of the effects of some reservoir properties on production in a gravity drainage well test is presented.

External dimensionless radius, r_e/r_w	10^6
Initial liquid level at rest	189 ft
Isotropic permeability (single phase)	80 mD
Liquid viscosity	1 cp
Liquid compressibility	$3 \times 10^{-6} \text{ psi}^{-1}$
Skin	0
Formation compressibility	0
Liquid density	62.4 lb/ft^3
Formation volume factor	1.0 res vol/std vol
Production flow rate	250 bbl/day
Porosity	0.25
Residual saturation in the desaturated region	0

Table 6.1: Basic input data for the sensitivity runs

7. Discussion

In attempts to analyze drawdown and buildup behavior for a low-pressure production well, it became evident that free surface gravity drainage was a major factor, *Ramey et al.* (1989). A literature review indicated no pertinent interpretative methods in the petroleum literature, and contradictory methods for pumping well data interpretation in hydrology, *Boulton* (1963) and *Neuman* (1975-a,b). There even appeared to be problems with interference data interpretation. Although interference data matched the *Theis* (1935) line source solution, the apparent storativity (or compressibility) from type curve matching was hundreds of times larger than the known compressibility of water. This led some authors to speculate that trapped gas increased apparent compressibility. It appeared to *Ramey et al.* (1989) that using a confined system model was simply incorrect for free surface drainage of an essentially incompressible flow system. It is easy to compute that maximum production of water by expansion from a low pressure (100 psi) aquifer is less than 0.1% of the water in place.

It also appeared to *Ramey et al.* (1989) that a solution of the Dupuit-Forchheimer equation similar to an early natural gas flow problem by *Jenkins and Aronofsky* (1953) could aid understanding. This study showed an incompressible liquid could match the line source solution exactly. However the parameter which appeared in dimensionless times was not compressibility, but the reciprocal initial head, which is much larger than the compressibility of water for a low pressure system.

During analysis of the Dupuit-Forchheimer problem, review of the *Wyckoff et al.* (1932) study convinced *Ramey et al.* (1989) that the Dupuit-Forchheimer model violated physical principles and could not match the free surface behavior, or the existence of a seepage face in a producing well. As *Muskat* (1937) clearly stated, analytical solution of the nonlinear free surface problem was essentially hopeless. Thus it was decided to produce a finite-difference computer model which could be used for investigation of behavior of both producing and observation wells. The report concerns construction of a program capable of solving this formidable nonlinear problem.

To explore the characteristics of the *Stream Layer Model* to simulate a gravity drainage well, drawdown and buildup pressure behaviors are presented in this chapter. Also, influences of some parameters were investigated. Skin effect, permeability, flow rate, permeability ratio (anisotropy) and the initial position of the liquid level at rest have been considered in the following sections. A basic set of data used during this study is presented in Table 7.1.

Dimension for horizontal grid (mdim)	60 or 80
Dimension for vertical grid (ndim)	100
Initial number of time-steps per cycle (ntd)	10
Initial dimensionless time (tdi)	10^{-4} or 10^{-2}
Dimensionless external radius (r_{eD})	10^5 or 10^6
Liquid compressibility (c_L)	$3 \times 10^{-6} \text{ psi}^{-1}$
Total compressibility (c_t)	$3 \times 10^{-6} \text{ psi}^{-1}$
Original height of the free surface (H_o)	189 ft
Initial pressure (p_i)	81.9 psi
Well bore radius (r_w)	3.18 in.
Porosity - fraction (ϕ)	0.25
Liquid viscosity (μ):	1 cp
Liquid density ($\rho g / g_c$)	62.4 lbf/ft ³
Formation volume factor (B_o)	1
Skin effect parameter (s)	0
Horizontal permeability (k_h)	120 mD
Vertical permeability (k_v)	120 mD
Constant production rate (q)	100 bbl/d
Time-relaxation parameter ($1 \geq \theta \geq 0$)	0.9
Residual saturation in the desaturating region (S_{rL})	0
Capillary fringe thickness (h_c)	0
Error tolerance (ϵ)	10^{-6}

Table 7.1: Basic data used as input to the *SLM*. Equivalent variables used in the program are shown between parentheses.

The *Stream Layer Model* (SLM), unlike the available analytical methods in the literature, is able to represent the wellbore and formation pressure (head) behaviors during drawdown and buildup periods, since the most important limitations of this single-phase simulator is the discretization of a numerical solution, regardless of the consideration of unsaturated two-phase flow. Other approaches, such as *Theis* (1935) and *Ramey et al.* (1989), have been used in this chapter for comparison at late times. Two auxiliary programs, namely *FLIGRAM* and *PDE* were used in some cases to consider wellbore storage and skin effects in the *Ramey et al.* and *Theis* solutions (see Appendix C). These modifications were not intended to represent the desaturating period, since both *Theis* and *Ramey et al.* neglected vertical flow in the reservoir.

7.1 Transient Wellbore Pressure Analysis

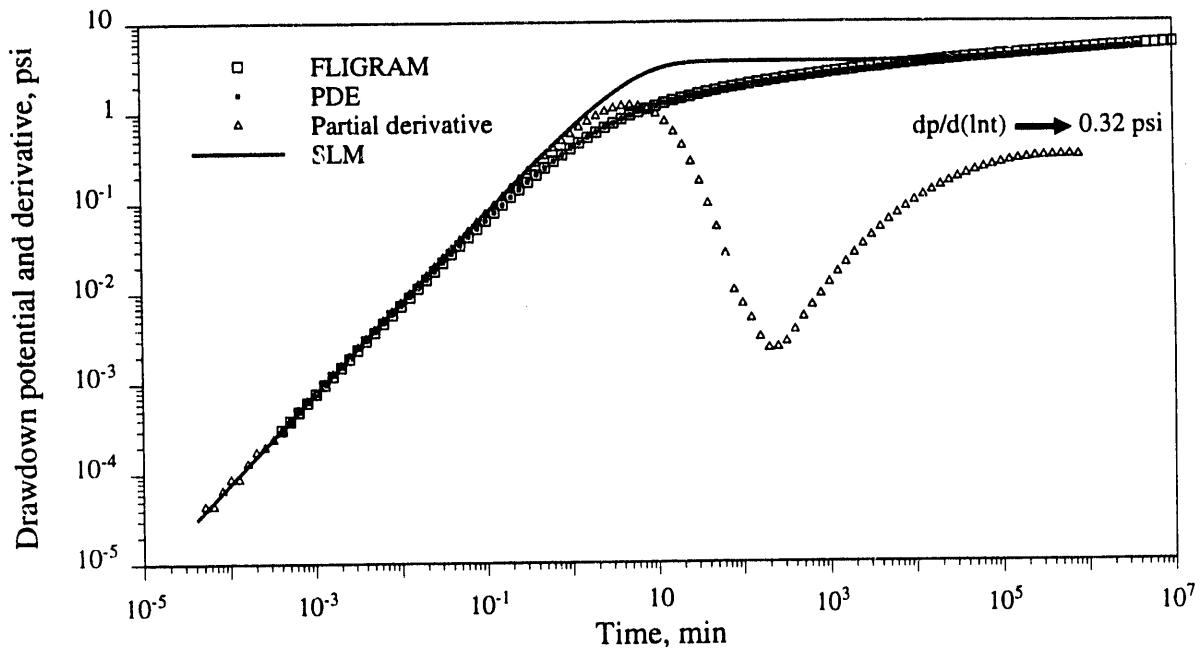


Figure 7.1: Wellbore flowing pressure and derivative log-log plot.

Using results of the *SLM* simulator, general observations of transient gravity drainage flow into a well are considered in this section, including drawdown and buildup behaviors of a typical oil field well. The basic reservoir properties are shown in Table 7.1. A complete run of the *SLM* is described and analyzed in the next subsections.

7.1.1 Drawdown

Figures from 7.1 to 7.7 present results from the *SLM* simulator for a flow period of 10^6 minutes (278 days) for a single well located in an infinitely large reservoir. As described in the groundwater literature, the *SLM* model results show three flow regimes during the flow period, namely wellbore storage, desaturation, and the pseudoconfined or pseudoradial flow. See Figs. 7.1 and 7.2.

Wellbore storage dominates during the first minute, according to the log-log plot of wellbore potential vs. time in Fig. 7.1. The potential derivative $\frac{\partial \Delta p_w}{\partial \ln t}$ is also shown in the same graph. From the definition of the dimensionless wellbore storage coefficient in the petroleum engineering literature (see *van Everdingen and Hurst*, 1949), we present a similar definition based on the liquid level variation in the well:

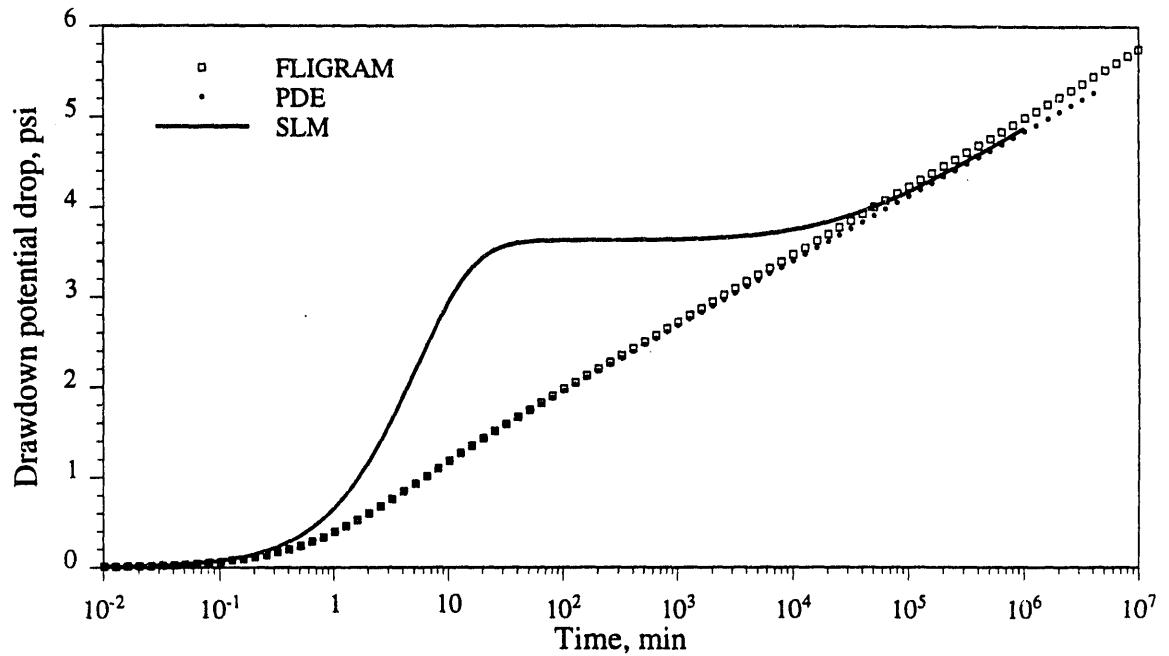


Figure 7.2: Wellbore flowing pressure semilog graph

$$C_D = \frac{C}{2\pi\phi r_w^2} = \frac{\frac{\pi r_w^2}{\rho g/g_c}}{\frac{2\pi\phi r_w^2}{\rho g/g_c}} = \frac{1}{2\phi} \quad (7.1.1)$$

According to Agarwal *et al.* (1970), the early time dimensionless pressure in the wellbore storage period is:

$$p_{wD} = \frac{t_D}{C_D} \quad (7.1.2)$$

In the following, we define a non-normalized p_{wD} and rewrite the definition of t_D in Ch. 3 in petroleum field units:

$$p_{wD} = \frac{k H_o}{(141.2) q B \mu} \Delta p_w \quad (7.1.3)$$

and:

$$t_D = \frac{(0.0002637) k t}{\phi \mu (1/p_i) r_w^2} \quad (7.1.4)$$

Using time in minutes, and defining the liquid gradient $\gamma = \frac{p_i}{H_o}$, we combine Eqs. 7.1.3, 7.1.4 and 7.1.2:

$$C_D = \frac{(0.0002637)(141.2)}{(60)} \frac{\gamma q}{\phi r_u^2} \frac{t'(\text{min})}{\Delta p}. \quad (7.1.5)$$

In the log-log plot of p_w vs. time in Fig. 7.1, we identified a straight line at early times with a unit slope as expected for a wellbore storage period. A given point at 0.1021 minutes corresponds to a pressure drop of 0.0766 psi (values read on the output list of the program). Using the input data of Table 7.1, the resulting dimensionless wellbore storage coefficient was calculated as 0.51 from Eq. 7.1.5. This value is close to the theoretical value of 0.50. Due to wellbore storage effects, the early time line source solution used by *Boulton* (1954-b, 1963) is not appropriate. Another qualitative observation is that the duration of the unit slope straight line corresponding to wellbore storage effects in the log-log plot in Fig. 7.1 is greater in the *SLM* solution than in either the *FLIGRAM* or *PDE* solutions.

As the wellbore storage effects are replaced by the formation response to pumping the well, the potential in the wellbore becomes almost constant, as verified in Figs. 7.2 and 7.1. This is the period during which the desaturating cone is intensely developed around the well, called the “delayed yield” period in the groundwater literature.

After the desaturating period, the semilog graph in Fig. 7.2 shows a potential behavior in the wellbore similar to the *Ramey et al.* (1989) and *Theis* (1935) solutions obtained from *FLIGRAM* and *PDE* programs. This is the pseudoradial flow or pseudoconfined radial flow period. This behavior occurs when the formation of the desaturating cone is stabilized and the flow regime becomes dominated by regions of the reservoir distant from the well where the streamlines are almost horizontal. For a quantitative evaluation of the late time potential response, and for a comparison of the *SLM* results with the *Ramey et al.* (1989) method, a p -squared vs. semilog time drawdown graph is presented on Fig. 7.3. In this figure, a late time m -permeability slope value of $114 \text{ psi}^2/\text{log-cycle}$ was obtained graphically. From the analytical approach, the difference was only 3%:

$$m = 325.2 \frac{q B \mu}{k(h/p)} = 117.4 \text{ psi}^2/\text{cycle}.$$

In the traditional semilog *Theis* (1935) analysis, the late time pressure drawdown slope in Fig. 7.2 of an apparent straight-line was $0.672 \text{ psi}/\text{log-cycle}$, 6.6% smaller than the theoretical value of $0.716 \text{ psi}/\text{log cycle}$:

$$m = (162.6) \frac{q B \mu}{k h} = 0.716 \text{ psi}/\text{cycle}.$$

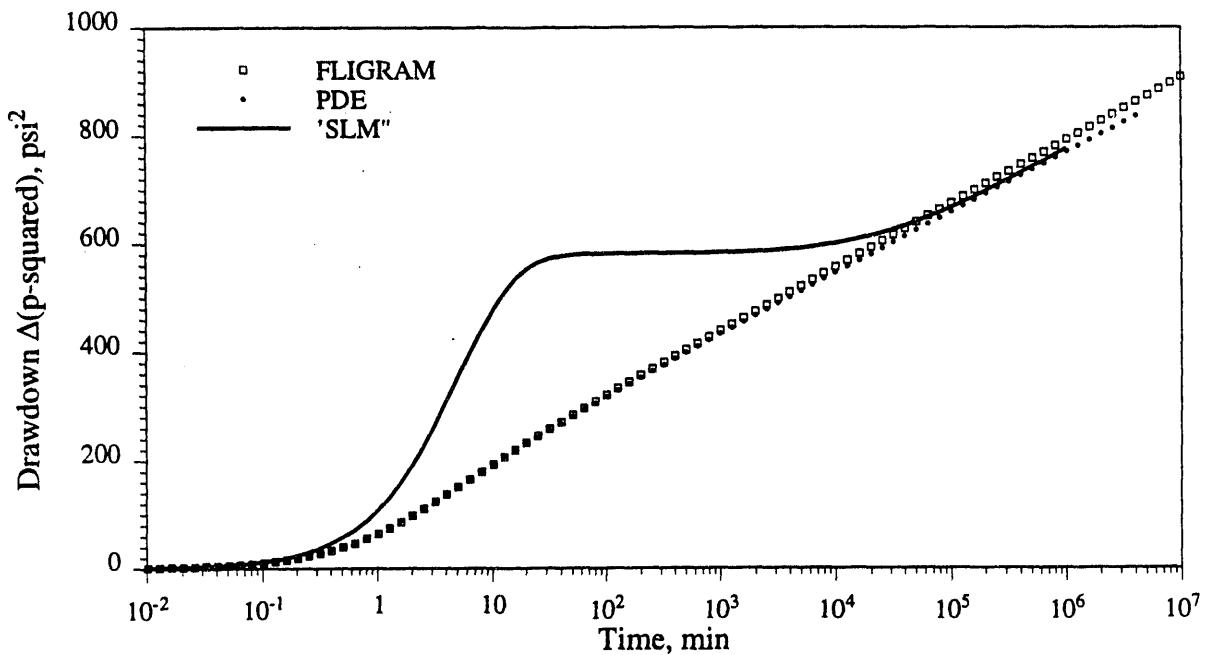


Figure 7.3: Wellbore flowing pressure-squared semilog plot

Another quantitative verification was made by calculating the derivative of dimensionless pressure obtained from the dimensional asymptotic value in the graph of 0.32 psi. Based on the input data, the conversion factor for the dimensionless pressure in Eq. 7.1.3 was 1.606 psi^{-1} . The resulting partial derivative $\frac{\partial p_{wD}}{\partial \ln t_D}$ was 0.51, close to the theoretical value of 0.5.

A graph in Fig. 7.4 shows the flow rate distribution along the sandface blocks of the *SLM* for the flow period. Relative flow rate profiles are shown for both the desaturating and the pseudoconfined periods for 100 vertical grid blocks. There is always a concentration of flow in the vicinity of the liquid level in the well indicating high flow velocities in that region. In addition, the flow distribution is more uniform at late times. In all observations, the contribution of the seepage face gradually increases from zero at the free surface position to a maximum value at the wellbore liquid level. The flow variations are small below the liquid level in the wellbore, mainly at late times, as we go far radially from the liquid level in the well.

Figure 7.5 shows the seepage face behavior with time. This is an important result from the present study which has not been found in any of the published papers in the surveyed literature. The seepage face presents an interesting transient behavior, following the three distinct periods of flow in a gravity drainage well. In this example, the seepage

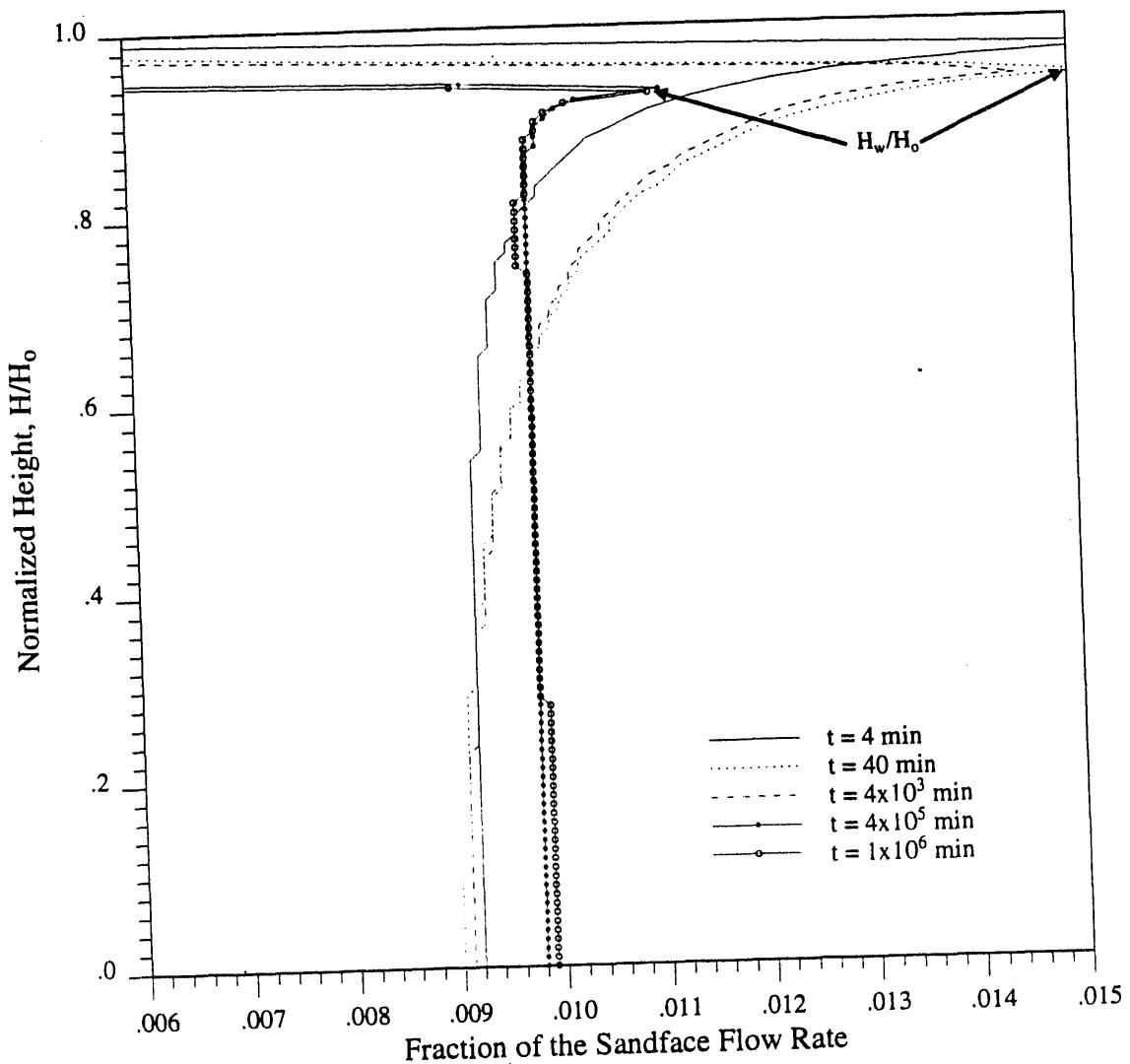


Figure 7.4: Sandface flow rate distribution. The liquid levels in the well coincide with the high flow rate values. See examples indicated by arrows in the figure.

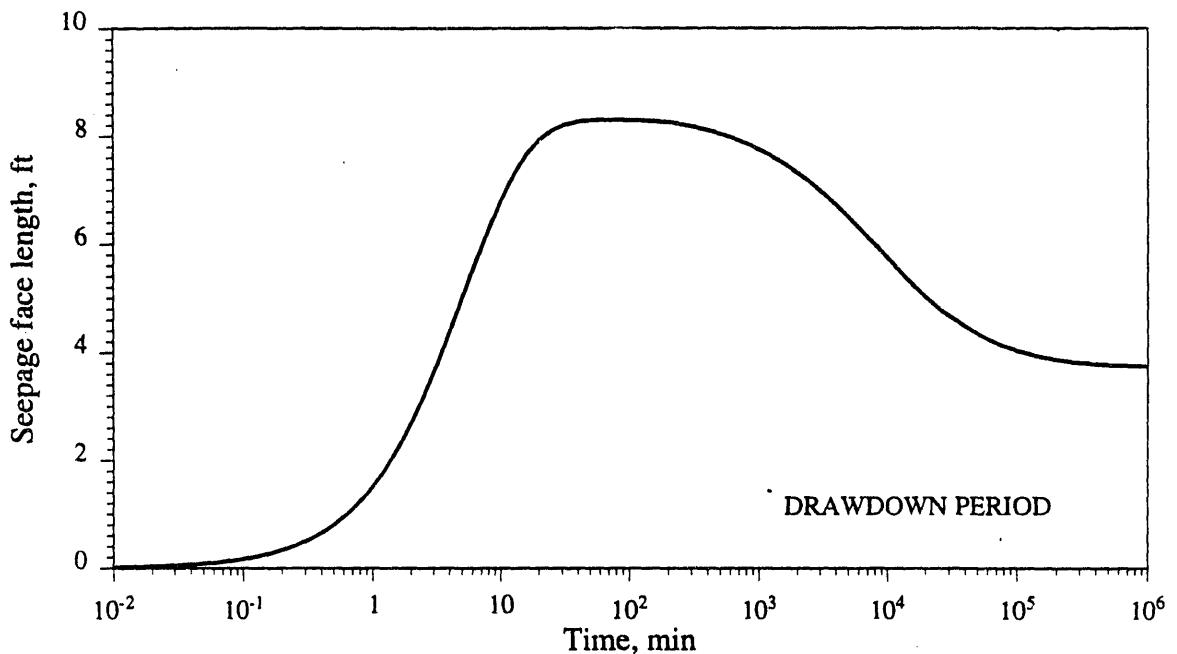


Figure 7.5: Seepage height vs. cumulative time for drawdown period

face starts developing after the pumping starts and reaches a maximum value of 8.3 ft at the end of wellbore storage effects. During the desaturating period, the seepage face length reduces while the free surface moves downward. The shrinkage face length seems to stabilize at about 3.7 ft as pseudoconfined behavior takes place.

An illustration of radial potential distribution, seepage face length and free surface radial profile is presented in Fig. 7.6. In this figure, the head potential in the bottom of the formation and the free surface position were normalized with respect to the original liquid height, H_o . Different flow regime periods are presented, starting at the end of the wellbore storage effect period. The wellbore liquid level is also shown, and the seepage length may be identified by the difference between the liquid level in the wellbore and the free surface level at a dimensionless radius of 1.

Figure 7.6 gives important information about transient gravity flow into a well. The formation of the desaturating cone at different times can be seen. In the profiles in Fig. 7.6 corresponding to 40 and 4,000 minutes of pumping, the bottom head potential distribution presented the same values in this example as well as the liquid level in the wellbore, and the only apparent difference was the free surface position. After the stabilization of the seepage face length, the profile at 400,000 minutes shows the free surface head becoming uniformly closer to the bottom potential head close to the wellbore. This is when the vertical flow,

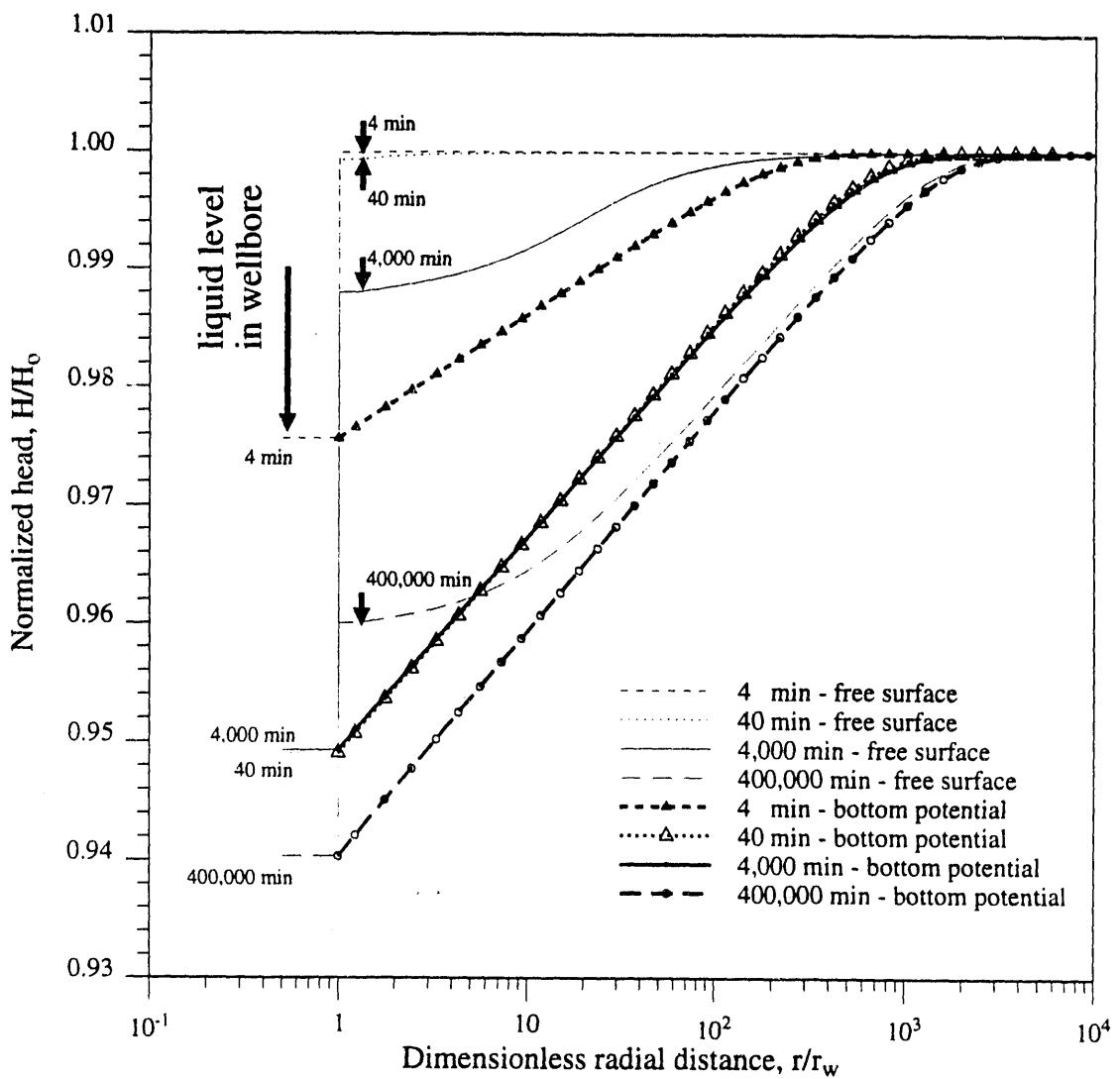


Figure 7.6: Dimensionless radial head profile during drawdown period. The sandface position corresponds to $r_D = 1$.

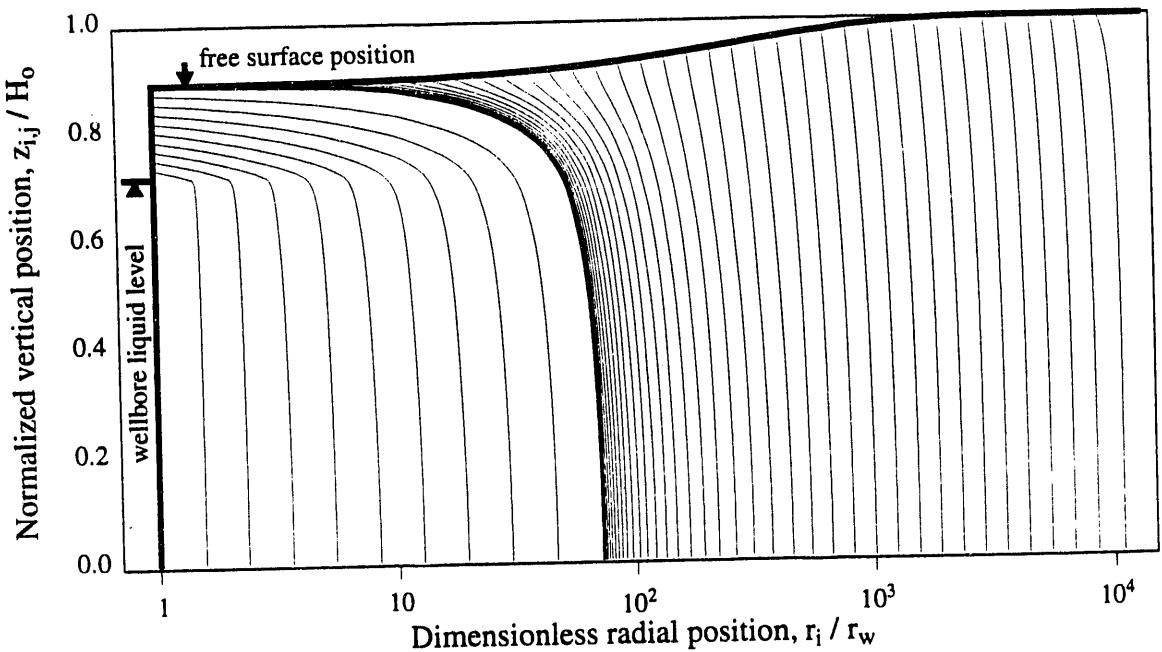


Figure 7.7: Iso-potential map and representation of a streamline at a production time of 10^6 minutes for a 30 mD reservoir.

given by the difference between the heads at the same radial location become stabilized. Based on the information obtained from Fig. 7.4 about the flow rate distribution along the sandface and in an isopotential mapping given in Fig. 7.7, we infer that there is a region immediately below the free surface where the potential gradients are steep near the wellbore. Figure 7.7 was generated for a 30 mD permeability to produce more pronounced gradients than in the case of a 120 mD reservoir.

7.1.2 Buildup

Contrary to the conclusion found in the groundwater literature that buildup behavior is similar to production behavior, results from the *SLM* model have shown that the pressure buildup process is different from drawdown in the case of a single well located in the center of an infinitely large reservoir. To analyze the buildup period, we will use the same set of data as in the drawdown run. The results are shown in Figs. 7.8 to 7.12.

A log-log graph of buildup pressure drop vs time in Fig. 7.8 may be compared with Fig. 7.1 for an analogy between the flow regimes with the flow period. In this figure, the pressure derivative $\frac{\partial p_w}{\partial \ln t}$ was graphed versus time, as well as the pressure difference between the buildup and the flowing pressure at shutin time. In both curves there is a unit slope

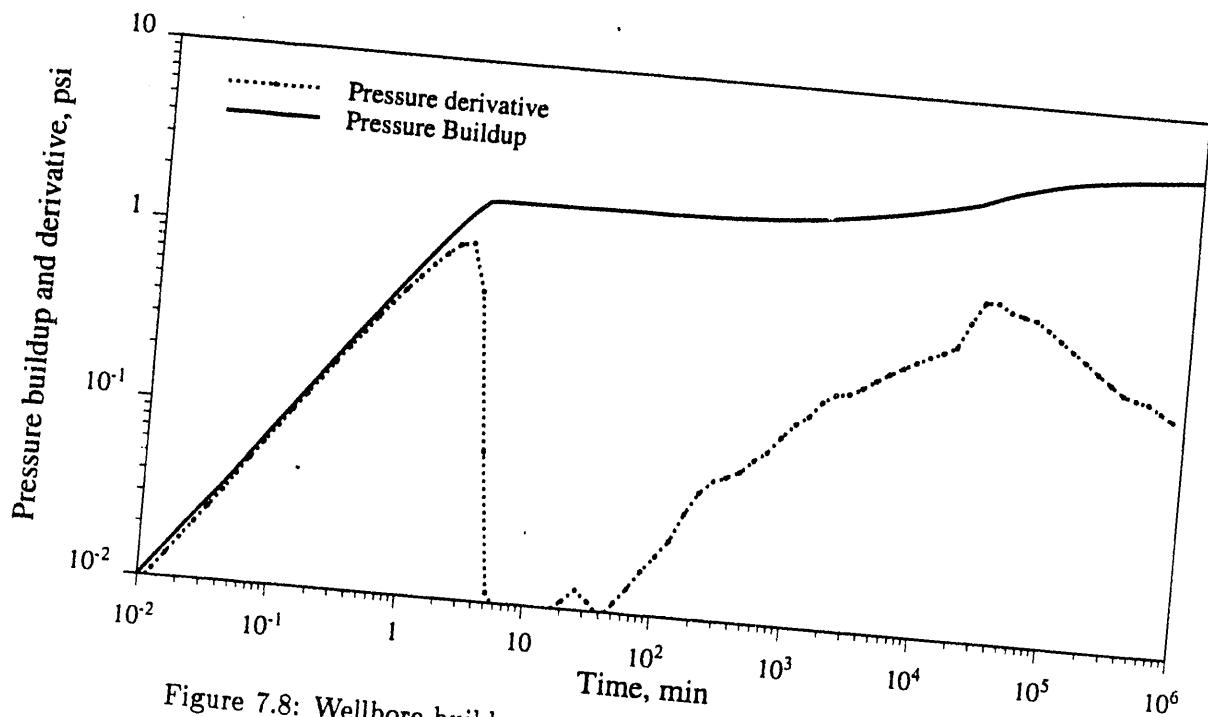


Figure 7.8: Wellbore buildup pressure and derivative - log-log plot

wellbore storage effect. At early times, the wellbore buildup pressure seems to be dominated by the wellbore storage. As the liquid level in the wellbore reaches the free surface level at the sandface, a different behavior in the log-log graph is evidenced by a discontinuity in the derivative curve. In the intermediate period in the pressure vs. time log-log graph in Fig. 7.8 and also in the semilog graph of Fig. 7.9, pressure buildup is very slow for about four cycles of time for this example. The pressure derivative curve increases during this period. The irregular shape observed in the pressure derivative curve in Fig. 7.8 is probably due to numerical effects in the calculations.

At late times, the buildup behavior becomes asymptotic to the confined flow solutions of both *Theis* (1935) and *Ramey et al.* (1989), but a slope similar to that of analytical approaches appears only at extremely long times. See Figs. 7.9 and 7.10.

A semilog *Horner* (1951) graph of the same data in Fig. 7.10 did not add further insight in the buildup pressure analysis. A conclusion, however, is that one could be misled by trying to extrapolate pressures or to obtain a permeability slope from the buildup data. The simulator results did not indicate pseudoradial behavior until very long times.

Another investigation of the pressure buildup behavior was performed by observing the seepage face length vs. time semilog graph in Fig. 7.11. Comparing Fig. 7.11 with

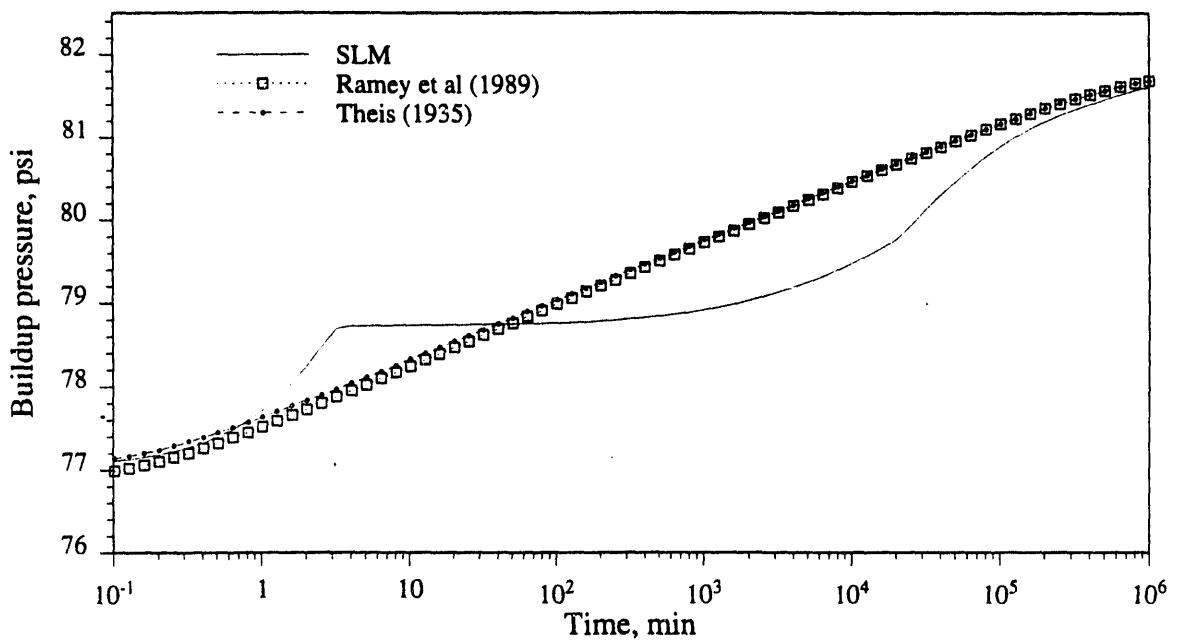


Figure 7.9: Wellbore buildup pressure - semilog plot

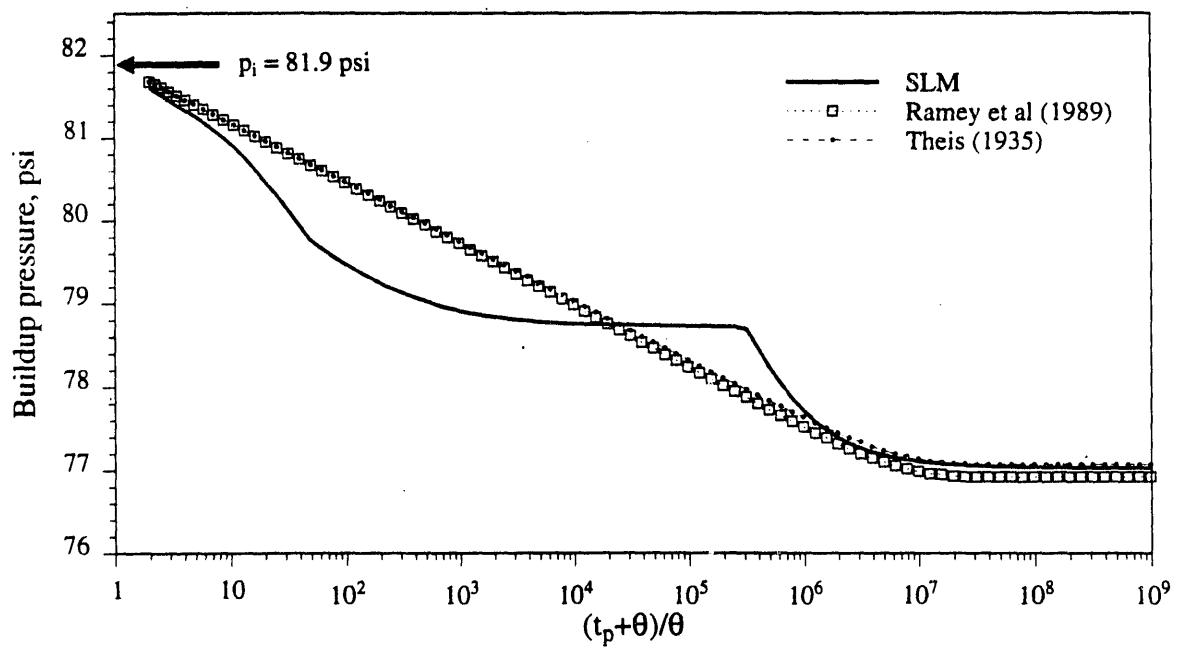


Figure 7.10: Wellbore buildup pressure - Horner plot

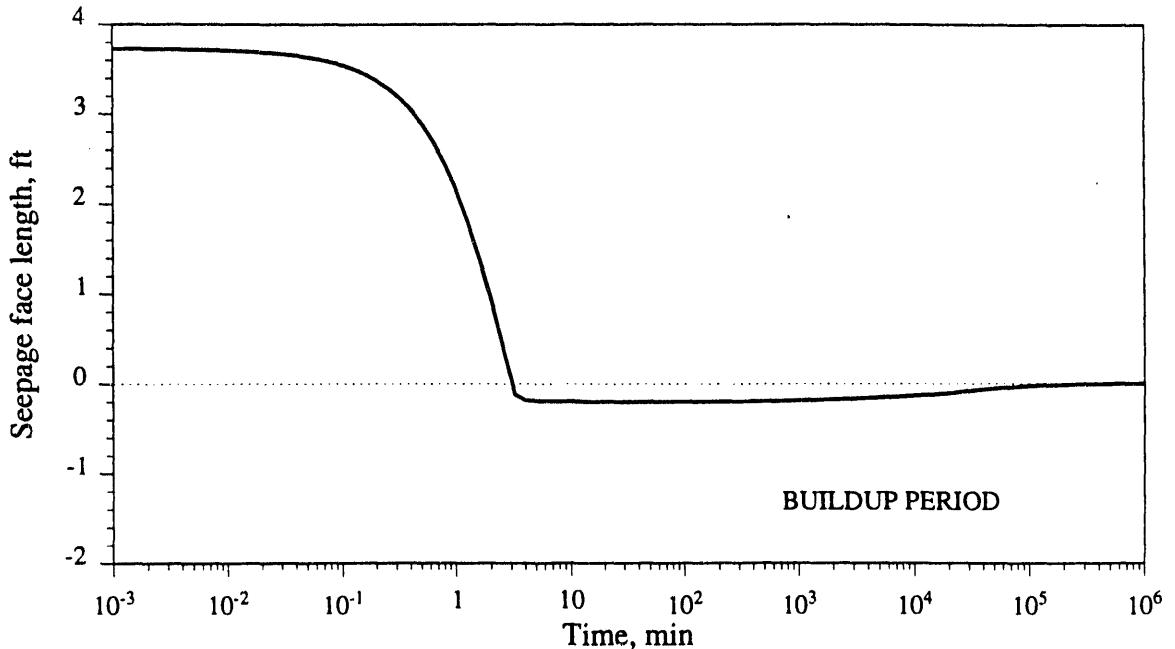


Figure 7.11: Seepage height vs. cumulative time for buildup period

the seepage face in Fig. 7.5, drawdown and buildup behaviors are completely different. As can be seen, the seepage face height drops to essentially zero within a few minutes after shut in. This appears to be an original and unique finding of this study. Actually the free surface height at the sandface appears to be negative 0.2 ft lower than the liquid level in the wellbore during most of the buildup. Similar behavior was found for other buildup cases.

A better understanding of the physical phenomenon of the pressure buildup mechanism in the gravity drainage well is obtained by inspection of the potential at the bottom of the formation and the free surface head distributions along the radial distance at several times in Fig. 7.12. The free surface profile indicates that the resaturation of the depressed cone is done uniformly. A flat liquid surface around the wellbore is grown progressively upwards, while the free surface at long distances remains almost stable. The radial extension of the flat free surface is increased with time as the liquid is replenished, and the liquid level rises. It appears that the depression cone fills like a tank of incompressible liquid. At the same time, the head potential in the wellbore is slightly higher than the free surface head at the sandface. This difference may be related to viscous resistance of flow in the vertical direction. The negative seepage face is gradually reduced as the vertical velocity of the free surface is reduced at long times. The results shown in Fig. 7.12 also brought some explanation for the excessive time step reductions resulting in very long CPU times to simulate the buildup period. A possible reason may be found in steep potential gradient development far from the wellbore, where the discretization is much less refined than close to the well.

As a conclusion, the approximate methods employed to analyze the drawdown period in a gravity drive well problem appear useless for buildup analysis. This observation contradicts *Neuman* (1975-a). According to *Neuman*, no hysteresis would be expected between drawdown and recovery processes, and the same analytical equations and interpretation methods should apply to both periods. *Neuman* based this assertion on the idea that no flow from the unsaturated region was considered in his method, and the unconfined aquifer was treated as a compressible system. However, as pointed out by *Ramey* (personal communication - 1989) and *Ramey et al.* (1989), the semilog plots in the *Neuman* paper show different times for the beginning of the apparent straight line behavior of drawdown and recovery periods from well test data. Also, *Ramey et al.* compared drawdown and buildup field data and concluded that there were "different physical mechanisms between drawdown and buildup". However, *Ramey et al.* (1989) did believe they identified a semilog straight line immediately after wellbore storage during buildup. A semilog straight line was not clear in *SLM* buildup runs. It appears that no thorough inspection of free surface recovery has been presented prior to this study.

7.2 Permeability

The effects of isotropic permeabilities in the pressure response from the *SLM* model studies are presented in two sets of drawdown curves for different formation permeabilities. Figs. 7.13 and 7.14. The permeability, as well as the flow rate, is a major factor driving the potential distribution in the formation because of the streamline shapes, and the consequent dynamic vertical pressure gradient development.

Two sets of permeability range were considered concerning the input data in Table 7.1: a low permeability range varying from 30 mD to 120 mD, and a high permeability range varying from 120 mD to 480 mD. Those ranges were defined only with the objective of graph scaling. In the low permeability range, Fig. 7.13 shows the same classical gravity drive curves, with the same late time trend, following the *Ramey et al.* (1989) slope. There is evidence from these results that the late time p-squared solution is somewhat closer to the simulated results when the distance between the p-squared and the line source solutions is enlarged. At early times, an apparent trend of the numerical solution to fit the analytical Theis (1935) solution can be explained by the discretization errors involved. All cases presented in both Figs. 7.13 and 7.14 were run with a time interpolation parameter $\theta = 0.9$, which generally produced small drawdowns after a long production time, as observed

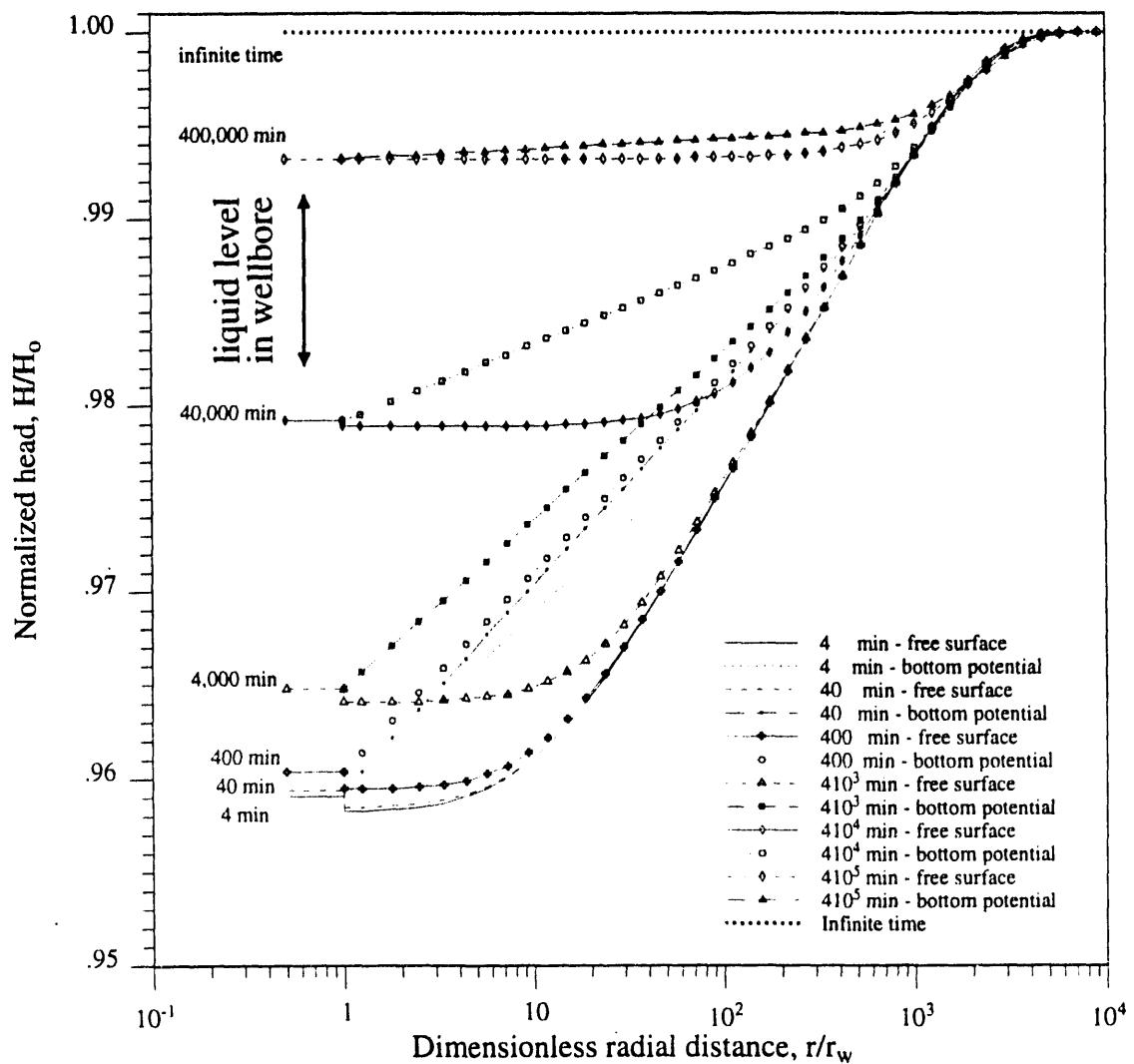


Figure 7.12: Dimensionless radial head profiles during buildup period. The sandface position corresponds to $r_D = 1$.

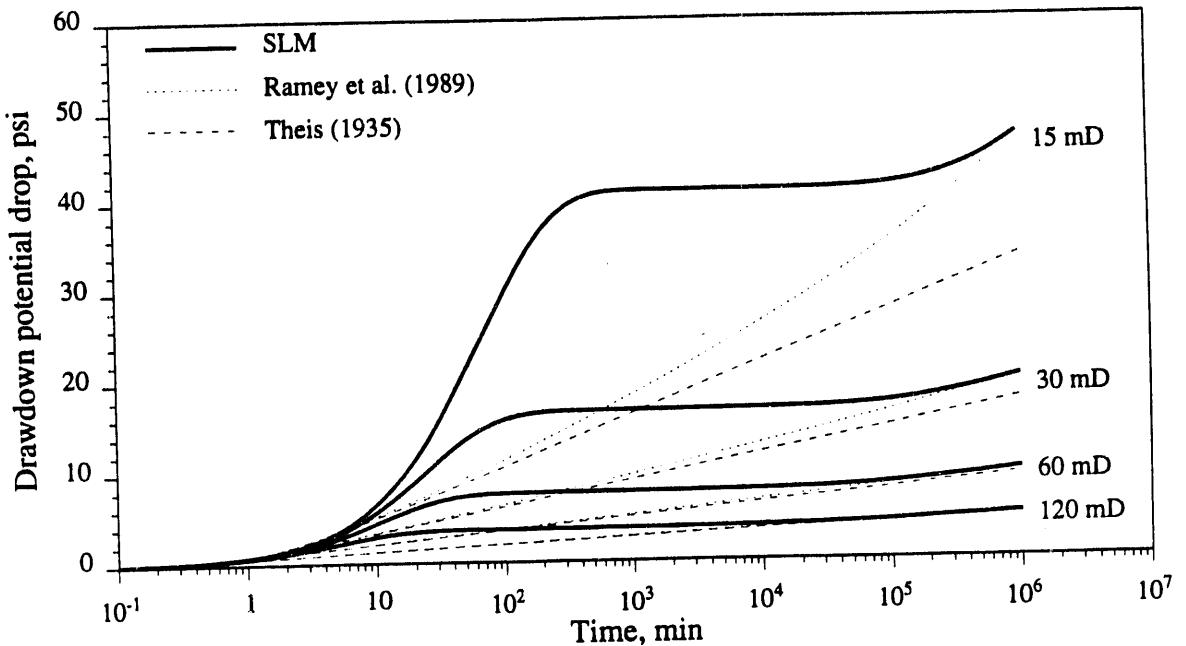


Figure 7.13: Wellbore pressure drawdown of intermediate to low permeability reservoirs under gravity drainage

in the Chapter 6.

In addition, the runs of the simulator for the case of a finite skin effect described in Section 7.4 shows that the long time drawdown potential solutions are the same for the *SLM* and the *Ramey et al.* solutions.

7.3 The Seepage Surface

In the process of gravity drainage through a porous medium, it is generally recognized that a thin curtain of liquid falls along the inner production face, whatever the geometry of the flow. This surface, called the seepage face, reflects the difference between the maximum potential inside and outside the porous medium, and, in the case of water or oil wells, the seepage face is not easily measurable in field experiments. The analytical methods used so far neglect the existence of the seepage face phenomenon. This may be one of the reasons that analytical models do not reproduce the well pressure (or head) behavior accurately.

The *SLM* simulator used the same classical inner boundary condition as in most formulations in the groundwater literature. The *SLM* results consistently produced the

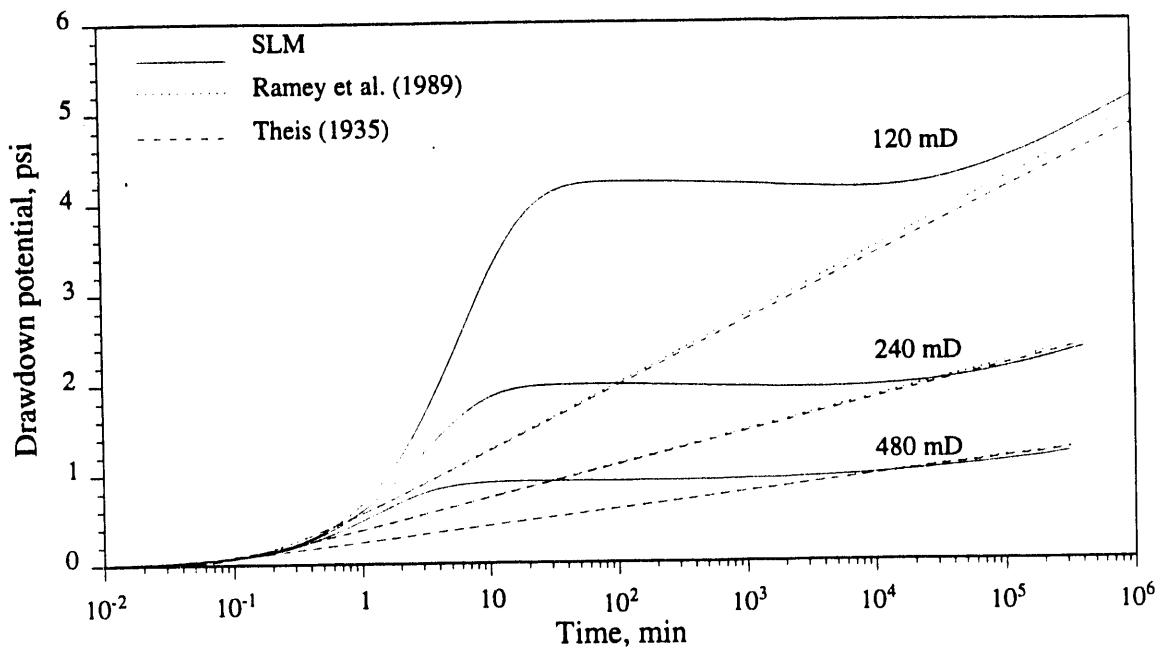


Figure 7.14: Wellbore pressure drawdown of intermediate to high permeability reservoirs under gravity drainage

seepage phenomenon with a clear visualization of the physics of the problem. No surface tension effect, though important in small scale sandbox experiments, was necessary to cause the existence of the seepage face, which can be explained entirely by viscous flow theory, and verified by the numerical model.

The seepage height should be related to the formation permeability at the wellbore, the pumping rate and the original liquid height in the formation, because the seepage height represents a difference between the maximum and the minimum potential at the inner boundary. A skin effect should also be a contributing factor to the seepage face development. Figures 7.15 and 7.16 illustrate how the computed seepage face height changes with permeability, and Figs. 7.5 and 7.11 illustrate the seepage face behavior during drawdown and buildup tests, already discussed in Section 7.1. The observed trend in Figs. 7.15 and 7.16 for different sets of permeability is to reduce the seepage face length as the permeability increases.

The existence of the seepage face is a consequence of the free surface position, which is a function of the liquid velocity. When two streamlines converge while moving toward the sink (well), the resulting streamline velocity increases to preserve a material balance. Since there is enough voided pore space above the free surface, a fraction of that dynamic

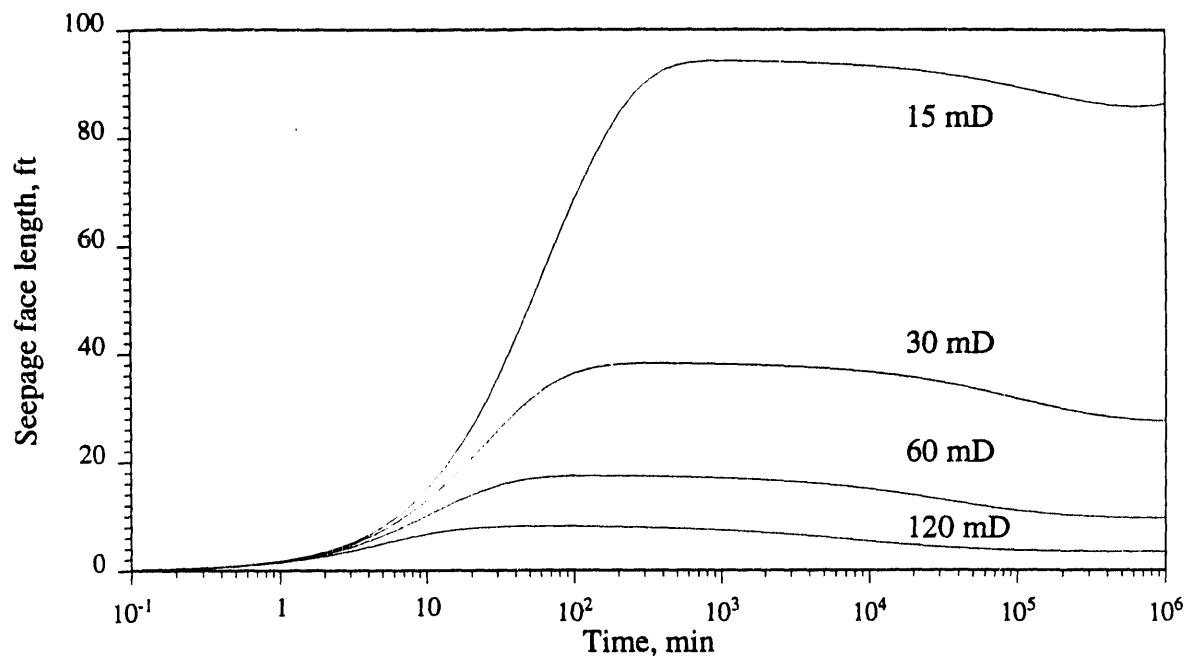


Figure 7.15: Semilog plot of the seepage height: low permeability range

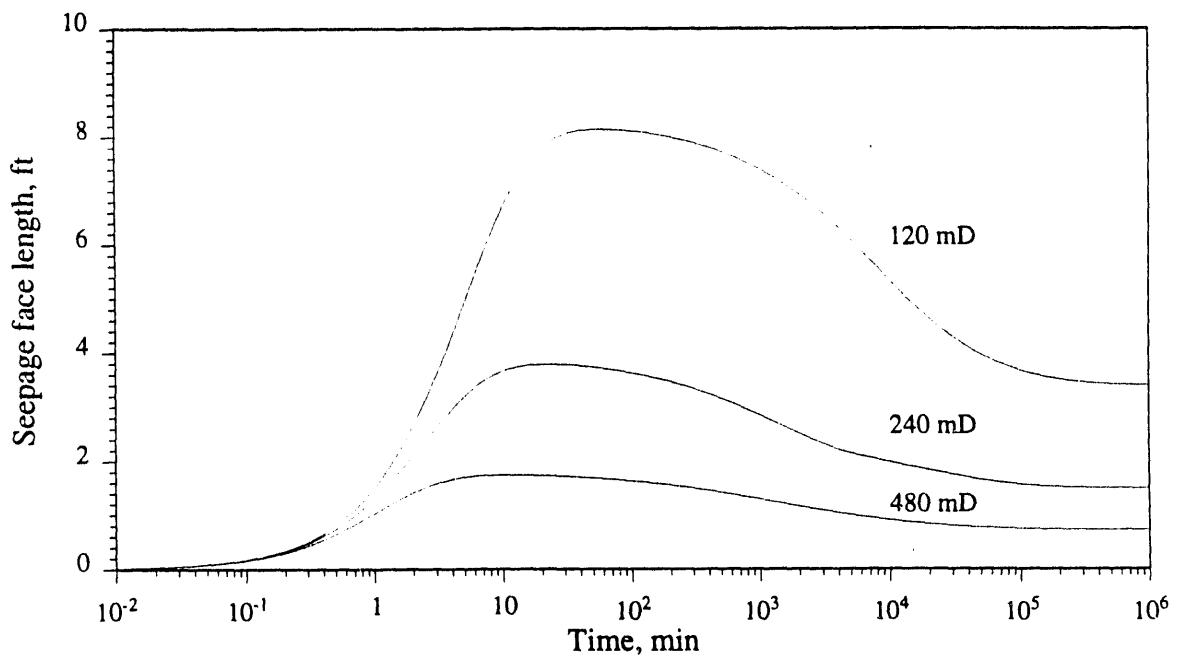


Figure 7.16: Semilog plot of the seepage height: high permeability range

energy may be used by increasing the liquid column. Figure 7.6 shows an almost flat free surface profile in a log radial scale in the vicinity of the well after a long drawdown period.

This natural phenomenon may be described by the following implicit equation, Eq. D.1 in App. D, representing the free surface height at a location r from the well center, according to *Wyckoff et al.* (1932):

$$h(r) = \frac{\Phi_b(r)}{\rho \frac{g}{g_c}} + \frac{\bar{v}_v(r) \mu}{k_v \rho \frac{g}{g_c}} h(r), \quad (\text{Eq. D.1})$$

The relationship between the free surface position and the head potential is verified in App. D.

There is little general information on the seepage face height for a free surface gravity drainage well. One conclusion in the groundwater literature is that the seepage face height is about half the drawdown for cases where the drawdown is less than 10 percent of the initial head. Some of the figures presented in this chapter show that the seepage face height increased during the drawdown period, then declined and became reasonably constant near the end of the drawdown period. Comparison of the drawdown head and the seepage face height shows that the seepage face height is rarely half the drawdown. No obvious relationship for seepage face height was found in this study.

7.4 Skin Effect

To complement the preceding analysis of the influence of permeability on the well pressure response, the effects of a damaged annular region are now considered. Figure 7.17 contains a set of time-drawdown curves for skin parameters of 0, 4 and 8. The same basic reservoir properties in Table 7.1 were used in this analysis. The analytical solutions by *Theis* (1935) and *Ramey et al.* (1989) were evaluated by numerical solutions using *FLIGRAM* and *PDE* programs which include wellbore storage effects.

A thin region of reduced permeability around the wellbore keeps the liquid level high in the sandface vicinity, and the free surface location and the potential distribution along the reservoir become different. A change in the seepage face is another consequence, because the seepage face length increases as skin increases. Figure 7.18 presents a radial profile of the potential head around a damaged well, compared with an undamaged well. An increase in skin effect does not appear to affect the bottom potential. The free surface is higher around the wellbore in a damaged formation than in an undamaged formation. However,

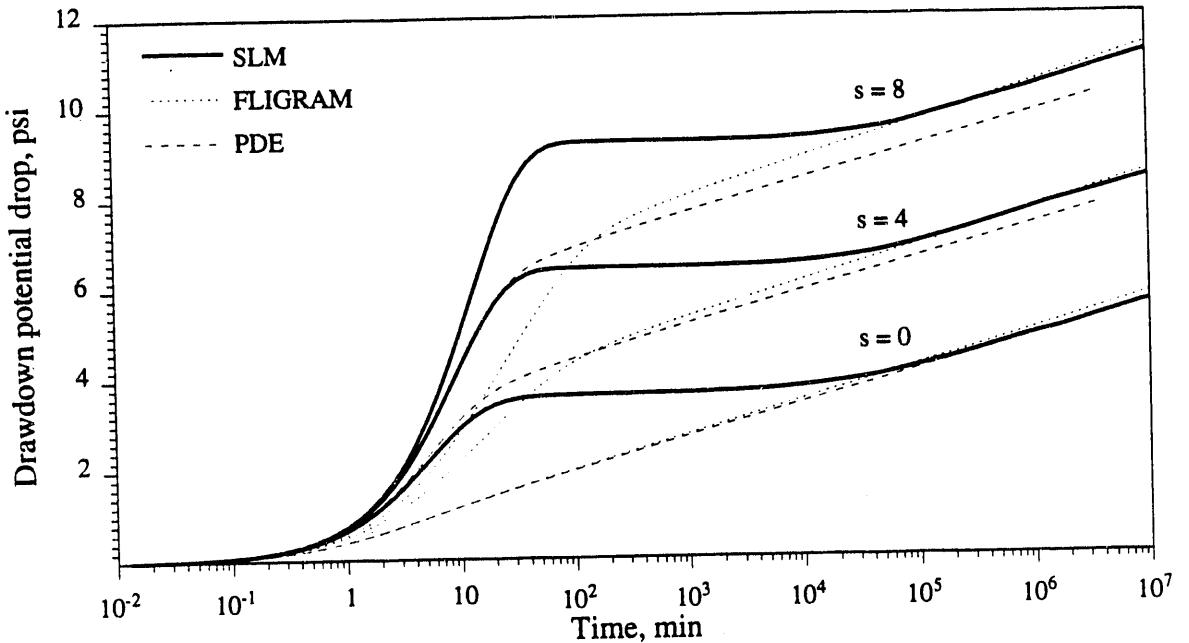


Figure 7.17: Pressure drawdown vs. time for different skin parameters.

at long distances from the wellbore, this situation is reversed, because the material balance must be preserved. In the graph of Fig. 7.18, it is not possible to see a small difference between the free surface positions due to the scale. From these results, we infer that the presence of the damaged region reduces the slope of the stream-lines behind the damaged annulus, the vertical head gradients are reduced, and the flow becomes proportional to the liquid height, as idealized by *Dupuit* (1863). The time-drawdown curve of Fig. 7.17 also supports this explanation.

7.5 Anisotropy

The free surface gravity drainage problem depends greatly on vertical permeability. A set of different permeability factors f_k ($f_k = k_v/k_h$) ranging from 1 to 120 was chosen to analyze the effects of anisotropy on producing well pressure. Results of *SLM* runs were graphed together in Fig. 7.19 to make qualitative comparisons with isotropic reservoirs. From the results, we observe that a high degree of formation anisotropy causes a strong influence on the wellbore potential, since vertical flow gradients near the well are steeper than in other regions.

In Fig. 7.19, the influence of vertical permeability is shown for several examples. In

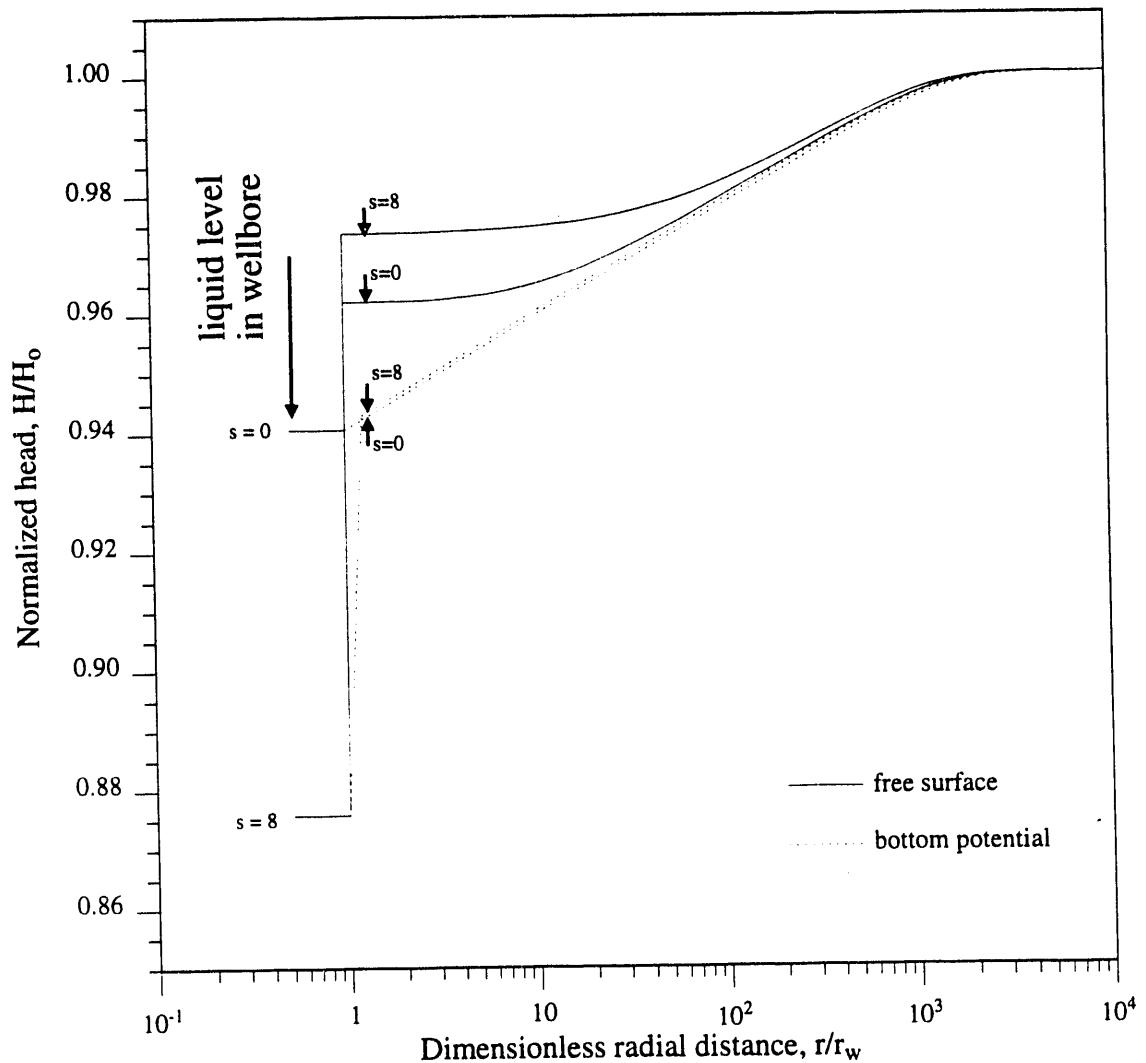


Figure 7.18: Radial head potential profile of a damaged well after a flowing time of 400,000 minutes. The sandface position corresponds to $r_D = 1$.

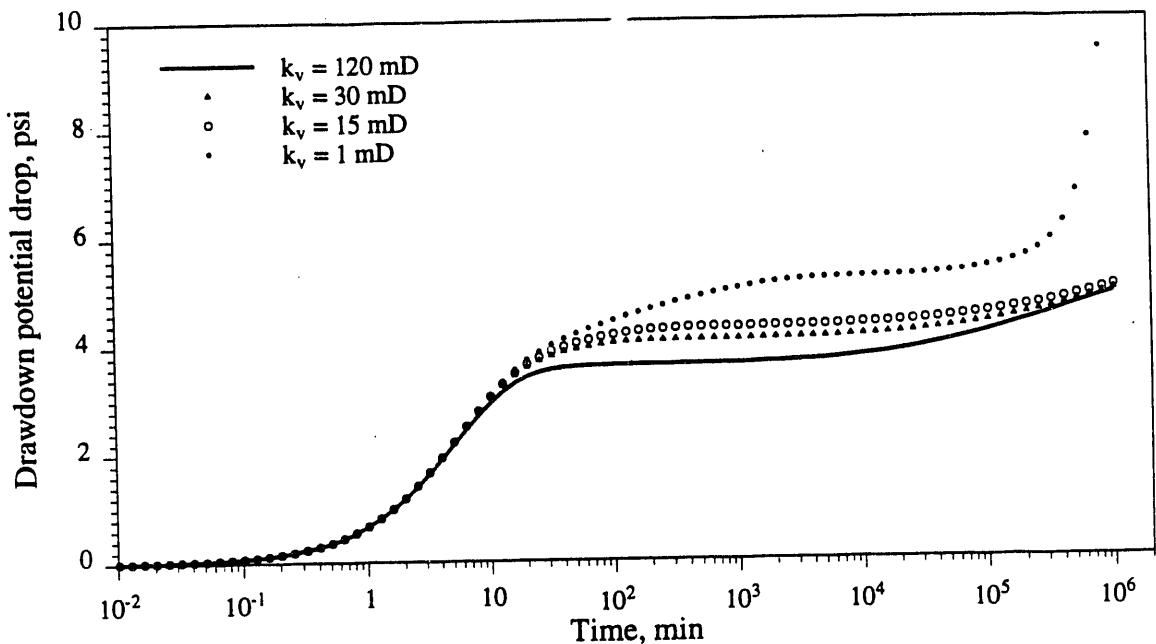


Figure 7.19: Results from *SLM* for a 120 mD horizontal permeability and different vertical permeabilities

a case where k_h of 120 mD is a fairly high permeability for a flow rate of 100 BPD, the streamlines do not bend sharply, even near the well, as inferred from the profile in Fig. 7.18. However, when the vertical permeability decreases to 1 mD, the wellbore liquid level drops rapidly to zero at late times.

The effects of anisotropy are most important during the desaturating period. On the other hand, the trend of late time potentials is to approximate the isotropic solution for the horizontal permeability. This observation may be an important key for future study of a method to obtain the vertical permeability.

7.6 Production Flow Rate

Figure 7.20 presents a pressure drawdown graph for four different production flow rates ranging from 50 BPD to 400 BPD. The nonlinearity of the potential near the wellbore and the flow rate variation is evident. A flow rate variation from 50 to 400 BPD (8-fold increase) produced an increase from 1.75 to 16.7 (9.5-fold increase) in the drawdown

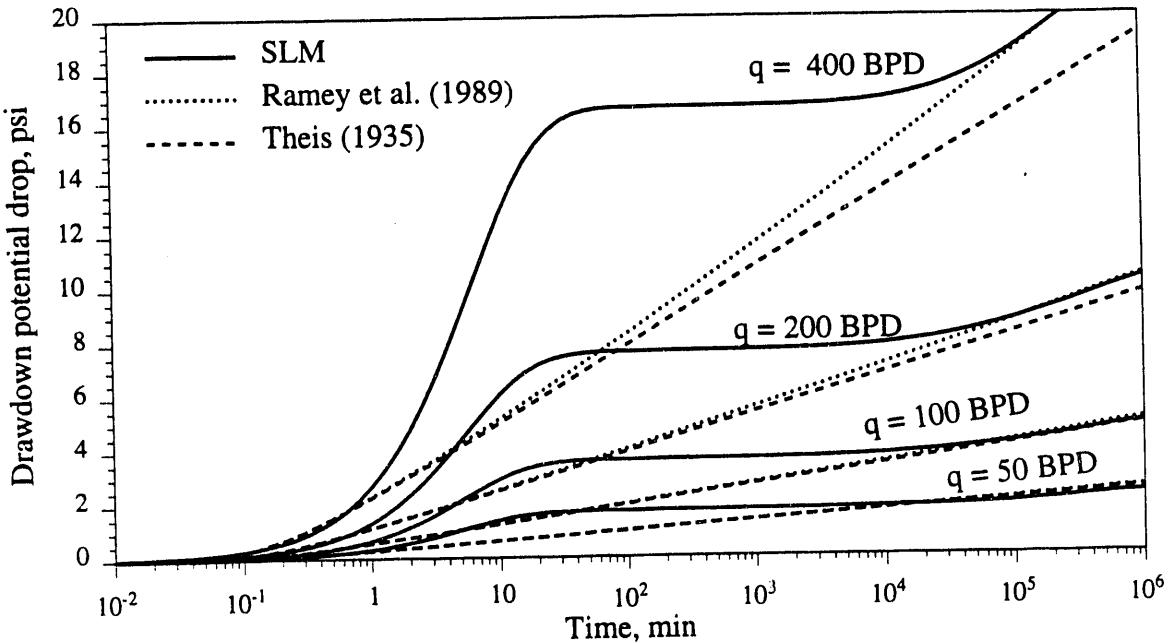


Figure 7.20: *SLM* results for different production flow rates

during the flat desaturating potential period. A possible reason for the disproportional drawdown is that the high flow rate causes the uppermost streamlines to reach the inner boundary late, causing an additional potential drop. Figure 7.21 is a graph of the *SLM* results for a 800 BPD flow rate case compared with the *Ramey et al.* (1989) approach. The late time pseudoconfined behavior shows the same slope for both solutions. The absolute values of the drawdown at long times are similar in the p-squared and the *SLM* numerical approaches.

An interesting result is presented in Fig. 7.22 where two different runs for a constant q/k ratio of 0.6 are compared. The shift in time in Fig. 7.22 is a consequence of real time graph. From the *Ramey et al.* (1989) solution, the following equation (Eq. 11 in *Ramey et al.* paper) applies at late times:

$$p_i^2 - p_w^2 = m \left\{ \log_{10} t' + \log_{10} \left[\frac{k}{\phi \mu (p_i^{-1}) r_w^2} \right] - 3.23 + 0.87S \right\} \quad (7.6.1)$$

In Eq. 7.6.1, the argument in the logarithmic term is a function of permeability k and does not depend on the the production flow rate. As a consequence, a change in permeability should produce a shift in the time proportional to the permeability-m. In the

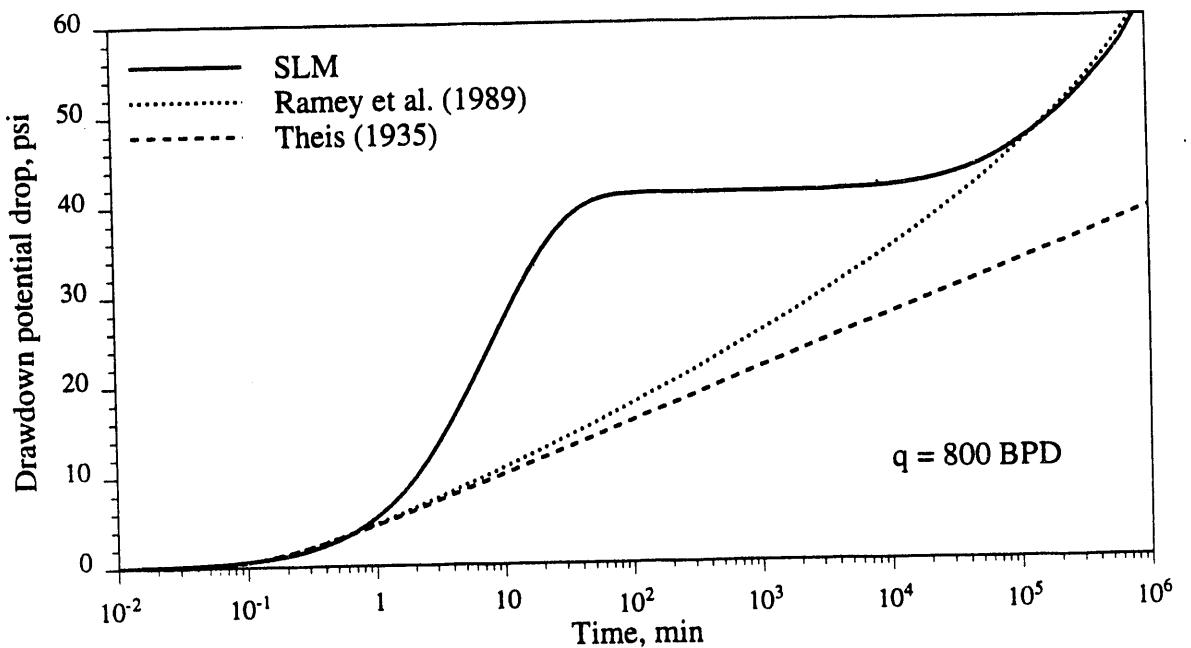


Figure 7.21: *SLM* result compared with *Ramey et al.* (1989) and *Theis* (1935) solutions for a high flow rate case

runs shown in Fig. 7.22, m is constant for both cases, according to the following expression also from *Ramey et al.* paper:

$$m = 325.2 \frac{q B \mu}{k(h/p)} = 234.9 \text{ psi}^2/\text{cycle}.$$

Both permeability and production flow rate produce the same nonlinear effects in the problem. We learn from this result that for the same q/k ratio, the potential drawdown will present similar behavior, displaced in time by a constant logarithmic factor.

7.7 Original Liquid Height and Transmissivity

The influence of the original height of the liquid column on the wellbore potential drawdown is verified in the following. Five different original heights were selected to run the examples in Fig. 7.23. Using the same set of basic data in Table 7.1, values of the liquid original height were measured from 63 ft to 252 ft. The graph in Fig. 7.23 shows that the potential drawdown was not directly proportional to H_o . The run for $H_o = 63$ ft dropped the liquid level in the well abruptly during the wellbore storage period, reaching

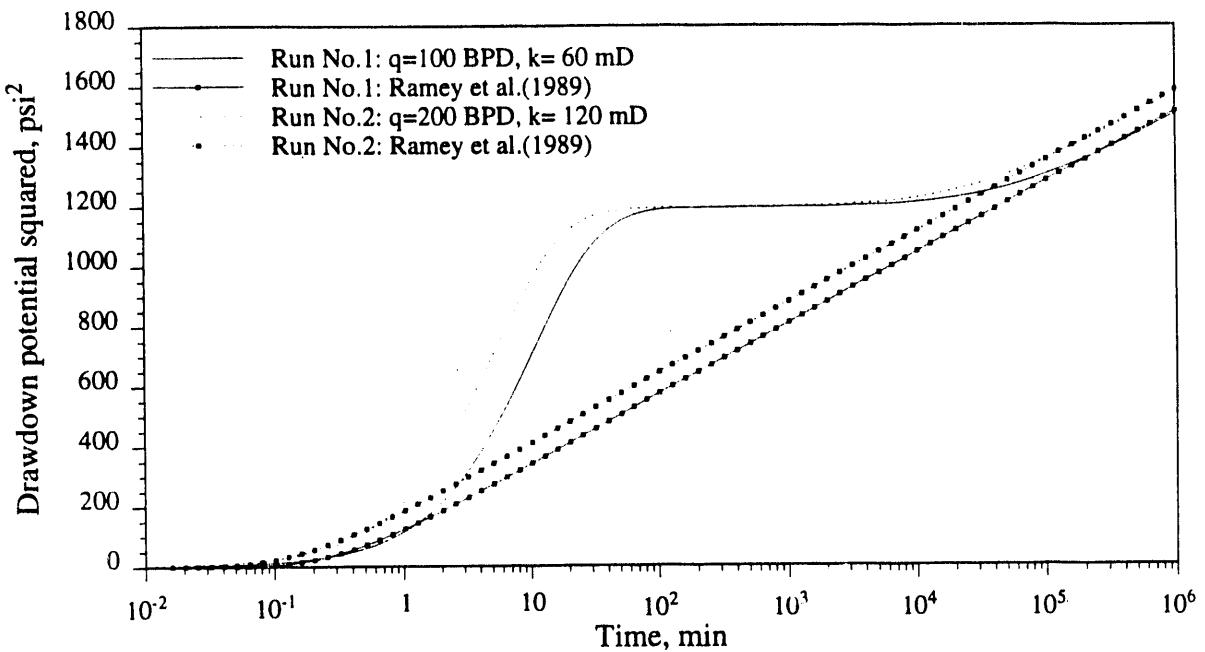


Figure 7.22: Comparison between *SLM* runs of two tests in formations with the same q/k ratio

the bottom of the interval. However, increasing H_o by a factor of 2 the potential drop during the desaturating period was only approximately 6 psi.

Another verification was made by comparing two runs with the same transmissivity $k H_o / \mu$, but different permeabilities and original liquid heights. A 120 mD reservoir with an original liquid height of 126 ft and an 80 mD reservoir with an original liquid height of 189 ft were selected for this analysis. The transmissivity in both cases was 15,120 mD-ft/cp. The results presented in Fig. 7.24 indicated that there was not a superposition of the curves during the desaturating period. However, at late times, the drawdown potential drops produced very close results with consistent slopes m , as compared with the *Ramey et al.* (1989) solution. The desaturating period presented a smaller potential in the wellbore (higher potential drop) in the case of smaller permeability.

An expected shift between the late time solutions was observed due to different initial pressures in the p -squared plot of the same data, as seen in Fig. 7.25. In this example, the effect of retarded development of velocity gradients in the formation at the beginning of pumping is more pronounced in the lower permeability case. During the desaturating period characterized by the flat shape in the time-drawdown graph, the velocity gradients become larger and the drawdown potential in the well gradually reduces before the pseudoconfined behavior.

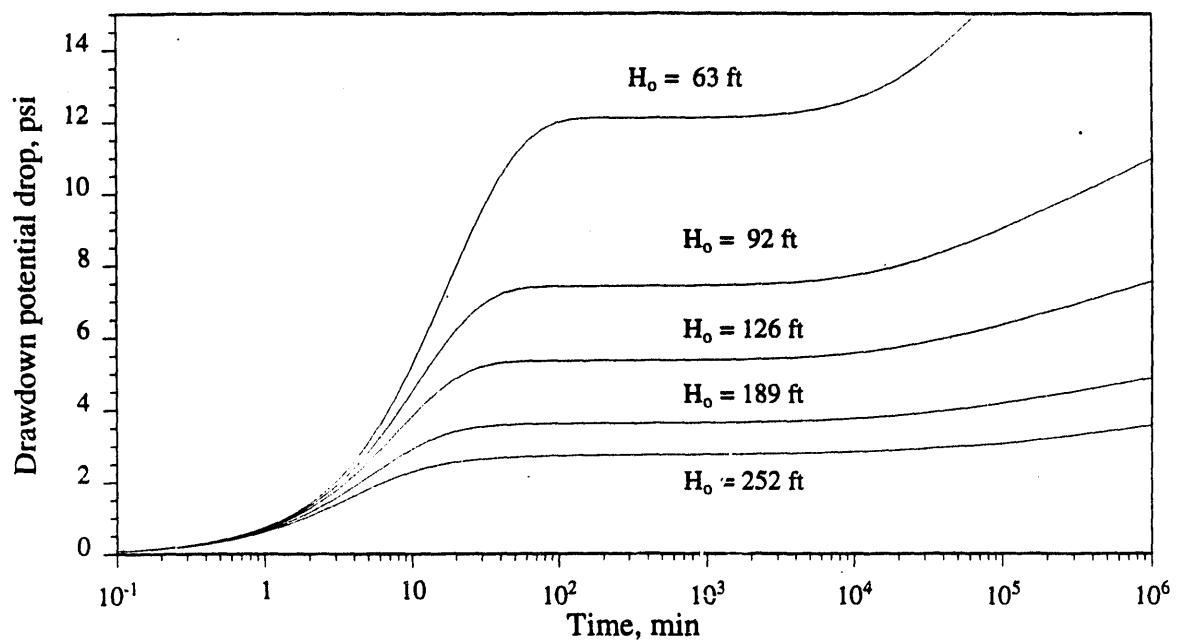


Figure 7.23: *SLM* results for different original static liquid level

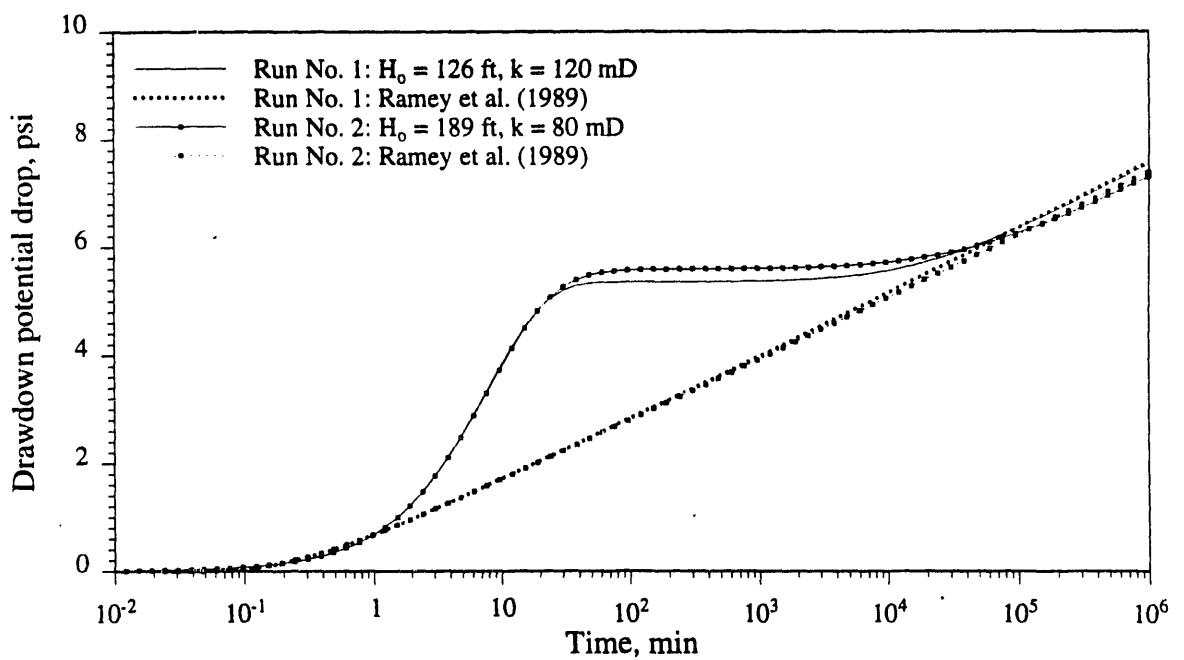


Figure 7.24: Potential drawdown results for the same transmissivity and different permeabilities

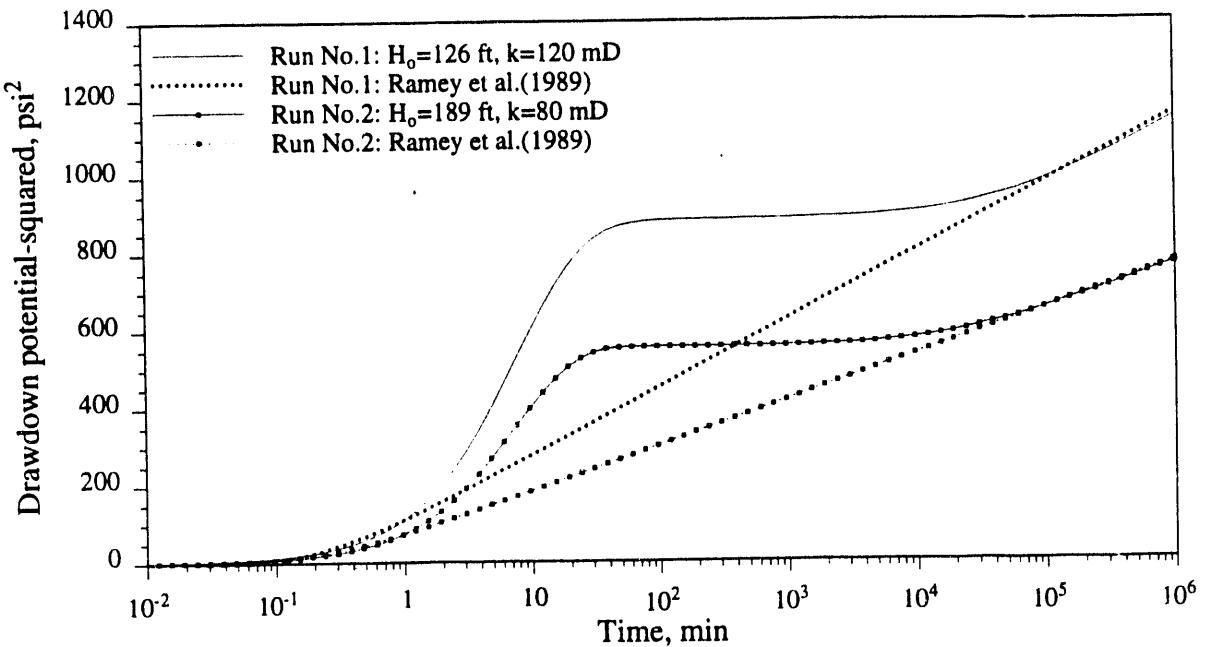


Figure 7.25: Potential squared drawdown results for the same transmissivity and different permeabilities

7.8 Considerations About Well Test Projects

From the preceding analysis, a consistent understanding of the physics of the problem allows us to reach conclusions about the use of the *Stream Layer Model* in planning a gravity drainage well test. We have seen the effects of the main variables that determine the wellbore potential drop during drawdown and buildup periods. We have also seen that for different combinations of those variables, the producing time required to start pseudoconfined behavior was longer than one or two months of producing time. In addition, a full log-cycle in the time drawdown graph necessary to define the m-permeability from the slope of a straight line in a p-squared plot would require one or two years of production at a constant pumping rate in the given examples. The preceding investigation did not include a full range of possible variations of the parameters, but parameter ranges were within many practical field value ranges.

To briefly extend the investigations, the *SLM* was run to simulate a short well test of 24 hours of pumping followed by a buildup period. The buildup period was set long enough for the liquid level in the wellbore to equalize with the liquid level in the sandface, shrinking the seepage face to a minimum, and starting the resaturating period. Considering that in a gravity drainage well the major unknowns are the permeability, the permeability ratio and

the skin parameters, sets of different values of permeability and skin were tested. We did not consider the effect of anisotropy in this study. In addition, other reservoir properties such as porosity, residual saturation in the desaturating region, wellbore radius, original liquid height at static condition, liquid viscosity and liquid density were known. The basic data in Table 7.1 were used.

Figures 7.26 to 7.28 show results of several sets of tests with skin effect parameters of 0, 4 and 8. Permeability ranged from 30 mD to 150 mD. A consolidation of the results is presented in Fig. 7.29 where only a few limiting permeability values were included. The drawdown potential drop and the buildup potential drop vs time plots were placed side-by-side to characterize the differences between drawdown and buildup behaviors.

Important conclusions may be reached from the results shown in Figs. 7.26 to 7.29. Some discoveries were described in previous sections of this chapter such as the degree of nonlinearity of the formation permeability and the differences between the drawdown and buildup process. What is remarkable in the results presented in this section is the information about the free surface position at the sandface during the buildup period. The inflection point in the buildup curve where the liquid level in the well reaches the same position as the free surface at the sandface varies only a few feet for formation permeabilities ranging from 30 mD to 150 mD. For usual field test accuracy, this information is not useful. The time in which the liquid levels equalize, however, increases as the permeability decreases with a large change as observed in Fig. 7.30.

In Fig. 7.31 the potential drops during the drawdown period are plotted for two different pumping times. The times when the resaturating period starts in both cases are very close, different only by 4 minutes for production times of 1 day and 10 days.

Figure 7.32 gives important information about the free surface at the sandface while the liquid level travels in the wellbore during the 24-hour drawdown and buildup periods. This example was run for a 30 mD formation permeability and no skin effect. The free surface during the early buildup period remains immobile until the liquid level in the wellbore reaches the same position. Then, the free surface at the sandface starts gently moving upwards during the resaturating period.

Based on the results of the series of runs described in this subsection, the graph of Fig. 7.33 is presented. This is an example of a correlation obtained from the *SLM* for a typical set of data as in Table 7.1, and for the short well tests.

In the next chapter, a summary of the most important conclusions from the results of the *SLM* are presented, as well as some recommendations for future work.

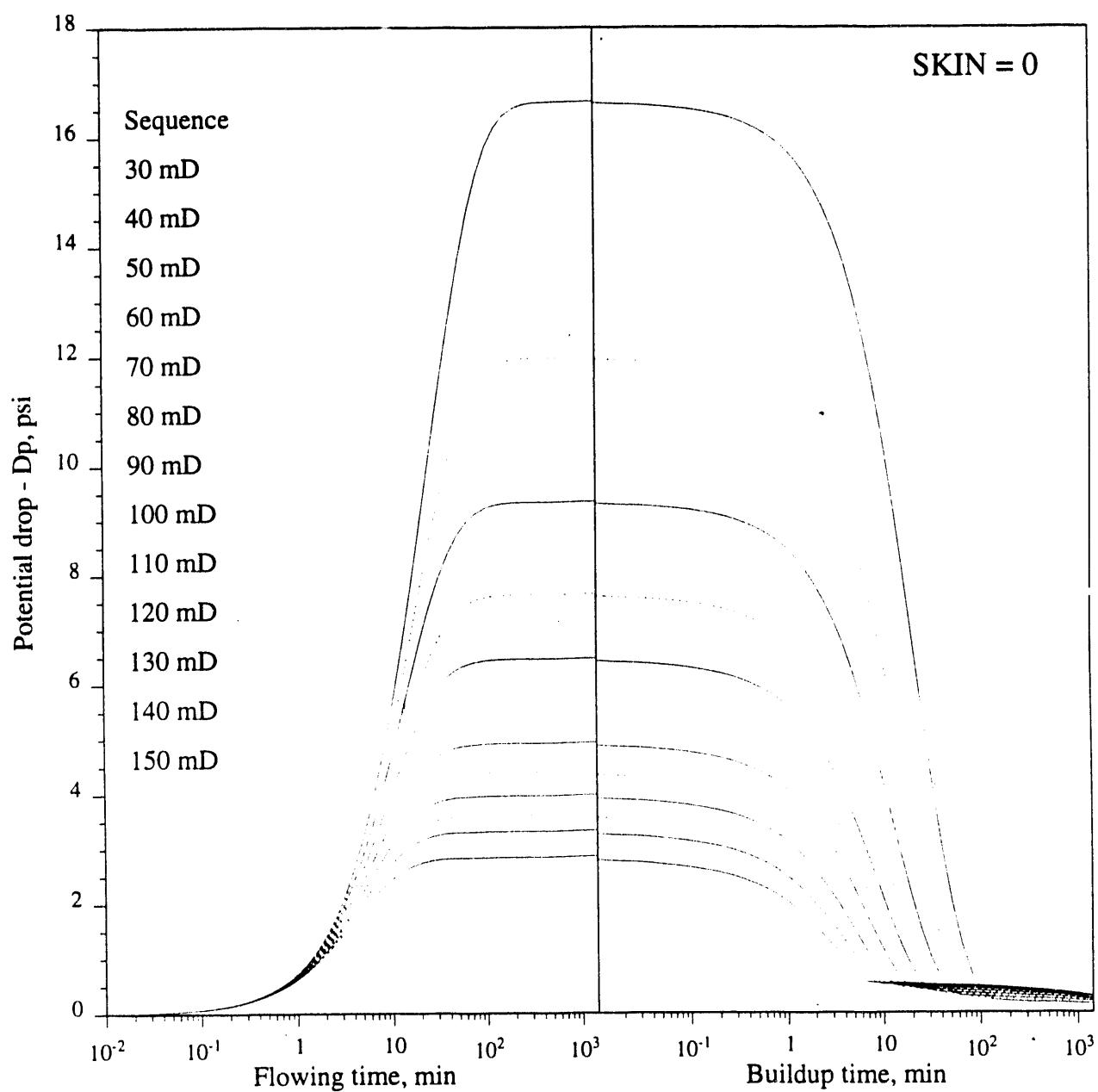


Figure 7.26: Combined 24-hour flowing and buildup test for a set of different permeabilities.
Skin = 0.

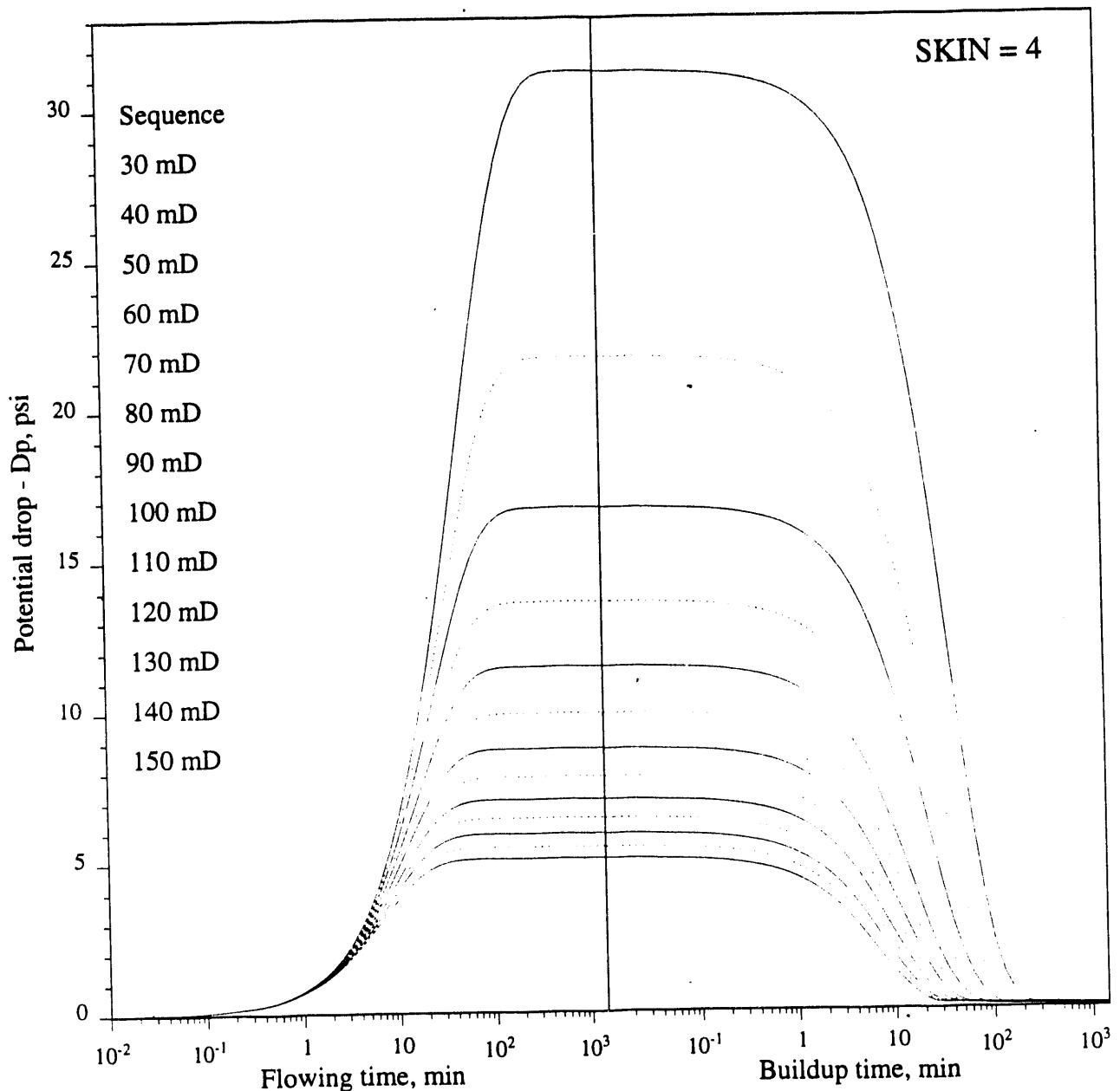


Figure 7.27: Combined 24-hour flowing and buildup test for a set of different permeabilities. Skin = 4.

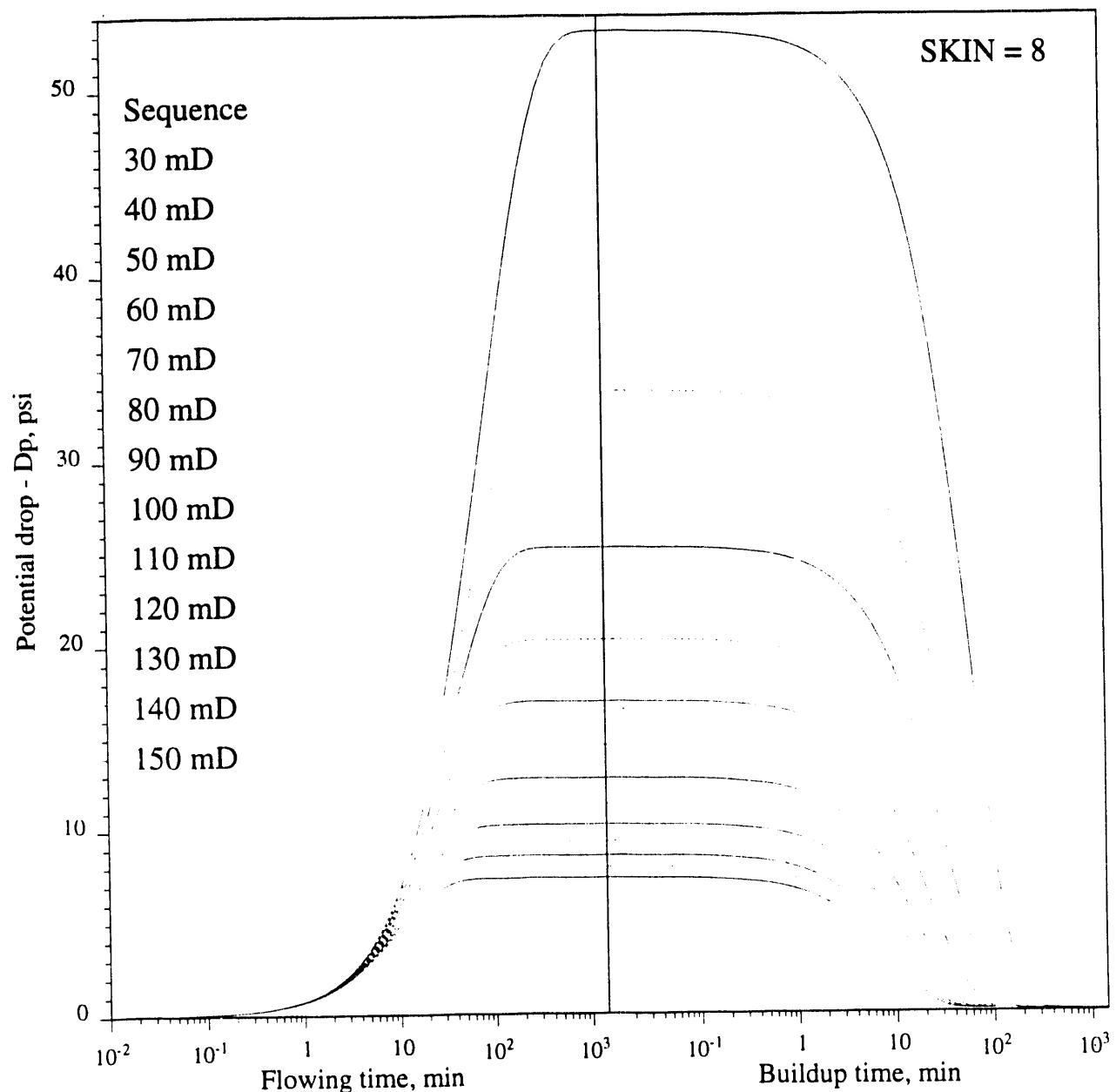


Figure 7.28: Combined 24-hour flowing and buildup test for a set of different permeabilities.
Skin = 8.

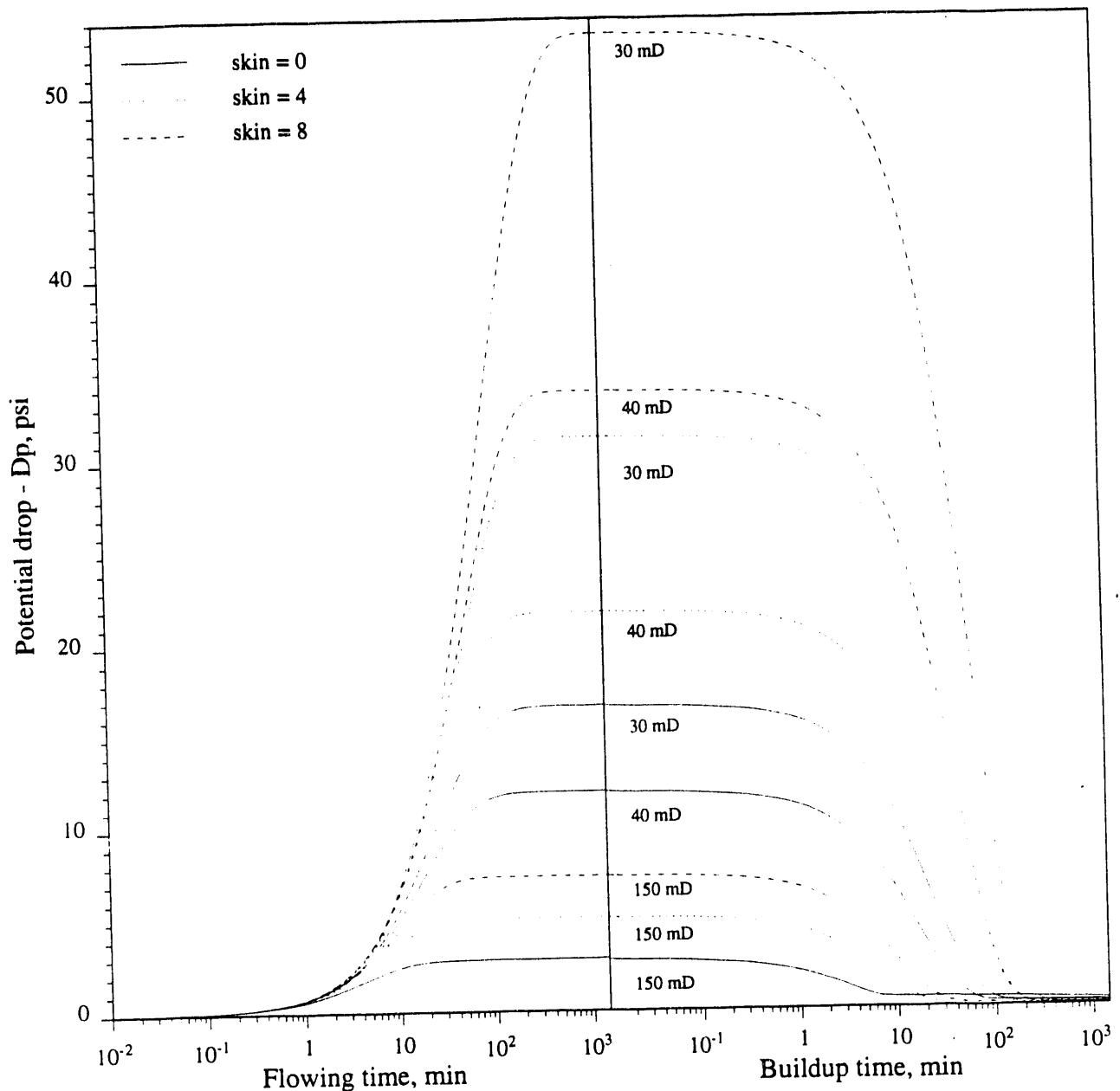


Figure 7.29: Combined 24-hour flowing and buildup test for a set of permeabilities and skin parameters.

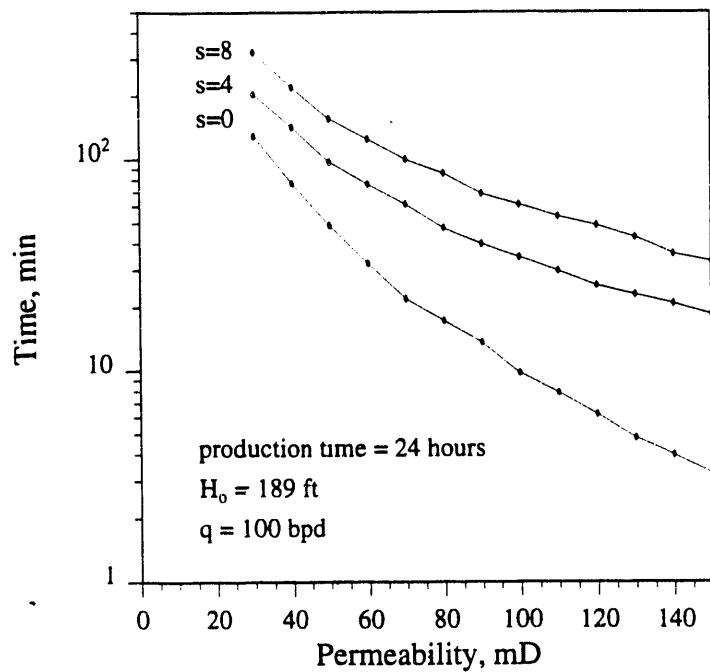


Figure 7.30: Time in which the liquid level in the well reaches the free surface level in the sandface during buildup.

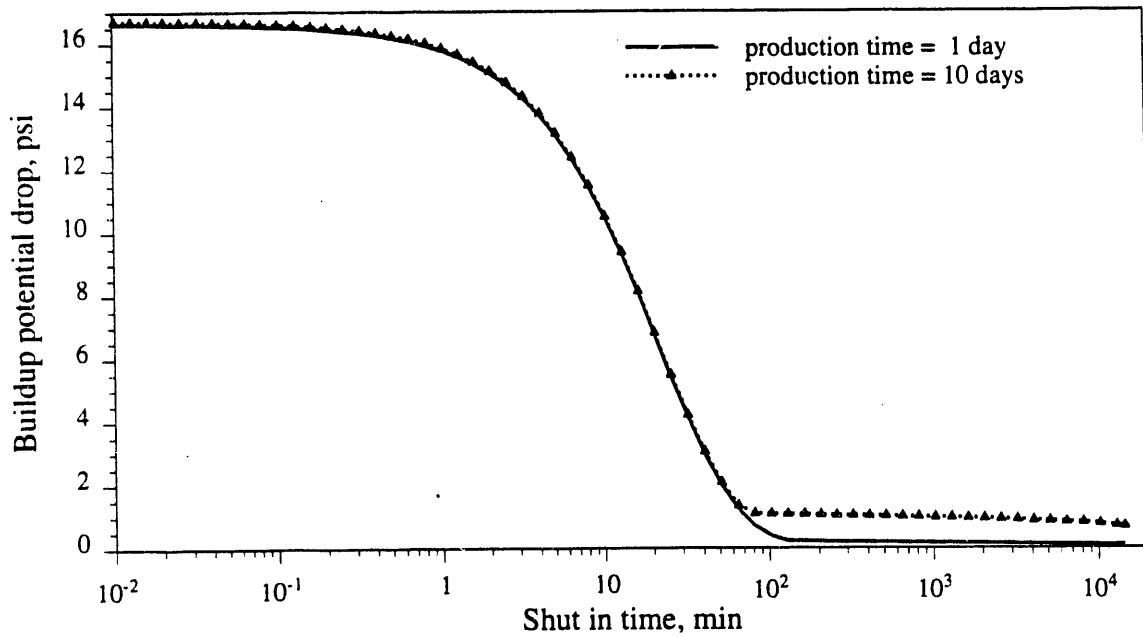


Figure 7.31: Buildup pressure drop behaviors for different production times

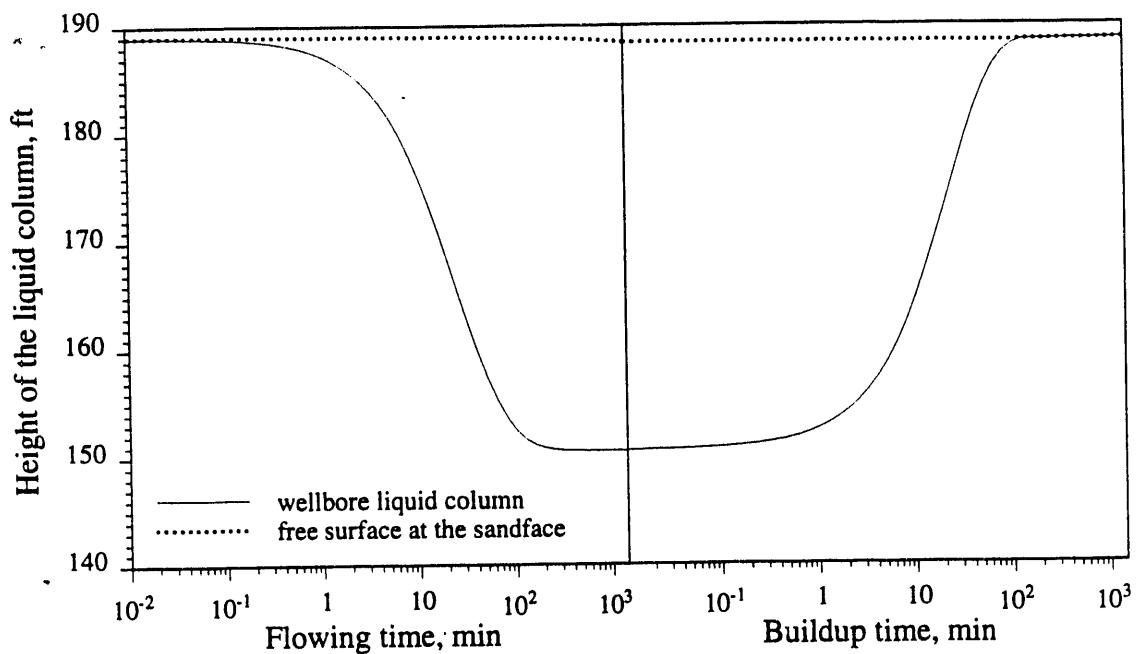


Figure 7.32: Liquid levels in the wellbore and at the sandface during drawdown and buildup periods

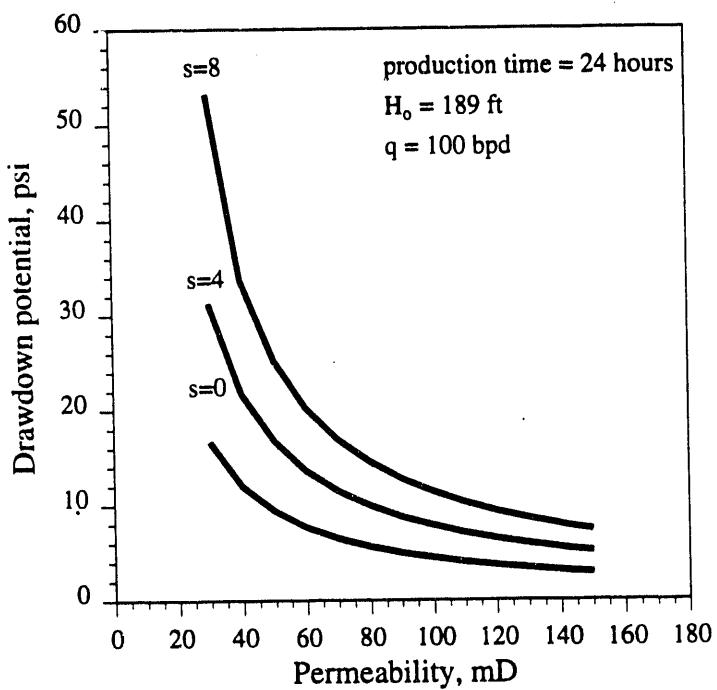


Figure 7.33: Potential drop in the well after a 24-hour production period.

8. Conclusions and Recommendations

8.1 Conclusions

- The Stream Layer Model is able to reproduce free surface gravity flow well pressure behavior and the formation response with accuracy, depending on the finite-difference discretization in time and space. All three recognized transient flow periods in a constant flow rate test, namely wellbore storage, growth of the desaturating cone, and the pseudoradial flow periods, are reproduced by the model.
- Steady-state flow was verified with the classical sandbox model experiments of *Wyckoff et al.* (1932), and *Hall* (1955). The seepage face phenomenon was reproduced. Also, verification runs compared interference test results for observation wells located far from the wellbore with the *Neuman* (1972) analytical approach and *Cooley* (1971) numerical solution.
- Capillary fringe effects, though negligible on field scales according to previous works, were important in the laboratory experiment verifications of *SLM*.
- The delayed yield period of a flat pressure-time response following wellbore storage was evident in all computer runs, with or without a capillary fringe. A capillary fringe is not necessary to generate a delayed yield effect. The major cause of delayed yield is vertical, viscous drainage. Analytical solutions without a vertical flow component do not exhibit delayed yield.
- The strong nonlinearities in the mathematical problem require a careful dimensioning of the grid parameters and time step sizes to obtain accuracy from the Stream Layer Model. Also, concerning the time interpolation parameter θ , the Crank-Nicolson method ($\theta = 0.5$) should be avoided because of the large number of iterations required for convergence.
- During the transient drawdown period, the seepage face length reached a maximum value at the end of wellbore storage which frequently masked the confined system-like behavior reported in the groundwater literature in the initial phase of production. After that, the liquid level in the formation adjacent to the wellbore starts to drop, as the potential gradients begin to increase. The pseudoconfined behavior is characterized by near stabilization of the potential gradients and free surface movement.
- The petroleum formation and liquid properties and well parameters tested by the *Stream Layer Model* caused the transient pressure drawdown behavior to persist longer than two or three log-cycles during the desaturating period. The pseudoconfined flow

behavior generally required weeks or months of production to be graphically represented in a logarithmic cycle. As a consequence, to use a logarithmic approximation such as in the *Ramey et al.* (1989) p-square approach, a long term drawdown test should be planned to assure correct application of the method.

- As an approximate solution at late time, the method of *Ramey et al.* (1989) presented slopes closer to the Stream Layer Model results than the *Theis* (1932) line source solution. Low permeability reservoirs tend to present an accentuated inflection in the p-squared analytical approach at late times, which is similar to the numerical results. High production flow rates also show the same trend.
- Computer runs for pressure buildup also showed delayed yield. However the seepage surface disappeared within a few minutes, and it appeared that the free surface desaturated region is refilled by radial flow, unlike the vertical drainage during production.
- Contrary to the drawdown results, the buildup pressures in the producing well generally require a short time to reach the resaturating period (equivalent to the desaturating period in the drawdown). Late time buildup pressures, however, could not be approximated by radial flow interpretation methods. The potential profiles in the radial direction taken at increasing buildup times show the distribution of the potential at the base with different slopes each time, tending to continuously reduce the radial potential gradient. The free surface profile shows uniform height from the wellbore into the reservoir during the buildup. This flat potential profile becomes extended with the shut-in time. The present study has shown that the buildup potential behavior in the producing well is not the reverse process of drawdown.

8.2 Recommendations

- The *Stream Layer Model* can be used for experimentation to aid developing a parametric correlation of dimensionless time and pressure drawdown in well test analysis, similar to the *Boulton* (1963) and *Neuman* (1972) correlations developed for observation wells. The use of the simulator is recommended for planning well tests for gravity drainage wells.
- The *SLM* simulator can be used to solve the semiconfined reservoir well problem. This simulator could be a useful tool to investigate the behavior of low pressure or low permeability pumping wells with dynamic liquid level reaching positions below the top of the formation and developing a free moving surface in the formation near the well. The computer program requires small modifications for this application.
- Wellbore pressure buildup analysis is a challenging matter remaining to be investigated, and an interpretation method should be studied using the simulator response for a physical understanding of the phenomenon. It is likely that computer aided interpretation with regression will be necessary.

- The observed anisotropic effects in the wellbore potential drawdown behavior by the *Stream Layer Model* may be used to produce an evaluation method to find vertical permeability of anisotropic reservoirs.

Nomenclature

A	=	area
A_1	=	areal dimensionless factor
A_v	=	cross-sectional area of the vertical flow
$A_{i,j}$	=	discretized equation coefficients
B_{wb}	=	formation volume factor of the wellbore liquid
$B_{i,j}$	=	discretized equation coefficient
c_t	=	total compressibility
c_L	=	liquid compressibility
$C_{i,j}$	=	discretized equation coefficient
$DA_{i,j}$	=	Jacobian matrix coefficient
$DB_{i,j}$	=	Jacobian matrix coefficient
$DC_{i,j}$	=	Jacobian matrix coefficient
$DD_{i,j}$	=	Jacobian matrix coefficient
$DE_{i,j}$	=	Jacobian matrix coefficient
$DF_{i,j}$	=	Jacobian matrix coefficient
$DG_{i,j}$	=	Jacobian matrix coefficient
$D_{i,j}$	=	discretized equation coefficient
$E_i(x)$	=	exponential integral
$E_{i,j}$	=	discretized equation coefficients
$F_{i,j}$	=	discretized equation coefficients
f_r	=	ratio of logarithms of radial distances
f_x	=	ratio of axial permeabilities
g	=	acceleration constant of the gravity force
g_c	=	dimensional conversion factor
$G_{i,j}$	=	wellbore discretized equation coefficients

H	=	vertical position of the free surface
H_o	=	original thickness at rest
H_s	=	vertical position of the liquid at the sandface
H_w	=	vertical position of the liquid at the wellbore
j_1	=	vertical location of the liquid level in the well
k	=	formation permeability
k_r	=	permeability in the flow direction
M	=	number of grid blocks in the radial direction
N	=	number of grid blocks in the vertical direction
p	=	pressure
q	=	flow rate
q_x	=	flow rate in the stream layer direction
q_{in}	=	flow rate into a grid block element
q_{out}	=	flow rate out of a grid block element
r	=	radial distance
r_w	=	wellbore radius
\bar{r}	=	radial distance of the gravity center of a block
s	=	skin effect parameter
S_g	=	gas saturation above the free surface
S_L	=	liquid saturation
S_w	=	water saturation
S_y	=	specific yield
S'	=	same as specific yield, S_y
T	=	transmissivity or transmissibility
t	=	time
V_{std}	=	volume of produced liquid at standard condition
V_{wb}	=	volume of produced liquid at wellbore condition
W	=	dimensionless potential
x_j	=	fraction of the inner block j_1 thickness above H_w
z	=	vertical coordinate
z_{r_w+}	=	vertical coordinate at the sandface

SUBSCRIPT

D	=	dimensionless
h	=	horizontal direction
i	=	horizontal position of a discretized block
j	=	vertical position of a discretized block
nw	=	non-wet or seepage region of sandface
o	=	initial condition
t	=	total
v	=	vertical direction
w	=	wellbore
wet	=	wet or drowned region of the sandface

SUPERSCRIPT

k	=	old time step
ν	=	level of iteration

GREEK

α	=	thickness interpolation parameter
β	=	Boulton dimensionless variable $= K_D r^2 / b^2$
δ	=	Dirac delta
δ_k	=	parameter: $\delta_{k1} = 1$ if $k = 1$, otherwise, $\delta_{k1} = 0$
δ_{j1}	=	parameter: $\delta_{j1} = 1$ if $j_1 = 1$; otherwise, $\delta_{j1} = 0$
Δ	=	finite difference operator
Δh	=	free surface block thickness at its gravity center
Δh_w	=	thickness of a free surface block at the sandface
$\overline{\Delta h}$	=	free surface block boundary thickness
Δz	=	thickness of a saturated block at its gravity center
Δz_w	=	thickness of a saturated block at the sandface
$\overline{\Delta h}$	=	saturated region block boundary thickness
ϵ	=	small residual number
Φ	=	potential
ϕ	=	porosity
γ	=	dimensionless liquid gradient
μ	=	liquid viscosity
θ	=	time relaxation parameter
ρ	=	density of the liquid
Ψ	=	residual equations

References

- [1] Agarwal, R. G., Al-Hussainy, R., and Ramey, H. J. Jr.: "An Investigation of Wellbore Storage and Skin Effect in Unsteady Liquid Flow: I. Analytical Treatment", *Soc. Pet. Eng. J.*, (1970) **249**, 279.
- [2] Amyx, J.W., Bass, D.M., Jr., and Whiting, R.W.: *Petroleum Reservoir Engineering*, McGraw Hill, Inc., New York (1960).
- [3] Aronofsky, J. S. and Jenkins, R.: "Unsteady Radial Flow of Gas Through Porous Media", *Jour. Appl. Mech.* (1953) **20**, 210.
- [4] Aziz, K. and Settari, A.: *Petroleum Reservoir Simulation*, Applied Science Publishers, London (1979).
- [5] Babbit, H.E. and Caldwell, D.H.: "The Free Surface Around, and Interface Between, Gravity Wells", *Engineering Experiment Station*, (1948) Bulletin Series, No. 374, University of Illinois.
- [6] Barenblatt, G. I., Zheltov Iu. P., and Kochina, I.F.: "Basic Concepts in the Theory of Seepage of Homogeneous Liquids in Fissured Rocks", *Jour. Appl. Math.* (1960), **24**(5), 1286.
- [7] Boulton, N. S.: "The Flow Pattern near a Gravity Well in a Uniform Water Bearing Medium", *J. Instn. Civ. Engrs* (London - 1951) **36**, 534.
- [8] Boulton, N. S.: "The Drawdown of the Water-Table Under Non-Steady Conditions Near a Pumped Well in an Unconfined Formation"; *Proc. Instn. Civ. Engrs.* (1954, a) **3**, Part III, 564.
- [9] Boulton, N. S.: "Unsteady Radial Flow to a Pumped Well Allowing for Delayed Yield from Storage", *Publication No. 37 de l'Association Internationale d'Hydrologie, Assemblée Générale de Rome* (1954, b) Tome **II**, 472.
- [10] Boulton, N. S.: "Analysis of Data from Non-Equilibrium Pumping Tests Allowing for Delayed Yield from Storage", *Proc. Instn. Civ. Engrs.* (London - 1963) **26**, 469.
- [11] Boulton, N. S.: "Analysis of Data from Pumping Tests in Unconfined Anisotropic Aquifers", *J. Hydrology* (1970), **10**, 369.

- [12] Boulton, N. S., and Pontin, J. M. A.: "The Influence of Delayed Drainage on Data from Pumping Tests in Unconfined Anisotropic Aquifers", *J. Hydrology* (1971), **19**(1), 157.
- [13] Bouwer, H.: "Soil Water Hysteresis as a Cause of Delayed Yield from Unconfined Aquifers", *Groundwater* (1979), **15**(4), 965.
- [14] Bouwer, H., and Rice, R. C.: "A Slug Test for Determining Hydraulic Conductivity of Unconfined Aquifers with Completely or Partially Penetrating Wells", *Water Resources Res.* (1976), **12**(3), 423.
- [15] Bouwer, H., and Rice, R. C.: "Delayed Aquifer Yield as a Phenomenon of Delayed Air Entry", *Water Resources Res.* (1978), **14**(6), 1068.
- [16] Brand, C., Girard, P., Nacul E. C., and Aziz, K.: "Efficient Solvers for Reservoir Simulation", 8th Stanford Reservoir Simulation Workshop, May 16, 1992.
- [17] Carslaw, H. S. and Jaeger, J. C.: *Conduction of Heat in Solids*, Oxford University Press (1959).
- [18] Cooper, H. H. Jr., Bredehoeft, J.D., and Papadopoulos, I.S.: "Response of a Finite Diameter Well to an Instantaneous Charge of Water", *Water Resour. Res.* (1967), **3**, 263-269.
- [19] Cooley, R. L.: "A Finite Difference method for Unsteady Flow in Variably Saturated Porous Media: Application to a Single Pumping Well", *Water Resour. Res.* (1971), **7**(6), 1607.
- [20] Cooley, R. L., and Donohue, D. A. T.: "Numerical Simulation of Unconfined Flow into a Single Pumping Water-Well Using Two-phase Flow Theory", Technical Completion Report, Inst. for Res. on Land and Water Resour., Pa. State Univ., State College, 1969.
- [21] Dagan, G.: "Second Order Linearized Theory of Free-Surface Flow in Porous Media", *Houille Blanche* (1964), **8**, 901.
- [22] Dagan, G.: "A Method of Determining the Permeability and Effective Porosity of Unconfined Anisotropic Aquifers", *Water Resour. Res.* (1967), **3**(4), 1059.
- [23] Dagan, G.: "A Note on Packer, Slug, and Recovery Tests in Unconfined Aquifer", *Water Resour. Res.* (1978), **14**(5), 929.
- [24] Dieleman, P. J.; Riedder N. A. and Wesseling, J.: "Drainage Principles and Applications", *Int'l Inst. for Land Reclamation and Improvement*, Wageningen The Netherlands (1972) **1**, 153.
- [25] Dupuit, J.: *Etudes Theoriques et Pratiques sur le Mouvement des Eaux dans l Couverts et a Travers les Terrains Permeables*, 2nd edn, Dunod, Paris (1863).

[26] Earlougher, R. C., Jr.: *Advances in Well Test Analysis*, SPE Monograph, Henry L. Doherty Series, v. 5, New York, 1977.

[27] Forchheimer, P.: *Zeits. Arch. Ing. Ver.*, Hannover (1886).

[28] Fligelman, H.: "Drawdown and Interference Test Analysis for Gas Wells with Wellbore Storage, Damage, and Nonlaminar Flow Effects", PhD dissert., Stanford Univ. (1980).

[29] Gambolati, G.: "Transient Free Surface Flow to a Well: An Analysis of Theoretical Solutions", *Water Resour. Res.* (1976) **12**(1), 27.

[30] Green, D. W., Dabiri, H., Weinaug, C. F. and Prill, R. : "Numerical Modeling of Unsaturated Groundwater Flow and Comparison of the Model to a Field Experiment", *Water Resour. Res.* (1970) **6**(3), 862.

[31] Hall, H.P.: "An Investigation of Steady Flow toward a Gravity Well", *La Houille Blanche* (1955) **25**, 8-35.

[32] Hantush, M. S.: "Hydraulics of Wells", *Advances in Hydroscience*, 1, edited by V .T. Chow, Academic Press, NY (1964).

[33] Hawkins, M. F.: "A Note on the Skin Effect", *Trans.*, AIME (1956), **207**, 356-357.

[34] Hornberger, G. M., and Remson, I.: "A Moving Boundary Model of a One-dimensional Saturated-Unsaturated, Transient Porous Flow System", *Water Resour. Res.* (1970) **6**(3), 898.

[35] Horner, D. R.: "Pressure Buildup in Wells", Proc. 3rd. World Pet. Cong., E. J. Brice. London. Heiden, Neth. (1951) Sect II, Preprint 7.

[36] Hurst, W.: "Establishment of the Skin Effect and its Impediment to Fluid-Flow Into a Well Bore", *Pet. Eng.* (1953) **25**, B-6.

[37] Kipp, K. L.: "Unsteady Flow to a Partially Penetrating Finite Radius Well in an Unconfined Aquifer", *Water Resour. Res.* (1973) **9**(2), 448.

[38] Kroszynski, U. I., and Dagan, G.: "Well Pumping in Unconfined Aquifers: The Influence of the Unsaturated Zone", *Water Resour. Res.* (1975) **11**(3), 479.

[39] Matthews, C. S., and Lefkovits, H. C.: "Gravity Drainage Performance of Depletion-Type Reservoirs in the Stripper Stage", *Trans.*, AIME (1956), **207**, 263-274.

[40] Moore, T. V., Shilthuis, R. J. and Hurst, W.: "The Determination of Permeability from Field Data", *Proc.*, API Meeting, Tulsa, Okla. (May 17-19, 1933) **4**.

[41] Muskat, M.: "The Seepage of Water Through Dams with Vertical Faces", *Physics* (1935) **6**, 402.

- [42] Muskat, M.: *Flow of Homogeneous Fluids Through Porous Media*, I.H.R.D.C., Boston (1937-1982).
- [43] Muskat, M.: *Physical Principles of Oil Production*, McGraw Hill Book Company, Inc., New York (1949).
- [44] Narasimhan, T.N. and Zhu, M.: "Transient Radial Flow to a Well in an Unconfined Aquifer, 1. Applicability of Some Conceptual Models", *Water Resour. Res.*, in press.
- [45] Neuman, S. P., and Witherspoon, P. A.: "Finite Element Method of Analyzing Steady Seepage with a Free Surface", *Water Resour. Res.* (1970-a) **6**(5), 889.
- [46] Neuman, S. P., and Witherspoon, P. A.: "Variational Principles for Confined and Unconfined Flow of Ground Water", *Water Resour. Res.* (1970-b) **6**(5), 1376.
- [47] Neuman, S. P., and Witherspoon, P. A.: "Analysis of Nonsteady Flow with a Free Surface Using the Finite Element Method", *Water Resour. Res.* (1971) **7**(3), 611.
- [48] Neuman, S. P.: "Theory of Flow in Unconfined Aquifers Considering Delayed Response of the Water Table", *Water Resour. Res.* (1972) **8**(4), 1031.
- [49] Neuman, S. P.: "Supplementary Comments on 'Theory of Flow in Unconfined Aquifers Considering Delayed Response of the Water Table'", *Water Resour. Res.* (1973) **9**(4), 1102.
- [50] Neuman, S. P.: "Effect of Partial Penetration on Flow in Unconfined Aquifers Considering Delayed Gravity Response", *Water Resour. Res.* (1974) **10**(2), 303.
- [51] Neuman, S. P., Feedes, R. E., and Bresler, E.: "Finite Element Simulation of Flow in Saturated-Unsaturated Soils Considering Water Uptake by Plants" completion rep., Proj. ALO-SWC-77 (1974), Hydrodyn. and Hydraul. Eng. Lab., Haifa, Israel.
- [52] Neuman, S. P.: "Analysis of Pumping Test Data From Anisotropic Unconfined Aquifers Considering Delayed Response", *Water Resour. Res.* (1975-a) **11**(2), 329.
- [53] Neuman, S. P.: "Galerkin Approach to Saturated-Unsaturated Flow in Porous Media", *Finite Element in Fluids*, vol. 1, edited by R. H. Gallagher, J. T. Oden, C. Taylor, and O. C. Zienkiewicz, John Wiley, New York, 1975-b.
- [54] Neuman, S. P.: Reply, *Water Resour. Res.* (1976) **12**(1), 115.
- [55] Neuman, S. P.: "Perspective on Delayed Yield", *Water Resour. Res.* (1979) **15**(4), 899.
- [56] Neuman, S. P.: "On Methods of Determining Specific Yield", *Groundwater* (1987) **25**(6), 67.
- [57] Nwankwor, G. I., Cherry J. A., and Gillham, R. W.: "A Comparative Study of Specific Yield Determinations for a Shallow Sand Aquifer", *Groundwater* (1984) **22**(6), 764.

[58] Peaceman, D. W.: *Fundamentals of Numerical Reservoir Simulation*, Elsevier Sci. Pub. Co., New York (1977).

[59] Papadopoulos, I. S.: "Drawdown Distribution around a Large Diameter Well", *Proc. Symp. on Groundwater Hydrology*, San Francisco, Nov., *Proc. Am. Water Resources Assoc.* (1967) **4**, 157-168.

[60] Papadopoulos, I. S. and Cooper, H. H., Jr.: "Drawdown in a Well of Large Diameter", *Water Resour. Res.* (1967) **3**(1), 241.

[61] Patel, S. C., and Mishra, G. C.: "Analysis of Flow to a Large-Diameter Well by a Discrete Kernel Approach", *Groundwater* (1983) **21** (5), 573.

[62] Polubarnova-Kochina, P.YA.: *Theory of Ground Water Movement*, Translated from the russian by J. M. Roger de Wiest, Princeton University Press, Princeton, NY (1962).

[63] Prickett T. A.: "Type Curve Solution to Aquifer Tests Under Water-Table Conditions", *Groundwater* (1965) **3**(3), 5.

[64] Ramey, H. J., Lichtenberg, S., and Davitt, J.: "Well Testing Analysis for Gravity Drainage System", Paper SPE No. 890017, *Proceedings of Petroleum Technology into the Second Century Meeting*, Socorro, New Mexico, Oct., 1989.

[65] Rushton, K. R., and Holt, S. M.: "Estimating Aquifer Parameters for Large-Diameter Wells", *Groundwater* (1981) **19**(5), 505.

[66] Rushton, K. R., and Singh, V. S.: "Pumping Test Analysis in Large Diameter Wells with a Seepage Face by Kernel Function Technique", *Groundwater* (1987) **25**(1), 81.

[67] Shamsai, A. and Narasimhan, T. N.: "A Numerical Investigation of Free Surface- Seepage Face Relationship Under Steady State Flow Conditions", *Water Resour. Res.* (1991) **27**(3), 409.

[68] Stehfest, H.: "Algorithm 368: Numerical Inversion of the Laplace Transform", *Communication of the A. C. M.*, **1** (13), (August 1970).

[69] Streltsova, T. D.: "Unconfined Aquifer and Slow Drainage", *J. Hydrology* (1972-a) **16**, 117.

[70] Streltsova, T. D.: "Unsteady Radial Flow in an Unconfined Aquifer", *Water Resour. Res.* (1972-b) **8**(4), 1059.

[71] Streltsova, T. D.: "Flow near a Pumped Well in an Unconfined Aquifer Under Non- steady Conditions", *Water Resour. Res.* (1973) **9**(1), 227.

[72] Streltsova, T. D.: "Drawdown in a Compressible Unconfined Aquifer", *Div., Amer. Soc. Civil Eng.*, **100**(HY11), 1974.

- [73] Streltsova, T. D.: "Comments on 'Analysis of Pumping Test Data from Anisotropic Unconfined Aquifers Considering Delayed Gravity Response' by Shlomo P. Neuman", *Water Resour. Res.* (1976) **9**(1), 236.
- [74] Streltsova, T. D. and Rushton, K. R.: "Water Table Drawdown Due to a Pumped Well in an Unconfined Aquifer", *Water Resour. Res.* (1973) **9**(1), 236.
- [75] Szabo, B. A. and McCaig, I. W.: "A Mathematical Model for Transient Free Surface Flow in Nonhomogeneous Porous Media", *Water Res. Bull.* (1968), **4** part 3.
- [76] Terán, M. A. and de la Garza, F. R.: "Efecto de la Geometria de la Malla sobre la Estabilidad de un Simulador Numerico", *Ingenieria Petrolera* (1988) **28**, 10.
- [77] Theis, C. V: "The Relation between the Lowering of the Piezometric Surface and the rate of Duration of Discharge of a Well using Ground-Water Storage", *Trans. Amer. Geophys. Un.* (1935) **16**, 519.
- [78] Thomas, G. W.: *Hydrocarbon Reservoir Simulation*, Int. Human Res. Dev. Corp., Boston (1982).
- [79] van Everdingen, A. F. and Hurst, W.: "The Application of the Laplace Transformation to Flow Problems in Reservoirs", *Trans. AIME* (1949) **186**, 305-324.
- [80] van Everdingen, A. F.: "The Skin Effect and Its Influence on the Productive Capacity of a Well", *Trans. AIME* (1953) **198**, 171-176.
- [81] Walton, W. C.: "Application and Limitations of Methods Used to Analyse Pumping Test Data", part **2**, *Water Well J.* (1960) March, 45.
- [82] 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. Inner Boundary Condition

A.1 Partial Derivatives of the Sandface Potential

Potentials at the sandface are a function of the vertical coordinate and of the liquid level in the wellbore. The criteria used to define the inner boundary conditions are shown in dimensionless form:

$$W_w = \gamma_o z_{w_D} \Rightarrow H_{w_D} < z_{w_D} \leq H_{s_D} \quad (\text{A.1.1})$$

A finite difference form of the equation and grid definitions of the reservoir requires that the inner boundary be reordered into different regions, according to the position of the liquid level in the wellbore at each time-step. A vertical grid parameter j_1 was defined as the grid position at the sandface adjacent to the wellbore liquid level. The potentials were calculated:

$$W_{w_j} = \gamma_o H_{w_D} \Rightarrow j > j_1 \quad (\text{A.1.2})$$

$$W_{w_j} = \gamma_o \left(z_{w_{D_j}} + \frac{\Delta h_{D_w}}{2} \right) \Rightarrow 1 = j < j_1 \quad (\text{A.1.3})$$

$$W_{w_j} = \gamma_o \left(z_{w_{D_j}} + \frac{\Delta z_{D_w}}{2} \right) \Rightarrow 1 \neq j < j_1 \quad (\text{A.1.4})$$

$$W_{w_j} = \frac{\gamma_o}{2} \left[(1 + x_{j_1}) H_{w_D} + (1 - x_{j_1}) (z_{w_{D_{j_1}}} + \Delta z_{D_w}) \right] \Rightarrow j = j_1 \neq 1 \quad (\text{A.1.5})$$

where:

$$x_{j_1} = \frac{H_{w_D} - z_{w_{D_{j_1}}}}{\Delta z_{D_w}} \quad (\text{A.1.6})$$

$$W_{w_j} = \frac{\gamma_o}{2} \left[(1 + x_{j_1}) H_{w_D} + (1 - x_{j_1}) (z_{w_{D_1}} + \Delta h_{D_w}) \right] \Rightarrow j = j_1 = 1$$

(A.1.7)

where:

$$x_{j_1} = \frac{H_{w_D} - z_{w_{D_1}}}{\Delta h_{D_w}} \quad (A.1.8)$$

Partial differentials with respect to $W_{1,k}$ were calculated from the above definitions. Initially, k is set different from 1:

(a) $j > j_1$:

$$\frac{\partial W_{w_j}}{\partial W_{1,\kappa}} = \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} \quad (A.1.9)$$

(b) $j < j_1$

$$\frac{\partial W_{w_j}}{\partial W_{1,\kappa}} = 0 \quad (\text{A.1.10})$$

(c) $j = 1$ and $1 < j_1$

$$\frac{\partial W_{w_j}}{\partial W_{1,\kappa}} = 0 \quad (\text{A.1.11})$$

(d) $j = j_1$ and $j_1 \neq 1$

$$\frac{\partial W_{w_j}}{\partial W_{1,\kappa}} = x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} \quad (\text{A.1.12})$$

(e) $j = j_1 = 1$

$$\frac{\partial W_{w_j}}{\partial W_{1,\kappa}} = x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} \quad (\text{A.1.13})$$

Partial differentiations with respect to $W_{1,1}$ are particular cases in which there is a dependence of several parameters on the potential:

(a) $j > j_1$

$$\frac{\partial W_{w_j}}{\partial W_{1,1}} = \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,1}} \quad (\text{A.1.14})$$

(b) $j < j_1$

$$\frac{\partial W_{w_j}}{\partial W_{1,1}} = 0 \quad (\text{A.1.15})$$

(c) $j = 1$ and $1 < j_1$

$$\frac{\partial W_{w_j}}{\partial W_{1,1}} = \frac{\gamma_o \alpha_w}{2 \bar{\gamma}_1} \quad (\text{A.1.16})$$

(d) $j = j_1$ and $j_1 \neq 1$

$$\frac{\partial W_{w_j}}{\partial W_{1,1}} = x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,1}} \quad (\text{A.1.17})$$

(e) $j = j_1 = 1$

$$\frac{\partial W_{w_j}}{\partial W_{1,1}} = \gamma_o \left[x_{j_1} \frac{\partial H_{w_D}}{\partial W_{1,1}} + (1 - x_{j_1}^2) \frac{\alpha_w}{2 \bar{\gamma}_1} \right] \quad (\text{A.1.18})$$

In the last case, Δh_{D_w} is also directly dependent on $W_{1,2}$. Thus, another condition must be considered when $j = 1$:

(a) $j = 1$ and $j_1 > 1$

$$\frac{\partial W_{w_j}}{\partial W_{2,1}} = \frac{\gamma_o (1 - \alpha_w)}{2 \bar{\gamma}_2} \quad (\text{A.1.19})$$

(b) $j = 1$ and $j_1 = 1$

$$\frac{\partial W_{w_j}}{\partial W_{2,1}} = \gamma_o \left[x_{j_1} \frac{\partial H_{w_D}}{\partial W_{2,1}} + (1 - x_{j_1}^2) \frac{1 - \alpha_w}{2 \bar{\gamma}_2} \right] \quad (\text{A.1.20})$$

A.2 Liquid level in the Wellbore from Material Balance Equation

From the material balance discretized equation and the substitution of the sandface potentials, it is possible to determine the liquid level in the wellbore within a time-step. Guessing the position j_1 and a global iterative procedure enable one to estimate the value of H_{w_D} .

There are two ways to determine the liquid level in the well, starting from the material balance equation represented by Eq.3.2.5 in the Ch. 3, presented below.

A.2.1 Quadratic Equation

A quadratic equation is obtained from the discretization of Eq. 3.2.5 starting from the introduction of a time interpolation parameter as follows:

$$q B_{wb} = -\frac{\pi r_w^2 \Delta H_w}{2 \phi \Delta t} + \frac{2 \pi}{\mu \ln(\bar{\gamma}_1)} \left[\theta \int_0^{h_s} k_x \Delta \Phi|_{r=r_w} dz + (1 - \theta) \int_0^{h_s} k_x \Delta \Phi^k|_{r=r_w} dz \right]$$

(A.2.1.1)

Multiply by $\frac{\mu}{2\pi r_w k_h H_o \rho g_c}$ to obtain the dimensionless expression:

$$q_D B_{wb} = -\frac{H_{wD} - H_{wD}^k}{2\phi \Delta t_D} + \frac{1}{k_h \ln(\bar{r}_1)} \left[\theta \int_0^{H_s} k_x \Delta W \Big|_{r_D=1} dz_D + (1-\theta) \int_0^{H_s} k_x \Delta W^k \Big|_{r_D=1} dz_D \right] \quad (\text{A.2.1.2})$$

A quadratic equation can be obtained isolating H_{wD} . The following parameter is introduced in order to handle all possible positions of the liquid level in the wellbore with a single equation:

$$\begin{cases} \delta = 1 & \text{as } j_1 = 1 \\ \delta = 0 & \text{as } j_1 \neq 1 \end{cases}$$

Next, a proportionality factor x_j is introduced to average the potential at the sand-face position adjacent to the liquid surface in the wellbore:

$$x_j = \frac{H_{wD} - z_{D_{w,j_1}}}{\delta \Delta h_{D_w} + (1-\delta) \Delta z_{D_w}} \quad (\text{A.2.1.3})$$

Using the parameters δ and x_j , the integrals can be simplified by calculating average permeabilities and factoring the average permeability out of the the integrands:

$$\bar{k}_{x_w}^k = \frac{\left(x_j^k k_{x_1,j_1} + \sum_{j_1+1}^n (k_{x_1,j}) \right) \bar{\Delta z}_{D_1}}{H_{wD}^k} \quad (\text{A.2.1.4})$$

$$\bar{k}_{x_{nw}}^k = \frac{\left[(1-x_j^k) k_{x_1,j_1} + \sum_1^{j_1-1} (k_{x_1,j}) \right] \bar{\Delta z}_{D_1}}{H_s^k - H_{wD}^k} \quad (\text{A.2.1.5})$$

$$\bar{k}_{x_w} = \frac{x_j k_{x_1,j_1} [\delta \bar{\Delta h}_{D_1} + (1-\delta) \bar{\Delta z}_{D_1}] + \bar{\Delta z}_{D_1} \sum_{j_1+1}^n (k_{x_1,j})}{H_{wD}} \quad (\text{A.2.1.6})$$

$$\begin{aligned} \bar{k}_{x_{nw}} = & \frac{1}{H_s - H_{wD}} \left\{ (1-x_j) k_{x_1,j_1} [\delta \bar{\Delta h}_{D_1} + (1-\delta) \bar{\Delta z}_{D_1}] + \bar{\Delta z}_{D_1} \sum_2^{j_1-1} (k_{x_1,j}) \right. \\ & \left. + (1-\delta) \bar{\Delta h}_{D_1} + k_{x_1,1} \right\} \end{aligned} \quad (\text{A.2.1.7})$$

The superscript k means the beginning of the time-step, and the subscripts nw and wet represent the seepage face (*non-wet*) and the liquid submerged (*wet*) wellbore regions, respectively. A first decomposition of the material balance equation at the wellbore gives the following:

$$\begin{aligned}
 q_D B_{wb} = & -\frac{H_{wD} - H_{wD}^k}{2\phi \Delta t_D} + \frac{1}{k_h \ln(\bar{r}_1)} \left\{ \theta \left[\bar{\Delta h}_{D1} + k_{x_{1,1}} W_{1,1} + \right. \right. \\
 & \left. \left. \Delta \bar{z}_{D1} \sum_2^n (k_{x_{1,j}} W_{1,j}) - \frac{H_{wD}^2}{H_{oD}} \bar{k}_{x_w} - \frac{H_s^2 - H_{wD}^2}{2H_{oD}} \bar{k}_{x_{nw}} \right] + \right. \\
 & \left. (1 - \theta) \left[\bar{\Delta z}_{D1} \sum_1^n k_{x_{1,j}} W_{1,j}^k - \frac{H_{wD}^k}{H_{oD}} \bar{k}_{x_w}^k - \frac{H_s^{k2} - H_{wD}^k}{2H_{oD}} \bar{k}_{x_{nw}}^k \right] \right\} \\
 \end{aligned} \tag{A.2.1.8}$$

Rearranging the terms in the above expression, a quadratic equation is obtained in the format:

$$A H_{wD}^2 + B H_{wD} + C = 0 \tag{A.2.1.9}$$

in which the coefficients are easily identified. This equation gives two possible solutions. Negative values are always neglected, and if there are two positive roots, some logical criteria of analysis are introduced in the computer program.

$$\begin{aligned}
 & \frac{\theta}{H_{oD} k_h \ln(\bar{r}_1)} \left[\bar{k}_{x_w} - \frac{\bar{k}_{x_{nw}}}{2} \right] H_{wD}^2 + \frac{1}{2\phi \Delta t_D} H_{wD} - \frac{H_{wD}^k}{2\phi \Delta t_D} + q_D B_{wb} - \\
 & \frac{1}{k_h \ln(\bar{r}_1)} \left\{ \theta \left[\bar{\Delta h}_{D1} k_{x_{1,1}} W_{1,1} + \bar{\Delta z}_{D1} \sum_2^n (k_{x_{1,j}} W_{1,j}) - \frac{H_s^2}{2H_{oD}} \bar{k}_{x_{nw}} \right] + \right. \\
 & \left. (1 - \theta) \left[\bar{\Delta z}_{D1} \sum_1^n k_{x_{1,j}} W_{1,j}^k - \frac{H_{wD}^k}{H_{oD}} \bar{k}_{x_w}^k - \frac{H_s^{k2} - H_{wD}^k}{2H_{oD}} \bar{k}_{x_{nw}}^k \right] \right\} = 0
 \end{aligned} \tag{A.2.1.10}$$

A.2.2 Iterative Solution for the Wellbore Liquid Level

An alternative way to solve the wellbore liquid level equation is to use an iterative process as the Newton-Raphson approach. This procedure gives only one of two roots for H_{w_D} from the quadratic equation, and the initial guess to start the iterative process is of fundamental importance for the usefulness of the result. The procedure starts from the discretized material balance equation below:

$$\begin{aligned}
q_D B_{w_b} = & -\frac{\Delta H_{w_D}}{2\phi\Delta t_D} + \frac{\overline{\Delta z}_{D_1}}{\ln(\bar{r}_1)} \left\{ \frac{k_{x_{1,1}}}{k_h} (1-\theta) [W_{1,1}^k - W_{w_1}^k] + \right. \\
& \left. \sum_{j=2}^n \frac{k_{x_{1,j}}}{k_h} [\theta [W_{1,j} - W_{w_j}] + (1-\theta) [W_{1,j}^k - W_{w_j}^k]] \right\} + \\
& \frac{\overline{\Delta h}_{D_1}}{\ln(\bar{r}_1)} \frac{k_{x_{1,1}}}{k_h} \theta [W_{1,1} - W_{w_1}]
\end{aligned} \tag{A.2.2.1}$$

Call the function:

$$\begin{aligned}
\mathcal{F}(H_{w_D}^v) = & q_D B_{w_b} + \frac{H_{w_D}^v - H_{w_D}^k}{2\phi\Delta t_D} - \frac{\overline{\Delta z}_{D_1}}{\ln(\bar{r}_1)} \left\{ \frac{k_{x_{1,1}}}{k_h} (1-\theta) [W_{1,1}^k - W_{w_1}^k] \right. \\
& \left. + \sum_{j=2}^n \frac{k_{x_{1,j}}}{k_h} [\theta [W_{1,j} - W_{w_j}] + (1-\theta) [W_{1,j}^k - W_{w_j}^k]] \right\} \\
& - \frac{\overline{\Delta h}_{D_1}}{\ln(\bar{r}_1)} \frac{k_{x_{1,1}}}{k_h} \theta [W_{1,1} - W_{w_1}]
\end{aligned} \tag{A.2.2.2}$$

and its derivative:

$$\begin{aligned}
\mathcal{F}'(H_{w_D}) = & \frac{\partial \mathcal{F}(H_{w_D})}{\partial H_{w_D}} = q_D \frac{\partial B_{w_b}}{\partial H_{w_D}} + \frac{1}{2\phi\Delta t_D} + \frac{\theta}{k_h \ln(\bar{r}_1)} \times \\
& \left\{ \overline{\Delta z}_{D_1} \sum_{j_1+1}^n k_{x_{1,j}} \frac{\partial W_{w_j}}{\partial H_{w_D}} + [(1-\delta) \overline{\Delta z}_{D_1} + \delta \overline{\Delta h}_{D_1}] k_{x_{1,j_1}} \frac{\partial W_{w_{j_1}}}{\partial H_{w_D}} \right\}
\end{aligned} \tag{A.2.2.3}$$

where δ is defined by the grid position of the liquid level in the wellbore j_1 :

$$\delta = 1 \Rightarrow j_1 = 1 \tag{A.2.2.4}$$

$$\delta = 0 \Rightarrow j_1 > 1 \tag{A.2.2.5}$$

and:

$$\frac{\partial B_{w_b}}{\partial H_{w_D}} = - \frac{\gamma_o c_{L_D} B_{w_b}}{4 + (H_{w_D}^v + H_{w_D}^k) \gamma_o c_{L_D}} \quad (\text{A.2.2.6})$$

From the definition of the inner boundary potentials:

$$W_{w_{j_1}} = \gamma_o \left[\frac{1 + x_{j_1}}{2} H_{w_D} + \frac{1 - x_{j_1}}{2} (z_{w_{D_{j_1-1}}}) \right] \quad (\text{A.2.2.7})$$

and:

$$\begin{aligned} \frac{\partial W_{w_{j_1}}}{\partial H_{w_D}} &= \gamma_o \left[\frac{1 + x_{j_1}}{2} - \frac{1 - x_{j_1}}{2} [(1 - \delta) \Delta z_{D_w} + \delta \Delta h_{D_w}] \frac{\partial x_{j_1}}{\partial H_{w_D}} \right] \\ &= \gamma_o x_{j_1} \end{aligned} \quad (\text{A.2.2.8})$$

Thus:

$$\begin{aligned} \mathcal{F}'(H_{w_D}) &= - \frac{q_D \gamma_o c_{L_D} B_{w_b}}{4 + (H_{w_D}^v + H_{w_D}^k) \gamma_o c_{L_D}} + \frac{1}{2\phi \Delta t_D} + \frac{\theta \gamma_o}{k_h \ln(\bar{r}_1)} \times \\ &\left\{ \bar{\Delta z}_{D_1} \left[x_{j_1} k_{x_{1,j_1}} (1 - \delta) + \sum_{j_1+1}^n k_{x_{1,j}} \right] + \delta x_{j_1} \bar{\Delta h}_{D_1} k_{x_{1,j_1}} \right\} \end{aligned} \quad (\text{A.2.2.9})$$

From the Newton-Raphson iteration approach:

$$H_{w_D}^{v+1} = H_{w_D}^v - \frac{\mathcal{F}(H_{w_D}^v)}{\mathcal{F}'(H_{w_D}^v)} \quad (\text{A.2.2.10})$$

where v is the iterative level. Convergence is achieved when the difference below becomes small, compared to a residue ϵ :

$$H_{w_D}^{v+1} - H_{w_D}^v \leq \epsilon \quad (\text{A.2.2.11})$$

A.3 Material Balance Equation Partial Derivatives

The discretized wellbore material balance equation in App. A.2 may be differentiated with respect to each inner block potential $W_{1,\kappa}$, to evaluate the partial differentials of the sandface potentials, which were written as functions of H_{w_D} and z_{w_D} in App. A.1. Thus $\partial H_{w_D} / \partial W_{1,\kappa}$ may be substituted into the equations presented in App. A.1. A step-by-step development of the Jacobian equations is presented next.

The discretized material balance equation in App. A.2 is:

$$q_D B_{w_b} = -\frac{\Delta H_{w_D}}{2\phi \Delta t_D} + \frac{\overline{\Delta z}_{D_1}}{\ln(\bar{r}_1)} \left\{ \frac{k_{x_{1,1}}}{k_h} (1-\theta) [W_{1,1}^k - W_{w_1}^k] + \sum_{j=2}^n \frac{k_{x_{1,j}}}{k_h} [\theta [W_{1,j} - W_{w_j}] + (1-\theta) [W_{1,j}^k - W_{w_j}^k]] \right\} + \frac{\overline{\Delta h}_{D_1}}{\ln(\bar{r}_1)} \frac{k_{x_{1,1}}}{k_h} \theta [W_{1,1} - W_{w_1}] \quad (A.3.1)$$

Differentiating Eq. A.3.1 with respect to $W_{1,\kappa}$:

$$\begin{aligned} q_D \frac{\partial B_{w_b}}{\partial W_{1,\kappa}} &= -\frac{1}{2\phi \Delta t_D} \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} + \frac{\theta}{k_h \ln(\bar{r}_1)} \\ &\left\{ \overline{\Delta z}_{D_1} \left[(1 - \delta_\kappa) k_{x_{1,\kappa}} - \sum_{j_1+1}^n k_{x_{1,j}} \frac{\partial W_{w_j}}{\partial W_{1,\kappa}} - \delta_{j_1} k_{x_{1,j_1}} \frac{\partial W_{w_{j_1}}}{\partial W_{1,\kappa}} \right] \right. \\ &\left. + \overline{\Delta h}_{D_1} k_{x_{1,1}} \left[\delta_\kappa - \frac{\partial W_{w_1}}{\partial W_{1,\kappa}} \right] + k_{x_{1,1}} (W_{1,1} - W_{w_1}) \frac{\partial \overline{\Delta h}_{D_1}}{\partial W_{1,\kappa}} \right\} \end{aligned} \quad (A.3.2)$$

Also:

$$\alpha_1 = \frac{1 + \alpha_w}{2} \quad (A.3.3)$$

and:

$$\frac{\partial \overline{\Delta h}_{D_1}}{\partial W_{1,\kappa}} = \delta_\kappa \alpha_1 \frac{\partial \Delta h_1}{\partial W_{1,\kappa}} = \frac{\delta_\kappa \alpha_1}{\bar{\gamma}_1} \quad (A.3.4)$$

From the derivatives in App. A.1:

$$\begin{aligned} \frac{\partial W_{w_1}}{\partial W_{1,\kappa}} &= (1 - \delta_{j_1}) x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,1}} \\ &+ \delta_k \frac{\gamma_o \alpha_w}{2 \bar{\gamma}_1} \left[\delta_{j_1} + (1 - \delta_{j_1}) (1 - x_{j_1}^2) \right] \end{aligned} \quad (A.3.5)$$

The parameters δ_κ and δ_{j_1} are:

$$\begin{cases} \delta_\kappa = 1 \rightarrow \kappa = 1 \\ \delta_\kappa = 0 \rightarrow \kappa \neq 1 \end{cases} \quad \begin{cases} \delta_{j_1} = 1 \rightarrow j_1 \neq 1 \\ \delta_{j_1} = 0 \rightarrow j_1 = 1 \end{cases} \quad (A.3.6)$$

Equation A.3.5 can be written as a function of H_{w_D} as the only unknown by differentiating B_{w_b} and by using the relationships defined in App. A.1 for the potential differentiations:

$$\begin{aligned} - \frac{q_D B_{w_b} \gamma_o c_{L_D}}{4 + \gamma_o (H_{w_D} + H_{w_D}^k) c_{L_D}} \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} &= - \frac{1}{2 \phi \Delta t_D} \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} + \frac{\theta}{k_h \ln(\bar{r}_1)} \times \\ & \left\{ \overline{\Delta z}_{D_1} \left[(1 - \delta_\kappa) k_{x_{1,\kappa}} - \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} \sum_{j_1+1}^n k_{x_{1,j}} - \delta_{j_1} k_{x_{1,j_1}} \gamma_o x_{j_1} \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} \right] \right. \\ & + \Delta \bar{h}_{D_1} k_{x_{1,1}} \left[\delta_\kappa - (1 - \delta_{j_1}) x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{1,\kappa}} + \right. \\ & \left. \left. \delta_k \frac{\gamma_o \alpha_w}{2 \bar{\gamma}_1} \left[\delta_{j_1} + (1 - \delta_{j_1}) (1 - x_{j_1}^2) \right] \right] + \right. \\ & \left. k_{x_{1,1}} (W_{1,1} - W_{w_1}) \frac{\delta_\kappa \alpha_1}{\bar{\gamma}_1} \right\} \end{aligned} \quad (A.3.7)$$

where x_{j_1} is the fraction defined in App. A.1. The general partial differentiation of H_{w_D} is:

$$\frac{\partial H_{w_D}}{\partial W_{1,\kappa}} = \frac{\frac{\theta k_{x_{1,\kappa}}}{k_h \ln(\bar{r}_1)} \left[F_1(j_1, x_{j_1}) + (W_{1,1} - W_{w_1}) \frac{\delta_\kappa \alpha_1}{\bar{\gamma}_1} \right]}{-\frac{q_D B_{w_b} \gamma_o c_{L_D}}{4 + \gamma_o (H_{w_D} + H_{w_D}^k) c_{L_D}} + \frac{1}{2 \phi \Delta t_D} + \frac{\theta \gamma_o}{k_h \ln(\bar{r}_1)} F_2(j_1, x_{j_1}, \bar{k}_1)}$$

(A.3.8)

where:

$$F_1(j_1, x_{j_1}) = (1 - \delta_\kappa) \Delta \bar{z}_{D_1} + \delta_\kappa \bar{\Delta h}_{D_1} \left[1 - \frac{\gamma_o \alpha_w}{2 \bar{\gamma}_1} \left[\delta_{j_1} + (1 - \delta_{j_1}) (1 - x_{j_1}^2) \right] \right] \quad (A.3.9)$$

and:

$$F_2(j_1, x_{j_1}, \bar{k}_1) = \Delta \bar{z}_{D_1} \left[\sum_{j_1+1}^n k_{x_{1,j}} + \delta_{j_1} k_{x_{1,j_1}} x_{j_1} \right] + \bar{\Delta h}_{D_1} k_{x_{1,1}} (1 - \delta_{j_1}) x_{j_1} \quad (A.3.10)$$

Next, differentiate the same expression with respect to $W_{2,1}$:

$$\begin{aligned} q_D \frac{\partial B_{w_b}}{\partial W_{2,1}} &= - \frac{1}{2 \phi \Delta t_D} \frac{\partial H_{w_D}}{\partial W_{2,1}} - \frac{\theta}{k_h \ln(\bar{r}_1)} \times \\ &\left\{ \bar{\Delta z}_{D_1} \left[\sum_{j_1+1}^n k_{x_{1,j}} \frac{\partial W_{w_1}}{\partial W_{2,1}} + \delta_{j_1} k_{x_{1,j_1}} \frac{\partial W_{w_{j_1}}}{\partial W_{2,1}} \right] + \bar{\Delta h}_{D_1} k_{x_{1,1}} \frac{\partial W_{w_1}}{\partial W_{2,1}} \right. \\ &\left. - k_{x_{1,1}} (W_{1,1} - W_{w_1}) \frac{\partial \bar{\Delta h}_{D_1}}{\partial W_{2,1}} \right\} \end{aligned} \quad (A.3.11)$$

Also:

$$\frac{\partial \bar{\Delta h}_{D_1}}{\partial W_{2,1}} = (1 - \alpha_1) \frac{\partial \Delta h_2}{\partial W_{2,1}} = \frac{1 - \alpha_1}{\bar{\gamma}_2} = \frac{1 - \alpha_w}{2 \bar{\gamma}_2} \quad (A.3.12)$$

$$\frac{\partial W_{w_j}}{\partial W_{2,1}} = \gamma_o \frac{\partial H_{w_D}}{\partial W_{2,1}} \Rightarrow j < j_1 \quad (A.3.13)$$

$$\frac{\partial W_{w_{j_1}}}{\partial W_{2,1}} = x_{j_1} \gamma_o \frac{\partial H_{w_D}}{\partial W_{2,1}} \Rightarrow j = j_1 \neq 1 \quad (A.3.14)$$

$$\frac{\partial W_{w_{j_1}}}{\partial W_{2,1}} = \gamma_o \left\{ (1 - \delta_{j_1}) x_{j_1} \frac{\partial H_{w_D}}{\partial W_{2,1}} + \frac{1 - \alpha_w}{2 \bar{\gamma}_2} \left[(1 - \delta_{j_1}) (1 - x_{j_1}^2) + \delta_{j_1} \right] \right\}$$

$$\Rightarrow j = j_1 = 1 \quad (A.3.15)$$

Substituting the partial differentials and considering alternative positions for the liquid level in the wellbore by taking j_1 :

$$\frac{-q_D B_{w_b} \gamma_o c_{L_D}}{4 + \gamma_o (H_{w_D} + H_{w_D}^k) c_{L_D}} \frac{\partial H_{w_D}}{\partial W_{2,1}} = -\frac{1}{2 \phi \Delta t_D} \frac{\partial H_{w_D}}{\partial W_{2,1}} - \frac{\theta}{k_h \ln(\bar{r}_1)} \times$$

$$\left\{ \bar{\Delta z}_{D_1} \left[\sum_{j_1+1}^n k_{x_{1,j}} + \delta_{j_1} x_{j_1} k_{x_{1,j_1}} \right] \gamma_o \frac{\partial H_{w_D}}{\partial W_{2,1}} \right.$$

$$+ \gamma_o \bar{\Delta h}_{D_1} k_{x_{1,1}} \left[(1 - \delta_{j_1}) \left(x_{j_1} \frac{\partial H_{w_D}}{\partial W_{2,1}} + (1 - x_{j_1}^2) \frac{(1 - \alpha_w)}{2 \bar{\gamma}_2} \right) + \delta_{j_1} \frac{(1 - \alpha_w)}{2 \bar{\gamma}_2} \right]$$

$$\left. - k_{x_{1,1}} (W_{1,1} - W_{w_1}) \frac{1 - \alpha_w}{2 \bar{\gamma}_2} \right\} \quad (A.3.16)$$

After rearrangement:

$$\frac{\partial H_{w_D}}{\partial W_{2,1}} = \frac{-\frac{\theta \gamma_o k_{x_{1,1}}}{k_h \ln(\bar{r}_1)} \frac{(1 - \alpha_w)}{2 \bar{\gamma}_2} \left\{ \bar{\Delta h}_{D_1} \left[(1 - \delta_{j_1}) (1 - x_{j_1}^2) + \delta_{j_1} \right] - \frac{W_{1,1} - W_{w_1}}{\gamma_o} \right\}}{-\frac{q_D B_{w_b} \gamma_o c_{L_D}}{4 + \gamma_o (H_{w_D} + H_{w_D}^k) c_{L_D}} + \frac{1}{2 \phi \Delta t_D} + \frac{\theta \gamma_o}{k_h \ln(\bar{r}_1)} F_2(j_1, x_{j_1}, \bar{k}_1)}$$

$$\quad (A.3.17)$$

Appendix B. Groundwater and Petroleum Engineering Units

Although dealing with similar properties of fluids and rocks and the same mathematical description of the physical phenomena, the hydrology and petroleum engineering literatures use different symbols. As an example, the groundwater transmissivity does not have the same units as the *transmissivity* in petroleum engineering. A good reference to have in mind regarding symbols is App. A in *Earlougher* (1977).

A short conversion set follows to help understanding and comparisons of units with those in the groundwater references used in this dissertation. Most of the petroleum engineering parameters can be directly measured in laboratories or in situ. Conversely, the hydrologic terminology frequently uses lumped properties related to the diffusivity equation. Thus, terms like *storage coefficient*, *specific yield*, *specific storage*, *transmissivity*, *coefficient of permeability*, etc. are not familiar to many petroleum engineers.

The following partial differential equation is used in hydrology to represent the *r-z* flow of liquid (water) in a porous medium in a cylindrical model:

$$K_r \frac{\partial^2 s}{\partial r^2} + \frac{K_r}{r} \frac{\partial s}{\partial r} + K_z \frac{\partial^2 s}{\partial z^2} = S_s \frac{\partial s}{\partial t} \quad (B.1)$$

where:

K_r, K_z - coefficient of permeability in both r and z directions, LT^{-1}

s - head drawdown, L

S_s - specific (elastic) storage, L^{-1}

t - time, T

r, z - radial and vertical coordinates, L

The same Eq. B.1 in petroleum engineering symbols is:

$$\frac{k_r}{\mu} \frac{\partial^2 p}{\partial r^2} + \frac{k_r}{\mu} \frac{\partial p}{\partial r} + \frac{k_z}{\mu} \frac{\partial^2 p}{\partial z^2} = \phi c_t \frac{\partial p}{\partial t} \quad (B.2)$$

where:

- k_r, k_z - permeability in r and z directions, L^2
- p - pressure, $\text{L}^{-1}\text{MT}^{-2}$
- μ - viscosity, $\text{L}^{-1}\text{MT}^{-1}$
- ϕ - porosity, fraction
- c_t - total system compressibility, LM^{-1}T^2
- t - time, T
- r, z - radial and vertical coordinates, L

Correspondence may be established by a dimensional analysis. Usual conversion of units into field units follows for a water saturated system ($\mu = 1 \text{ cp}$):

Coefficient of permeability and permeability:

$$K = \frac{k}{\mu} \left(\rho \frac{g}{g_c} \right) \quad (\text{B.3})$$

Field units:

$$(K, \text{ gal}/D/\text{ft}^2) = 0.047339 \frac{(k, \text{ mD})}{(\mu, \text{ cp})} \left(\rho \frac{g}{g_c}, \text{ psi}/\text{ft} \right) \quad (\text{B.4})$$

Specific storage and diffusivity:

$$S_s = \phi c_t \left(\rho \frac{g}{g_c} \right) \quad (\text{B.5})$$

Field units:

$$(S_s, \text{ ft}^{-1}) = (\phi) (c_t, \text{ psi}^{-1}) \left(\rho \frac{g}{g_c}, \text{ psi}/\text{ft} \right) \quad (\text{B.6})$$

Head drawdown and pressure drawdown:

$$s = \frac{p_o - p}{\rho \frac{g}{g_c}} \quad (B.7)$$

Field units:

$$s, ft = \frac{(p_o - p, \text{ psi})}{\left(\rho \frac{g}{g_c}, \text{ psi/ft}\right)} \quad (B.8)$$

There are other important variables. Storage coefficient, transmissivity and specific yield are:

Storage coefficient, S (dimensionless):

$$S = S_s h = \phi c_t h \left(\rho \frac{g}{g_c} \right) = \phi c_t p \quad (B.9)$$

where h is the thickness of the formation in consistent units. In the unconfined flow theory, h is the liquid height, and the pressure p may replace $h(\rho g/g_c)$. Some theories of unconfined aquifers consider a constant S_s , neglecting the unsaturated flow (Neuman (1972), Streltsova (1972), and others). In this case, h should be the original liquid level at rest.

Transmissivity ($L^2 T^{-1}$)

$$T = K_r h \quad (B.10)$$

Field units:

$$(T, \text{ gal/D/ft}) = 0.047339 \frac{(k, \text{ mD})(h, \text{ ft})}{(\mu, \text{ cp})} \left(\rho \frac{g}{g_c}, \text{ psi/ft} \right) \quad (B.11)$$

From the unconfined flow theory

Specific yield, S_y (dimensionless):

This variable was introduced in the free surface boundary equation (see Neuman, (1972), (1974), among others):

$$K_r \frac{\partial s}{\partial r} n_r + K_z \frac{\partial s}{\partial z} n_z = \left(S_y \frac{\partial \xi}{\partial t} - I \right) n_z, \quad \text{at} \quad z = \xi$$

(B.12)

where:

- n_r, n_z - component of unit normal in r and z directions
- ξ - free surface elevation from the aquifer bottom, L
- I - recharge source rate at the free surface, LT^{-1}

Equation B.12 may be related to the free surface boundary condition in the Chapter 3. S_y is related to the effective porosity:

$$S_y = \phi (1 - S_{w_r}) \quad (B.13)$$

where S_{w_r} is the residual water saturation in the unsaturated region. The term I may be considered a net flow liquid left behind by the water table moving boundary as an effect of capillarity. Generally, S_y has been considered constant and I takes care of the recharge, but in the *Neuman* (1972, 1974) studies the recharge I was neglected.

Usual dimensionless parameters

Dimensionless permeability:

$$K_D = \frac{(K_z, \text{ gal/D/ft}^2)}{(K_r, \text{ gal/D/ft}^2)} = \frac{(k_z, \text{ mD})}{(k_r \text{ mD})} \quad (B.14)$$

Dimensionless drawdown, constant producing flow rate - see *Neuman* (1972):

$$\begin{aligned} s_D &= \frac{4\pi (T, \text{ gal/D/ft})}{(Q, \text{ gal/D})} (s, \text{ ft}) \\ &= 0.014164 \frac{(k, \text{ mD})(h_o, \text{ ft})}{(\mu, \text{ cp})(Q, \text{ bbl/D})} (p, \text{ psi}) \end{aligned} \quad (B.15)$$

Dimensionless time with respect to the specific storage S :

$$\begin{aligned}
t_s &= 5.57 \times 10^{-3} \frac{(T, \text{ gal/D/ft})}{S (r^2, \text{ ft}^2)} (t, \text{ hr}) \\
&= 2.637 \times 10^{-4} \frac{(k_r, \text{ mD}) (t, \text{ hr})}{\phi(\mu, \text{ cp}) (c_t, \text{ psi}^{-1}) (r^2, \text{ ft}^2)} \quad (\text{B.16})
\end{aligned}$$

Dimensionless time with respect to the specific yield S_y :

$$\begin{aligned}
t_y &= 5.57 \times 10^{-3} \frac{(T, \text{ gal/D/ft})}{S_y (r^2, \text{ ft}^2)} (t, \text{ hr}) \\
&= 2.637 \times 10^{-4} \frac{(k_r, \text{ mD}) (h_o, \text{ ft}) (t, \text{ hr})}{\phi(1 - S_{w_r}) (\mu, \text{ cp}) (r^2, \text{ ft}^2)} \quad (\text{B.17})
\end{aligned}$$

β -parameter:

$$\beta = K_D \frac{(r^2, \text{ ft}^2)}{(h_o^2, \text{ ft}^2)} = \frac{(k_z, \text{ mD})}{(k_r, \text{ mD})} \frac{(r^2, \text{ ft}^2)}{(h_o^2, \text{ ft}^2)} \quad (\text{B.18})$$

σ -parameter:

$$\sigma = \frac{S}{S_y} = \frac{(c_t, \text{ psi}^{-1}) (p_o, \text{ psi})}{(1 - S_{w_r})} \quad (\text{B.19})$$

Appendix C. Simplified Approaches

The analytical approximations of *Theis* (1935) and *Ramey et al.* (1989) used to give late time approaches for the gravity well problem do not consider the vertical velocity component in the reservoir, as well as wellbore storage effects. A need to incorporate wellbore storage effects in those solutions required additional formulation, and two computer programs were developed. These improvements in the approaches were not intended to reproduce the unconfined flow behavior at early times, but only to reproduce a better transient flow behavior, mainly when skin effects were considered. In this appendix, a brief description of each program is presented.

The *Theis* (1935) method has been used by hydrologists as a late time reference solution for the unconfined flow of liquid in a gravity drainage well. The derivation of the *Theis* solution did not consider either a skin effect nor a wellbore storage effect in the inner boundary condition. *Theis* used a partial differential equation developed for heat transfer to approximate the unconfined flow problem, but his solution is mainly applicable to confined compressible flow. To incorporate wellbore storage and skin effects, a computer program called *PDE* was developed. Details of this approach are described in App. C.1.

Recently, *Ramey et al.* (1989) developed an approximate method in which the flow was proportional to the column of liquid, and considered skin effects. The transient p^2 solution of the gravity problem proposed by *Ramey et al.* (1989) was an improvement to the *Theis* (1935) solution, because the variable height of the liquid column in the reservoir was considered in a partial differential equation. As a result, a nonlinear Forchheimer-type equation in dimensionless form was obtained similar to the nonlinear partial differential equation in the ideal gas problem studied by *Jenkins and Aronofsky* (1953). *Jenkins and Aronofsky* correlated their numerical solutions with the *van Everdingen and Hurst* (1949) liquid solutions to conclude that the later could be reasonably applied to solve for the transient pressure at the wellbore, if an appropriate dimensionless pressure (in terms of p^2) were used. Using a late time logarithmic approximation, *Ramey et al.* included the *van Everdingen and Hurst* skin effect, but not wellbore storage effects. To consider storage effects in the wellbore, a numerical model for ideal and real gas radial flow developed by *Fligelman* (1980) was adapted to the liquid gravity drainage well problem, since the nonlinear partial differential equations for ideal gas flow and the Dupuit-Forchheimer approach are exactly the same when appropriate dimensionless parameters are used (see App C.2). This program was named *FLIGRAM*.

C.1 Wellbore Effects in the Theis Solution

Appropriate boundary conditions were applied to the diffusivity equation for wellbore storage and skin effects for an infinitely large reservoir. Some results were compared to the *Theis* (1935) line source solution.

The basic diffusivity equation for radial flow in Darcy units is:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} = - \frac{\phi \mu c_t}{k} \frac{\partial p}{\partial t}. \quad (C.1.1)$$

Theis applied Eq. C.1.1 for a constant rate well to obtain the pressure response at any radial distance r and time t , for both a confined and an unconfined aquifer. Wellbore effects were not considered. Based on the simplified assumption of a line-sink, the *Theis* approach had no valid response close to the wellbore, where the storage effects are important. Eventually, at late time, the pressure response approaches that of a confined reservoir, and the line source solution becomes accurate enough. In terms of petroleum engineering nomenclature, the *Theis* approach was expressed by:

$$p(r, t) = \frac{q B \mu}{2\pi k h} p_D(r_D, t_D), \quad (C.1.2)$$

where:

$$p_D(r_D, t_D) = - \frac{1}{2} E_i \left(- \frac{\phi \mu c_t r^2}{4 k t} \right), \quad (C.1.3)$$

For a small argument, the exponential integral function $E_i(-x)$ may be approximated by:

$$\begin{aligned} E_i(-x) &= - \int_x^\infty \frac{e^{-u}}{u} du = \left[\ln x - \frac{x}{1!} + \frac{x^2}{2 \times 2!} - \frac{x^3}{3 \times 3!} + \dots \right]_x^\infty \\ &\approx \frac{1}{2} \left[\ln \left(\frac{k t}{\phi \mu c_t r^2} \right) + 0.80907 \right]. \end{aligned} \quad (C.1.4)$$

Wellbore storage and skin effects were added to the solution of the partial differential equation, Eq. C.1.1. The inner boundary condition was:

$$q = C \frac{\partial p_w}{\partial t} - \frac{2\pi k h}{\mu} \left(r \frac{\partial p}{\partial r} \right)_{r=r_w} \quad (C.1.5)$$

and:

$$p_w = \left[p - s \left(r \frac{\partial p}{\partial r} \right) \right]_{r=r_w}, \quad (C.1.6)$$

where C is the wellbore storage parameter. The initial condition was:

$$p(r, 0) = p_i. \quad (C.1.7)$$

As an infinite reservoir, the pressure approaches the initial value, as the distance from the wellbore center becomes large:

$$p(r, t) \underset{r \rightarrow \infty}{=} p_i. \quad (C.1.8)$$

It was convenient to introduce dimensionless variables as defined below. Both Darcy and field units are given.

Darcy units:

$$r_D = \frac{r}{r_w} \quad r_D = \frac{r}{r_w} \quad (C.1.9)$$

$$t_D = \frac{k t}{\phi \mu c_t r_w^2} \quad t_D = \frac{(0.0002637) k t}{\phi \mu c_t r_w^2} \quad (C.1.10)$$

$$p_D = \frac{2\pi k h (p_i - p)}{q \mu} \quad p_D = \frac{k h (p_i - p)}{141.2 q B \mu} \quad (C.1.11)$$

$$C_D = \frac{1}{2 p_i c_t} \quad C_D = \frac{1}{2 p_i c_t} \quad (C.1.12)$$

Substituting the dimensionless parameters into the diffusivity equation:

$$\frac{\partial^2 p_D}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial p_D}{\partial r_D} = \frac{\partial p_D}{\partial t_D}. \quad (C.1.13)$$

Initial Condition:

$$p_D(r_D, 0) = 0. \quad (C.1.14)$$

Outer Boundary Condition:

$$\lim_{r_D \rightarrow \infty} p_D(r_D, t_D) = 0. \quad (C.1.15)$$

Combined Wellbore and Sandface Conditions:

$$1 = C_D \left[\frac{\partial p_D}{\partial t_D} - \varepsilon r_D \frac{\partial}{\partial t_D} \left(\frac{\partial p_D}{\partial r_D} \right) \right]_{r_D=1+} - \left(r_D \frac{\partial p_D}{\partial r_D} \right)_{r_D=1+} \quad (C.1.16)$$

The solution for the dimensionless pressure in the wellbore in Laplace space is:

$$\bar{p}_{wD} = \frac{K_0(\sqrt{z}) + s\sqrt{z}K_1(\sqrt{z})}{z^{3/2} [K_1(\sqrt{z}) + C_D\sqrt{z}K_0(\sqrt{z}) + zC_DsK_1(\sqrt{z})]} \quad (C.1.17)$$

A computer program was written to calculate the wellbore pressures using Eq. C.1.17. Inversion from the Laplace space was performed numerically using the *Stehfest* (1970) algorithm. Early and late time approximate analytical solutions of Eq. C.1.17 were also derived. An early time approximation ($z \rightarrow \infty$) inverted to real space after simplification of the Bessel functions produced:

$$p_{wD} = \frac{t_D}{C_D} \quad (C.1.18)$$

A late time approximation was similar to that expressed by Eq. C.1.4, including the skin parameter:

$$p_{wD} = \frac{1}{2} [\ln(t_D) + 0.80907 + 2s]. \quad (C.1.19)$$

Figures C.1 and C.2 show the results in both semilog and log-log plots of the solutions from the computer program PDE and the line source solution. Also, the buildup behavior is shown in Fig. C.3. A perfect match between the solutions when the wellbore effects were no longer important verify the results from the program PDE.

C.2 Wellbore Effects in the p^2 Solution

Wellbore storage effects were considered in the inner boundary condition for a constant production rate well in an infinitely large reservoir. The wellbore storage coefficient was coupled with a partial differential equation (Eq. C.2.1 in *Ramey et al.*, 1989) for an approximate late time solution of a gravity drainage well. A resulting system of nonlinear equations similar to the ideal gas problem solved numerically by *Fligelman* (1980) was adapted to a single phase liquid with a free surface. Transmissivity was taken to be proportional to the liquid column in the reservoir, and vertical velocities were neglected.

Ramey et al. (1989) reintroduced the p^2 approach for the free surface gravity drainage, rewriting the *Dupuit-Forchheimer* partial differential equation in terms of a variable height $h(r, t)$ converted into pressure. The partial differential equation presented a pressure-factor in the diffusivity term, obtained for an incompressible system. Equation 1.2 in the Introduction Ch. 1, presented below as Eq. C.2.1, shows the proposed equation in terms of pressure. This is the same as the ideal gas equation studied by *Jenkins* and *Aronofsky* (1953). Notice the pressure-dependent diffusivity coefficient.

$$\frac{\partial^2 p^2}{\partial r^2} + \frac{1}{r} \frac{\partial p^2}{\partial r} = \frac{\phi \mu (p^{-1})}{k} \frac{\partial p^2}{\partial t}. \quad (\text{C.2.1})$$

The fundamental assumption to obtain Eq. C.2.1 was that at given a time t and a radial position r the liquid pressure at the lower boundary was assumed to be proportional to the column of liquid. The assumption of incompressible flow requires that pressure is proportional to liquid height:

$$p(r, t) = h \rho \frac{g}{g_c}. \quad (\text{C.2.2})$$

Jenkins and *Aronofsky* correlated the numerical solution of Eq. C.2.1 with the *van Everdingen and Hurst* (1949) solutions of the linear diffusivity problem. *Ramey et al.* adapted the *Jenkins* and *Aronofsky* correlations for a constant rate wellbore condition to write an approximate analytical solution, not valid at early times:

$$p_i^2 - p_w^2 = \frac{2.303}{4\pi} \frac{q B \mu}{k \rho \frac{g}{g_c}} \left\{ \log_{10} \left(\frac{k t}{\phi \mu (p_i^{-1}) r_w^2} \right) + 0.80907 + 2s \right\}. \quad (\text{C.2.3})$$

To consider wellbore storage effects, and to allow the *Fligelman* (1980) program

to handle pressure buildup behavior as well as drawdown, a new simplified program was written and named *FLIGRAM*. The mathematical procedure is described next.

First, a gravity drainage pseudo-pressure was defined:

$$m_p = \frac{k}{\mu} p^2. \quad (C.2.4)$$

The inner and outer boundary conditions were the same as those described in App. C.1, as well as the initial condition. The dimensionless parameters were:

Darcy units:

Field units:

$$r_D = \frac{r}{r_w} \quad r_D = \frac{r}{r_w} \quad (C.2.5)$$

$$t_D = \frac{k p_i t}{\phi \mu r_w^2} \quad t_D = \frac{(0.0002637) k p_i t}{\phi \mu r_w^2} \quad (C.2.6)$$

$$m_D = \frac{m_{p_i} - m_p}{m_{p_i}} = \frac{p_i^2 - p^2}{p_i^2} \quad m_D = \frac{m_{p_i} - m_p}{m_{p_i}} = \frac{p_i^2 - p^2}{p_i^2} \quad (C.2.7)$$

$$C_D = \frac{1}{2\phi} \quad C_D = \frac{1}{2\phi} \quad (C.2.8)$$

$$\alpha_D = \frac{h_o}{h} \quad \alpha_D = \frac{h_o}{h} \quad (C.2.9)$$

$$q_D = \frac{q B \rho \frac{q}{g_c}}{\pi m_{p_i}} \quad q_D = \frac{282.41 q B \rho \frac{q}{g_c}}{m_{p_i}} \quad (C.2.10)$$

Substituting the dimensionless parameters into the diffusivity equation (Eq. C.2.1) and into the boundary conditions:

$$\frac{\partial^2 m_D}{\partial r_D^2} + \frac{1}{r_D} \frac{\partial m_D}{\partial r_D} = \alpha_D \frac{\partial m_D}{\partial t_D}. \quad (C.2.11)$$

Initial Condition:

$$m_D(0, r_D) = 0. \quad (C.2.12)$$

Outer Boundary Condition:

$$m_D(t_D, r_D) = 0. \quad (C.2.13)$$

Wellbore Material Balance:

$$q_D = C_D \alpha_D \frac{dm_{D1}}{dt_D} \Big|_{r_D=1+} - \frac{k_1}{k} r_D \frac{dm_{D1}}{dr_D} \Big|_{r_D=1+}, \quad (C.2.14)$$

where the subscript 1 stands for the damaged region around the wellbore. The skin effect was translated into an equivalent permeability using *Hawkins* (1956) method (see Eq. 4.1).

Results from the *FLIGRAM* program are presented next and verified with the late time *Ramey et al.* approach. Figures C.4 through C.6 are graphs of drawdown pressure vs. time for a typical set of data. Two skin values, 0 and 5 respectively, were used. At late times, when wellbore effects have no significant influence on the pressure response, the numerical method matches the analytical approach. In a semilog graph in Fig. C.4, the late time behavior shows an approximate linear relationship between pressure drawdown and the logarithm of time. Fig. C.5 is a log-log graph where the wellbore storage effects can be clearly identified by the unit slope straight line at very early times. In the p-squared vs. time semilog graph in Fig. C.6, the slope of the logarithmic behavior period calculated from Eq. 10 in the *Ramey et al.* (1989) paper, 132.1 psi²/cycle, was matched.

Using the given example for a test, we calculated the m-permeability using an approximate late time semilog straight line obtained in Fig. C.4, and compared with the input data of 240 mD. The semilog straight line slope was 0.89 psi/cycle, corresponding to a permeability of 217 mD:

$$\text{Radial Flow: } k = 162.6 \frac{q B \mu}{m h_o} = (162.6) \frac{(225)(1)(1)}{(0.89)(189)} = 217.4 \text{ mD}$$

This rough solution is less than 10% lower than the p-squared solution, for the given example.

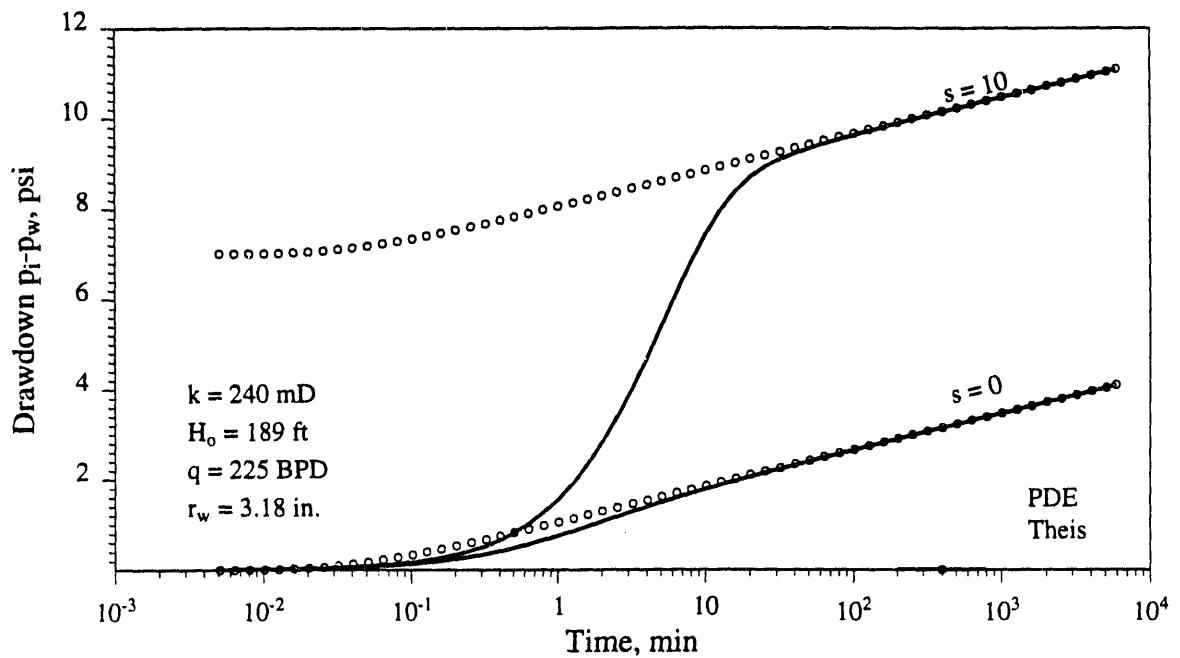


Figure C.1: PDE and analytical Theis (1935) solution. Semilog plot.

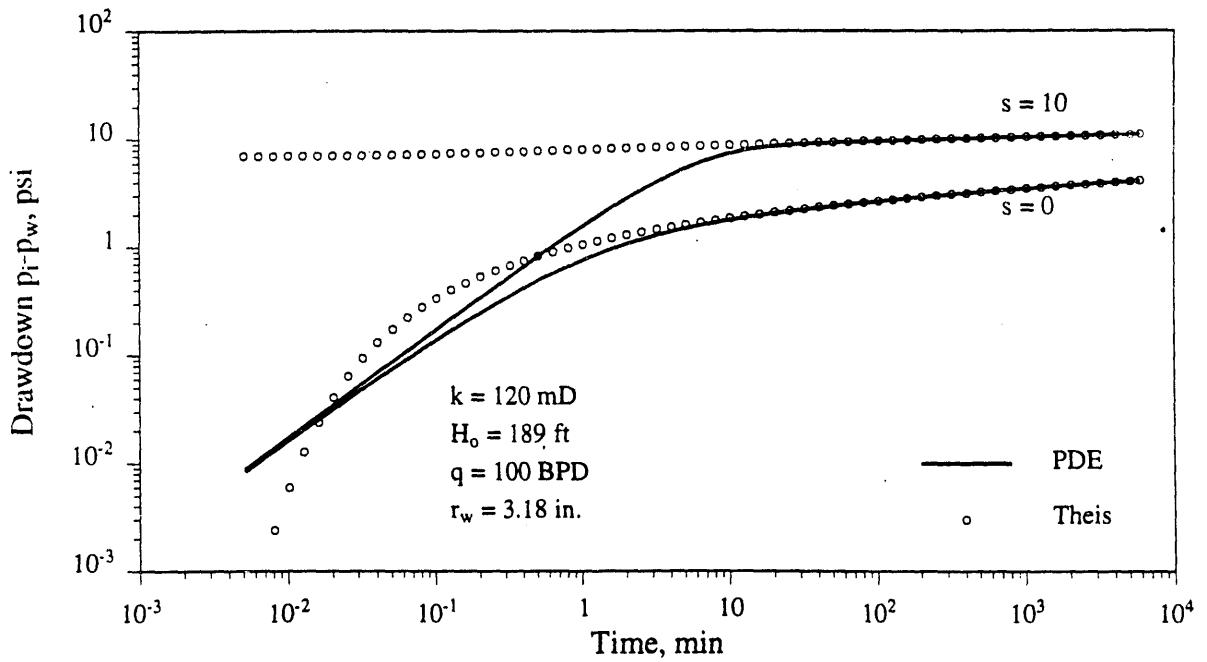


Figure C.2: PDE and the analytical Theis (1935) solution. Log-log plot.

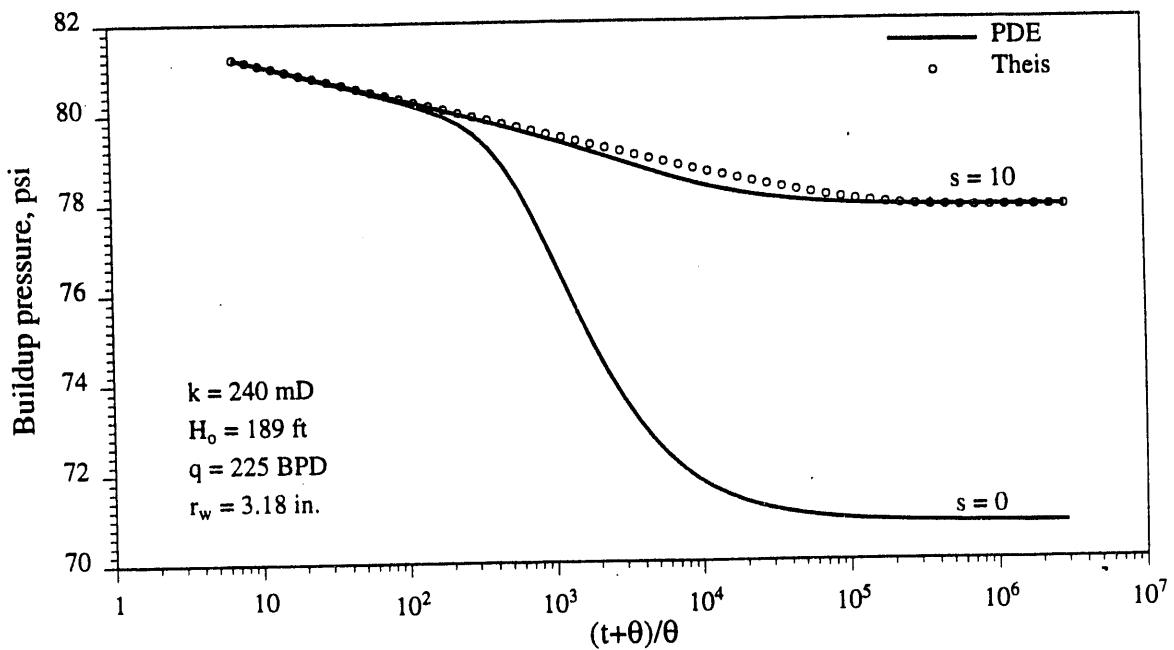


Figure C.3: *PDE* and analytical *Theis* (1935) solution. Buildup pressure semilog plot.

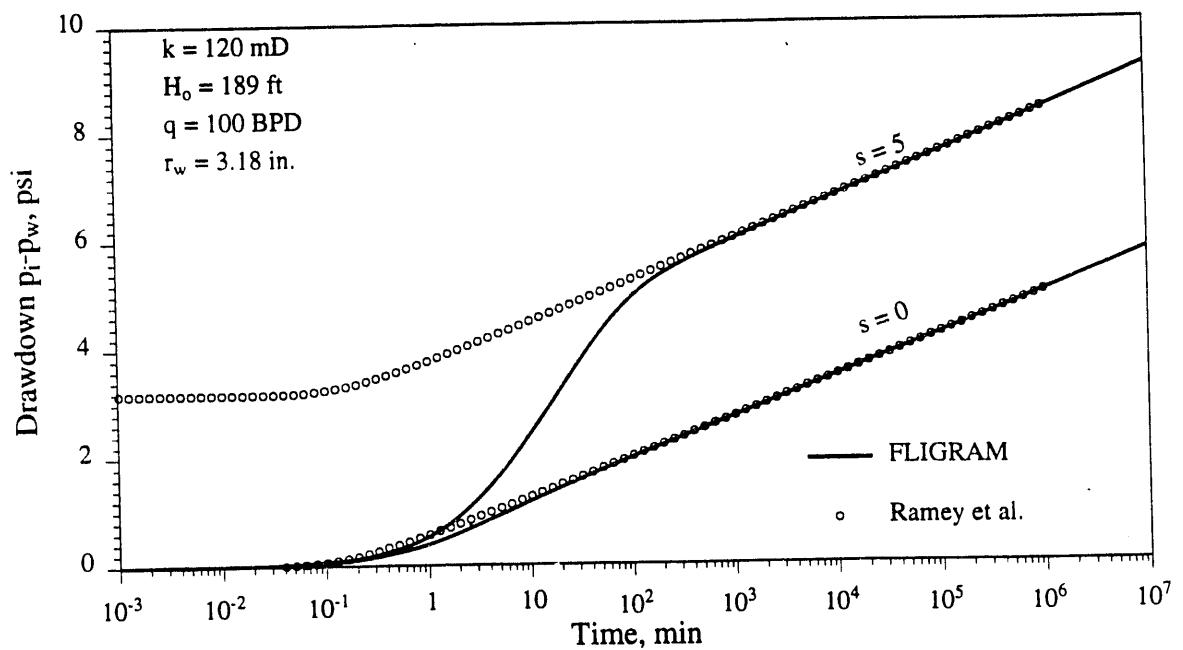


Figure C.4: *FLIGRAM* and *Ramey et al.* approaches for the drawdown period, semilog plot.

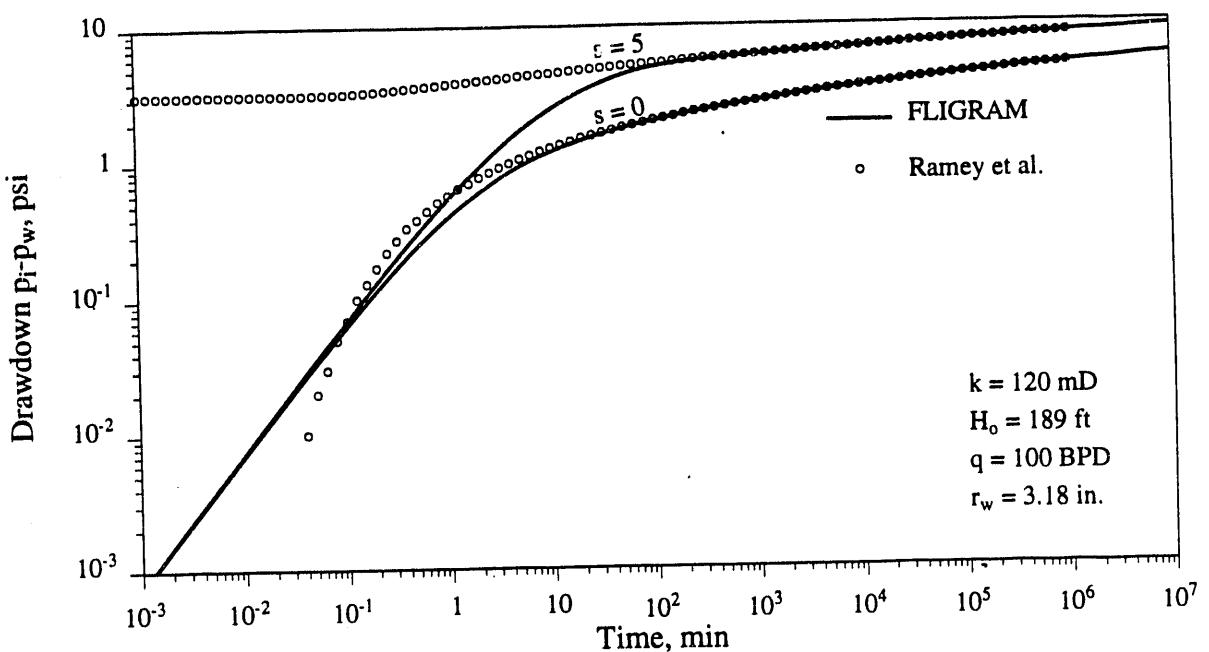


Figure C.5: *FLIGRAM* and *Ramey et al.* approaches for the drawdown period, log-log graph.

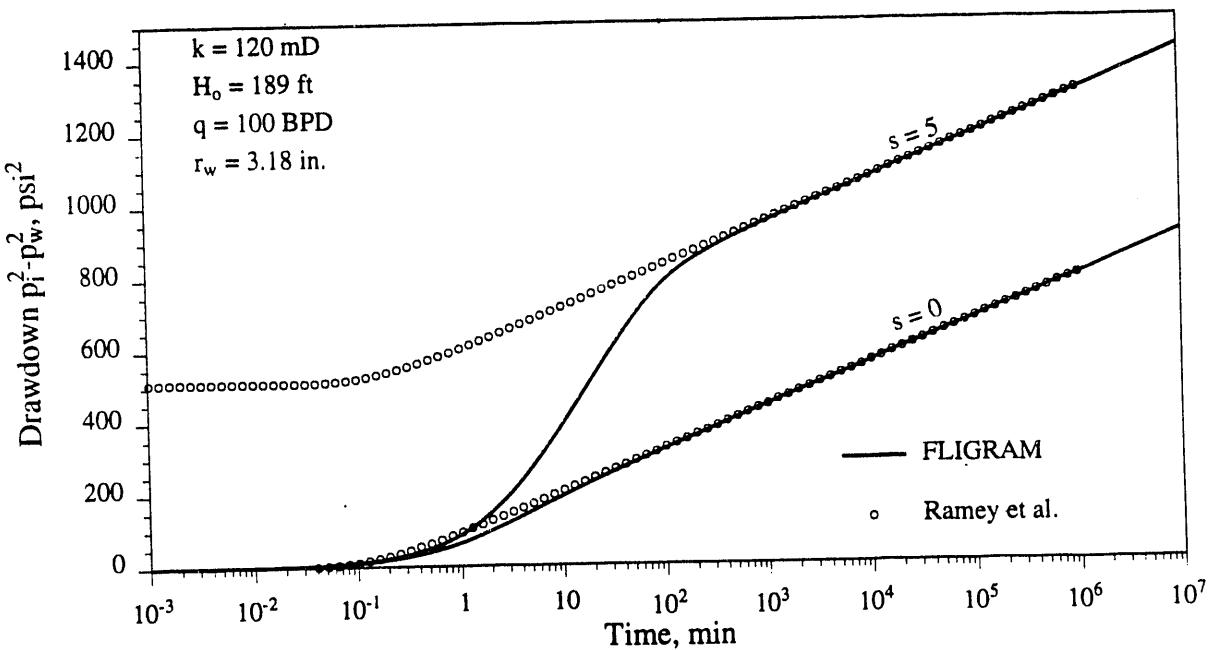
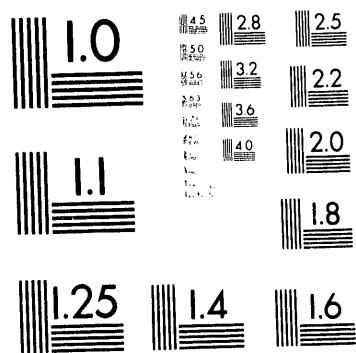


Figure C.6: *FLIGRAM* and *Ramey et al.* approaches for the drawdown period, p -squared semilog graph.



3 of 3

Appendix D. Verification of the Free Surface Boundary Condition Relationship with the Average Vertical Velocity

An interesting way of representing a relationship between the free surface position and the vertical velocity can be found in the *Wyckoff et al.* (1932) work. Using the symbology employed in the present study:

$$h(r) = \frac{\Phi_b(r)}{\rho \frac{g}{g_c}} + \frac{\bar{v}_v(r) \mu}{k_v \rho \frac{g}{g_c}} h(r), \quad (D.1)$$

where $h(r)$ is the free surface vertical position at a radial location r from the well center, $\Phi_b(r)$ is the potential in the base of the reservoir at the same r , $v_v(r)$ is the average vertical velocity of the liquid, μ is the liquid viscosity, k_v is the vertical permeability, and $\rho g/g_c$ is the liquid pressure gradient. If we define v_v as:

$$\bar{v}_v(r) = \frac{1}{h(r)} \int_0^{h(r)} \frac{k_v}{\mu} \frac{\partial \Phi}{\partial z} \Big|_r dz, \quad (D.2)$$

the integration produces:

$$\bar{v}_v(r) = \frac{k_v}{\mu} \frac{\Phi_{f_s}(r) - \Phi_b(r)}{h(r)} \quad (D.3)$$

where f_s refers to the free surface vertical position. Direct substitution of Eq. D.3 vertical velocity v_v into the *Wyckoff et al.* expression gives:

$$h(r) = \frac{\Phi_{f_s}(r)}{\rho \frac{g}{g_c}} \quad (D.4)$$

which is the free surface boundary condition in the current study.

Appendix E. SLM Computer Program

In this Appendix, the SLM computer program is described in detail in Section E.1, and the program code is presented in Section E.5. The computer program was written in double precision FORTRAN. An exception was the subroutine that calculates the liquid level in the wellbore which gives an appropriate accuracy using extended precision on IBM machines. In that subroutine, when calculating the production flow rate for a constant-pressure inner boundary condition, a quadratic equation computes differences of squared large numbers and divisions which require a large number of significant digits in the operations. In Section E.3, an example of an input data file is given. Example of output file is given in Section E.4.

E.1 Structure of the Program

The simulator is composed of a main program, 30 subroutines and 5 functions. A flow chart is presented next to aid understanding of the program. Some controls, such as the start of a buildup period or the start of constant flow rate in the well, were omitted in the diagram.

E.2 Description of some Characteristics

In this section, some important characteristics of the computer program are summarized.

E.2.1 Radial Block Distribution

The radial mesh distribution uses a block-centered scheme given by a progressive radial positioning of the block boundaries, according to the *MULTIMODAL* gridding. See *Terán and de la Garza (1988)* for details of the algorithm. The log-spacing is a particular case of the *MULTIMODAL* method. The parameter necessary to define the ratio between the last and the first block radial size, $\frac{\Delta r_{mm}}{\Delta r_1}$, is an input data, as well as a flag integer that provides the selection of the *MULTIMODAL* of the standard logarithmic spacing.

Another important characteristic concerning the computer program is the progressive number of radial blocks considered in the matrix set-up. The remaining blocks, far from the inner boundary are added to the system as the external boundary is disturbed by a small dimensionless potential drop (10^{-9} in the program). After several runs, a number equal the half of the radial dimension (MM/2) was considered a good start.

E.2.2 Axial Permeability

The variable geometry of the problem required a calculation of directional permeability. The permeability tensor for an anisotropic case is defined by an elliptic equation as:

$$k_x = \frac{k_h k_v}{\sqrt{k_h^2 \sin^2 \beta + k_v^2 \cos^2 \beta}} \quad (E.2.1)$$

where β is the angle of the flow axis with the horizontal.

E.2.3 Sandface Block Transmissivity Control

The program contains a parameter $part(j)$ which is an input data for each sandface block. This parameter multiplies each transmissivity term of the inner boundary blocks and can be set from 0 for a no-flow block to 1 for a fully transmissive block. Thus, any block or set of blocks at any position in the wellbore can be partially or totally plugged or partially plugged by this transmissivity control parameter.

E.2.4 Skin Effect

The skin effect was simulated by calculating a modified permeability for the first block. Thus, both vertical and horizontal permeabilities should be affected. However, the method used to determine the skin parameter was the *Hawkins* (1956) approach which deals only with the horizontal permeability:

$$s = \left(\frac{k}{k_d} - 1 \right) \ln \left(\frac{r_d}{r_w} \right) \quad (E.2.2)$$

$$k_d = \frac{1}{1 + s \ln \bar{r}_{D_1}} k_h = f_s k_h \quad (E.2.3)$$

where \bar{r}_{D_1} is the first block radial mesh location.

We must understand this expression. First, Eq. E.2.2 expresses effects in the radial direction only, and there is nothing in the *Hawkins* approach relating to the vertical permeability, an important component of flow near wellbore. In the program, the vertical permeability is not affected by the skin effect defined this way.

Another problem concerns negative skins. Since we the damaged (or stimulated) region was set to be in the first block only, we are limited to a negative skin range defined by the denominator of Eq. E.2.3:

$$1 + s \ln \bar{r}_{D_1} > 0 \quad (E.2.4)$$

$$s > -\frac{1}{\ln \bar{r}_{D_1}}. \quad (E.2.5)$$

E.3 Input-Data File

A sample of the input-data file is presented next. The integer values are right justified at column 64, and the decimal numbers are left justified at column 60. The blank lines must be observed.

ORIGINAL INPUT DATA FILE FOR DRAWDWON AND BUILDUP ANALYSIS IN CHAP. 7:

Number of horizontal grid blocks (M):	80
Number of vertical grid blocks (N):	100
Use MULTINODAL radial spacing? (mult) (no=0, otherw,1)	1
Maximum number of iterations (itermx):	05
Maximum number of matrix calculations (matmax)	6000000
Initial Number of time-step per cycle (ntdi)	10
Var. flowrate indx (iflow: 0=const, 1=cont.var, 2=step.var)	0
Boundary limit dimensionless external radius - (red)	100000.
First and last blck ratio - multimodal (Rmult)	20000.
Liquid compressibility - 1/psi:	0.000003
Total compressibility - 1/psi:	0.000003
Original heighth of the free surface - ft, (ho):	189.
Well bore radius - in, (rw):	3.18
Porosity - fraction, (poros):	0.25
Liquid viscosity - cp, (visc):	1.00
Liquid density times g/gc - lbf/cuft, (rho):	62.4
Volume formation factor - (Bo)	1.
Skin effect parameter - (skin):	0.
Horizontal permeability - mD, (permh):	120.
Vertical permeability - mD, (permv):	120.
Residual Saturation - (srw)	0.
Capillary Fringe thickness - hc, (ft)	0.
Constant Production rate - bbl/D, (q):	100.
Minimum Pressure - psi, (pmin)	0.0
Small number to be used as residue (eps)	0.000001
Time-interpolation parameter (1>=teta>=0)	1.000000
Output control of wellbore dimless. press. (iprnt1)	1
Output control of head profile (iprnt2)	1
Output control of total pressure dist. (iprnt3)	1
Output control: 0=dimless., 1=field units (iprnt4)	1
Initial dimensionless time - (tdi)	1.000000e-04
Buildup time or recovery period - min, (tbuild)	1.000000e+06
Flowing or pumping time - (tflw)	1.000000e+06
Number of tdpr for pres. prof. line below - (ntpr)	8
1. 10. 100. 1000. 10000. 1.e+05 1.e+06 1.e+07	
Number of observation wells - (nobs)	5

Distance from wellbore center, ft (dist(i))
1. 10. 100. 1000. 10000.

Partial penetration factor - part(j)
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.

Flow rate table: time - tq(i) (min) and qvar(i) (bbl/d)

Number of variable flow rates - (nqv)

0

| | |
|-------|------|
| 0. | 214. |
| 1230. | 242. |
| 2640. | 218. |
| 4080. | 191. |
| 5520. | 183. |

E.4 Example of the Program Output

The program has a main output file containing informations for each time step selected for output printing. There are also some output files for specific applications such as the sandface flow rate profile, the radial head distribution, observation wells, and others. The following labels generate corresponding outputs:

Label 2

Normalized head potentials: sand base, vertically averaged, and free surface.

Label 3

Isopotential xy distribution at prescribed times.

Label 11

Main output table: time, wellbore pressure or drawdown, number of iterations in the time step, cumulative number of matrix calculations, position J_1 of the block adjacent to the liquid level in the wellbore, seepage face length, result from *Ramey et al.* (1989) approach, result from *Theis* (1935) approach, cumulative CPU time in seconds.

Label 14

Sandface flow rate profile at prescribed times.

Label 21

Same as **label 11** for observation wells. Each label starting on 21 corresponds to a different locations or observation well.

Any output subroutine may be included by the user. If the user intends to have the same output time step control as in the program, just add the new subroutine call line next to the callings for *PRINTA* subroutine. A frequency of ten output data per logarithmic cycle of the dimensionless time was designed in the program, and controlled by a variable *ntprt*. Another important information for the user is how the information are storaged. At the end of each time step, the following variables are updated. In the next time step, the vectors or scalar parameters are normally substituted by new values in the iterative process. The notation of the variables is the same as that used in the program, regardless capitalization.

| | | |
|----------|---|---|
| H_{wD} | = | wellbore dimensionless liquid level |
| H_s | = | free surface dimensionless height at the sandface |
| $H(i)$ | = | free surface dimensionless height at position i |
| $dh(i)$ | = | free surface block dimensionless thickness |
| $dzd(i)$ | = | block dimensionless thickness |
| $W_w(j)$ | = | normalized sandface potential |
| $W(i,j)$ | = | normalized i,j block potential |

An example of the main output run is presented next.

tmin= .4065E+000x td, Hc= .000ft, Red= 100000., MULTIMODAL, Rmult= .2000E+05
 mm= 80, n=100 Ho=189.000 ft, Skin= .0, Kh= 120., Kv= 120., visc=1.000
 POROS= .25 Ct= .30E-05psi-1, Cl= .30E-05psi-1, Q= 100.0 Swr= .000 Tetia=1.00

.....
 FIELD UNITS:

Time = minutes

Pw = psi

Iter = Number of iterations to complete the printed time-step

Matrix = Cumm. number of matrix calc. during the execution

J1 = Vertical grid position of the wellbore liquid level

Seepage = Seepage face length, ft

Ramey = Well pressure from Ramey et al. (1989) solution, psi

cpu1 = CPU time to calculate the matrix once, sec.

cpu2 = Fraction of time: matrix solver / entire iteration loop

cpu3 = Cummulative CPU time, sec,

| Time | Pw | Iter | Matrix | J1 | Seepage | Ramey | Theis | cpu3 |
|------------|------------|------|---------------|------------|------------|--------|-------|------|
| .4065E-04* | .00003*0* | 1 | 1* .7182E-04* | NaNQ* | NaNQ* | .33 | | |
| .5117E-04* | .00004*0* | 2 | 1* .9042E-04* | NaNQ* | NaNQ* | .65 | | |
| .6442E-04* | .00005*0* | 3 | 1* .1138E-03* | NaNQ* | NaNQ* | .97 | | |
| .8110E-04* | .00006*0* | 4 | 1* .1433E-03* | NaNQ* | NaNQ* | 1.29 | | |
| .1021E-03* | .00008*0* | 5 | 1* .1804E-03* | NaNQ* | NaNQ* | 1.62 | | |
| .1285E-03* | .00010*0* | 6 | 1* .2271E-03* | NaNQ* | NaNQ* | 1.93 | | |
| .1618E-03* | .00012*0* | 7 | 1* .2859E-03* | NaNQ* | NaNQ* | 2.25 | | |
| .2037E-03* | .00016*0* | 8 | 1* .3599E-03* | NaNQ* | NaNQ* | 2.57 | | |
| .2565E-03* | .00020*0* | 9 | 1* .4531E-03* | NaNQ* | NaNQ* | 2.89 | | |
| .3229E-03* | .00025*0* | 10 | 1* .5704E-03* | NaNQ* | NaNQ* | 3.21 | | |
| .4065E-03* | .00031*0* | 11 | 1* .7180E-03* | NaNQ* | NaNQ* | 3.53 | | |
| .5117E-03* | .00039*0* | 22 | 1* .9039E-03* | NaNQ* | NaNQ* | 6.96 | | |
| .6442E-03* | .00049*0* | 33 | 1* .1138E-02* | NaNQ* | NaNQ* | 10.36 | | |
| .8110E-03* | .00062*0* | 44 | 1* .1432E-02* | NaNQ* | NaNQ* | 13.70 | | |
| .1021E-02* | .00078*0* | 55 | 1* .1803E-02* | 00000E+00* | 00000E+00* | 17.03 | | |
| .1285E-02* | .00098*0* | 66 | 1* .2270E-02* | 00000E+00* | 00000E+00* | 20.37 | | |
| .1618E-02* | .00124*0* | 77 | 1* .2857E-02* | 00000E+00* | 00000E+00* | 23.84 | | |
| .2037E-02* | .00156*0* | 88 | 1* .3596E-02* | 00000E+00* | 00000E+00* | 27.32 | | |
| .2565E-02* | .00196*0* | 99 | 1* .4527E-02* | 00000E+00* | 00000E+00* | 30.83 | | |
| .3229E-02* | .00247*0* | 110 | 1* .5698E-02* | 00000E+00* | 00000E+00* | 34.33 | | |
| .4065E-02* | .00311*0* | 121 | 1* .7171E-02* | 00000E+00* | 00000E+00* | 37.75 | | |
| .5117E-02* | .00391*0* | 142 | 1* .9025E-02* | 00000E+00* | 00000E+00* | 44.38 | | |
| .6442E-02* | .00492*0* | 163 | 1* .1136E-01* | 00000E+00* | 00000E+00* | 51.08 | | |
| .8110E-02* | .00619*0* | 184 | 1* .1429E-01* | 00000E+00* | 00000E+00* | 57.80 | | |
| .1021E-01* | .00779*0* | 205 | 1* .1798E-01* | 00000E+00* | 00000E+00* | 64.46 | | |
| .1285E-01* | .00980*1* | 242 | 1* .2263E-01* | 00000E+00* | 00000E+00* | 73.15 | | |
| .1618E-01* | .01233*1* | 284 | 1* .2846E-01* | .7362E-04* | .7362E-04* | 82.56 | | |
| .2037E-01* | .01551*1* | 326 | 1* .3579E-01* | .3673E-03* | .3673E-03* | 92.09 | | |
| .2565E-01* | .01950*1* | 368 | 1* .4501E-01* | .1236E-02* | .1236E-02* | 101.27 | | |
| .3229E-01* | .02452*1* | 410 | 1* .5658E-01* | .3376E-02* | .3375E-02* | 110.58 | | |
| .4065E-01* | .03081*1* | 452 | 1* .7111E-01* | .7761E-02* | .7761E-02* | 119.91 | | |
| .5117E-01* | .03871*1* | 514 | 1* .8932E-01* | .1553E-01* | .1553E-01* | 133.79 | | |
| .6442E-01* | .04860*1* | 576 | 1* .1122E+00* | .2778E-01* | .2778E-01* | 147.74 | | |
| .8110E-01* | .06099*1* | 638 | 1* .1407E+00* | .4538E-01* | .4538E-01* | 161.47 | | |
| .1021E+00* | .07648*1* | 700 | 1* .1765E+00* | .6887E-01* | .6884E-01* | 175.32 | | |
| .1285E+00* | .09581*1* | 762 | 1* .2211E+00* | .9842E-01* | .9836E-01* | 189.21 | | |
| .1618E+00* | .11991*1* | 824 | 1* .2767E+00* | .1339E+00* | .1338E+00* | 203.16 | | |
| .2037E+00* | .14987*1* | 886 | 1* .3458E+00* | .1749E+00* | .1747E+00* | 217.54 | | |
| .2565E+00* | .18701*1* | 948 | 1* .4316E+00* | .2209E+00* | .2206E+00* | 232.10 | | |
| .3229E+00* | .23291*1* | 1010 | 1* .5375E+00* | .2714E+00* | .2710E+00* | 246.51 | | |
| .4065E+00* | .28939*1* | 1072 | 1* .6678E+00* | .3257E+00* | .3251E+00* | 261.03 | | |
| .5117E+00* | .35854*1* | 1154 | 1* .8273E+00* | .3833E+00* | .3824E+00* | 280.22 | | |
| .6442E+00* | .44271*1* | 1236 | 1* .1021E+01* | .4436E+00* | .4424E+00* | 299.45 | | |
| .8110E+00* | .54442*1* | 1318 | 1* .1256E+01* | .5063E+00* | .5047E+00* | 319.25 | | |
| .1021E+01* | .66632*1* | 1400 | 1* .1537E+01* | .5708E+00* | .5688E+00* | 339.10 | | |
| .1285E+01* | .81097*1* | 1482 | 1* .1871E+01* | .6369E+00* | .6344E+00* | 359.65 | | |
| .1618E+01* | .98068*1* | 1564 | 2* .2262E+01* | .7042E+00* | .7012E+00* | 381.06 | | |
| .2037E+01* | .117672*1* | 1646 | 2* .2714E+01* | .7726E+00* | .7689E+00* | 403.26 | | |
| .2565E+01* | .139892*1* | 1728 | 2* .3226E+01* | .8418E+00* | .8375E+00* | 426.21 | | |
| .3229E+01* | .164503*1* | 1810 | 3* .3794E+01* | .9117E+00* | .9067E+00* | 449.10 | | |
| .4065E+01* | .191015*1* | 1892 | 3* .4404E+01* | .9823E+00* | .9764E+00* | 473.17 | | |
| .5117E+01* | .218646*1* | 1994 | 3* .5041E+01* | .1053E+01* | .1046E+01* | 504.13 | | |
| .6442E+01* | .246304*1* | 2036 | 4* .5677E+01* | .1125E+01* | .1117E+01* | 536.26 | | |
| .8110E+01* | .272711*1* | 2198 | 4* .6285E+01* | .1196E+01* | .1188E+01* | 569.27 | | |
| .1021E+02* | .296572*1* | 2300 | 4* .6833E+01* | .1268E+01* | .1258E+01* | 604.10 | | |
| .1285E+02* | 3.16809*1* | 2402 | 4* .7297E+01* | .1340E+01* | .1329E+01* | 639.60 | | |
| .1618E+02* | 3.32808*1* | 2504 | 5* .7663E+01* | .1413E+01* | .1401E+01* | 676.88 | | |
| .2037E+02* | 3.44535*1* | 2606 | 5* .7929E+01* | .1485E+01* | .1472E+01* | 714.77 | | |
| .2565E+02* | 3.52493*1* | 2708 | 5* .8106E+01* | .1558E+01* | .1543E+01* | 752.90 | | |
| .3229E+02* | 3.57402*1* | 2810 | 5* .8213E+01* | .1631E+01* | .1615E+01* | 787.74 | | |
| .4065E+02* | 3.60342*1* | 2912 | 5* .8272E+01* | .1704E+01* | .1686E+01* | 822.62 | | |
| .5117E+02* | 3.61839*2* | 3036 | 5* .8296E+01* | .1777E+01* | .1758E+01* | 860.22 | | |
| .6442E+02* | 3.62777*1* | 3210 | 5* .8305E+01* | .1850E+01* | .1829E+01* | 904.09 | | |
| .8110E+02* | 3.63063*0* | 3301 | 5* .8304E+01* | .1923E+01* | .1901E+01* | 935.06 | | |
| .1021E+03* | 3.63260*1* | 3393 | 5* .8298E+01* | .1997E+01* | .1972E+01* | 966.52 | | |
| .1285E+03* | 3.63402*0* | 3484 | 5* .8289E+01* | .2070E+01* | .2044E+01* | 998.12 | | |

1618E+03* 3.63493**1* 3576* 5* 8276E+01* 2144E+01* 2116E+01* 1029.82
 2037E+03* 3.63547**0* 3667* 5* 8258E+01* 2217E+01* 2187E+01* 1061.76
 2565E+03* 3.63602**2* 3816* 5* 8222E+01* 2291E+01* 2259E+01* 1103.62
 3229E+03* 3.63658**2* 3999* 5* 8171E+01* 2365E+01* 2331E+01* 1150.53
 4065E+03* 3.63730**2* 4182* 5* 8110E+01* 2439E+01* 2402E+01* 1197.07
 5117E+03* 3.63831**2* 4395* 5* 8039E+01* 2512E+01* 2474E+01* 1251.73
 6442E+03* 3.63971**2* 4608* 5* 7956E+01* 2586E+01* 2546E+01* 1307.17
 8110E+03* 3.64158**2* 4821* 5* 7860E+01* 2660E+01* 2617E+01* 1362.15
 1021E+04* 3.64404**2* 5034* 5* 7749E+01* 2735E+01* 2689E+01* 1417.98
 1285E+04* 3.64737**2* 5247* 5* 7621E+01* 2809E+01* 2761E+01* 1475.01
 1618E+04* 3.65104**2* 5460* 4* 7476E+01* 2883E+01* 2832E+01* 1535.08
 2037E+04* 3.65708**2* 5673* 4* 7313E+01* 2957E+01* 2904E+01* 1595.93
 2565E+04* 3.66391**2* 5886* 4* 7130E+01* 3032E+01* 2976E+01* 1658.26
 3229E+04* 3.67250**2* 6099* 4* 6929E+01* 3106E+01* 3047E+01* 1721.25
 4065E+04* 3.68314**2* 6312* 4* 6711E+01* 3181E+01* 3119E+01* 1786.82
 5117E+04* 3.69611**2* 6555* 4* 6477E+01* 3255E+01* 3191E+01* 1861.89
 6442E+04* 3.71164**2* 6798* 4* 6231E+01* 3330E+01* 3262E+01* 1937.09
 8110E+04* 3.73009**2* 7041* 4* 5978E+01* 3405E+01* 3334E+01* 2012.32
 1021E+05* 3.75200**2* 7284* 4* 5723E+01* 3480E+01* 3406E+01* 2089.22
 1285E+05* 3.77716**2* 7527* 3* 5472E+01* 3554E+01* 3477E+01* 2168.68
 1618E+05* 3.80586**2* 7770* 3* 5231E+01* 3629E+01* 3549E+01* 2248.17
 2037E+05* 3.83828**2* 8013* 3* 5007E+01* 3704E+01* 3621E+01* 2327.64
 2565E+05* 3.87451**2* 8256* 3* 4803E+01* 3780E+01* 3692E+01* 2407.06
 3229E+05* 3.91461**2* 8499* 3* 4620E+01* 3855E+01* 3764E+01* 2486.56
 4065E+05* 3.95874**2* 8742* 3* 4460E+01* 3930E+01* 3836E+01* 2566.91
 5117E+05* 4.00701**2* 9015* 3* 4322E+01* 4005E+01* 3907E+01* 2657.68
 6442E+05* 4.05929**3* 9313* 3* 4205E+01* 4081E+01* 3979E+01* 2753.05
 8110E+05* 4.11497**3* 9677* 3* 4104E+01* 4156E+01* 4051E+01* 2859.26
 1021E+06* 4.17441**3* 10042* 3* 4021E+01* 4232E+01* 4122E+01* 2965.61
 1285E+06* 4.23695**5* 10433* 3* 3953E+01* 4307E+01* 4194E+01* 3078.18
 1618E+06* 4.30316**3* 10816* 3* 3898E+01* 4383E+01* 4266E+01* 3197.40
 2037E+06* 4.37127**3* 11185* 3* 3853E+01* 4459E+01* 4337E+01* 3318.35
 2565E+06* 4.44124**3* 11549* 3* 3918E+01* 4535E+01* 4409E+01* 3440.22
 3229E+06* 4.51289**4* 11940* 3* 3792E+01* 4611E+01* 4481E+01* 3571.83
 4065E+06* 4.58539**4* 12395* 3* 3771E+01* 4687E+01* 4552E+01* 3719.49
 5117E+06* 4.65862**5* 12900* 3* 3757E+01* 4763E+01* 4624E+01* 3880.04
 6442E+06* 4.73237**4* 13410* 3* 3746E+01* 4839E+01* 4696E+01* 4044.06
 8110E+06* 4.80660**4* 13977* 3* 3738E+01* 4915E+01* 4767E+01* 4220.62
 1000E+07* 4.87286**2* 14523* 3* 3732E+01* 4984E+01* 4833E+01* 4388.52
 Buildup starting at tdf= 24601372E+07 CPU(MTRX CALC.)= 2806.630
 .4065E-04* 77.02718**2* 14526* 3* 3732E+01* NaNQ* NaNQ* 4390.89
 .5117E-04* 77.02719**1* 14528* 3* 3732E+01* NaNQ* NaNQ* 4391.54
 .6442E-04* 77.02721**1* 14530* 3* 3732E+01* NaNQ* NaNQ* 4392.19
 8110E-04* 77.02723**1* 14532* 3* 3732E+01* NaNQ* NaNQ* 4392.84
 1021E-03* 77.02726**1* 14534* 3* 3732E+01* NaNQ* NaNQ* 4393.48
 1285E-03* 77.02729**1* 14536* 3* 3732E+01* NaNQ* NaNQ* 4394.13
 1618E-03* 77.02733**1* 14538* 3* 3732E+01* NaNQ* NaNQ* 4394.78
 2037E-03* 77.02737**1* 14540* 3* 3732E+01* NaNQ* NaNQ* 4395.43
 2565E-03* 77.02743**1* 14542* 3* 3732E+01* NaNQ* NaNQ* 4396.08
 3229E-03* 77.02749**1* 14544* 3* 3731E+01* NaNQ* NaNQ* 4396.72
 4065E-03* 77.02757**1* 14546* 3* 3731E+01* NaNQ* NaNQ* 4397.37
 5117E-03* 77.02774**1* 14568* 3* 3731E+01* NaNQ* NaNQ* 4404.41
 6442E-03* 77.02797**1* 14590* 3* 3730E+01* NaNQ* NaNQ* 4411.46
 8110E-03* 77.02822**1* 14612* 3* 3730E+01* NaNQ* NaNQ* 4418.55
 1021E-02* 77.02851**1* 14634* 3* 3729E+01* 7692E+02* 7707E+02* 4425.62
 1285E-02* 77.02885**1* 14656* 3* 3728E+01* 7692E+02* 7707E+02* 4432.68
 1618E-02* 77.02924**1* 14678* 3* 3727E+01* 7692E+02* 7707E+02* 4439.74
 2037E-02* 77.02970**1* 14700* 3* 3726E+01* 7692E+02* 7707E+02* 4446.77
 2565E-02* 77.03024**1* 14722* 3* 3725E+01* 7692E+02* 7707E+02* 4453.81
 3229E-02* 77.03082**1* 14744* 3* 3723E+01* 7692E+02* 7707E+02* 4460.88
 4065E-02* 77.03168**1* 14766* 3* 3722E+01* 7692E+02* 7707E+02* 4467.95
 5117E-02* 77.03271**1* 14808* 3* 3719E+01* 7692E+02* 7707E+02* 4481.45
 6442E-02* 77.03398**1* 14850* 3* 3716E+01* 7692E+02* 7707E+02* 4494.93
 8110E-02* 77.03552**1* 14892* 3* 3713E+01* 7692E+02* 7707E+02* 4508.39
 1021E-01* 77.03738**1* 14934* 3* 3708E+01* 7692E+02* 7707E+02* 4521.88
 1285E-01* 77.03965**1* 14976* 3* 3703E+01* 7692E+02* 7707E+02* 4535.34
 1618E-01* 77.04245**1* 15018* 3* 3697E+01* 7692E+02* 7707E+02* 4548.82
 2037E-01* 77.04589**1* 15060* 3* 3689E+01* 7692E+02* 7707E+02* 4562.23
 2565E-01* 77.05015**1* 15102* 3* 3679E+01* 7692E+02* 7707E+02* 4575.71
 3229E-01* 77.05544**1* 15144* 3* 3667E+01* 7692E+02* 7707E+02* 4589.19
 4065E-01* 77.06201**1* 15186* 3* 3652E+01* 7692E+02* 7708E+02* 4602.68
 5117E-01* 77.07030**1* 15248* 3* 3632E+01* 7693E+02* 7708E+02* 4622.60
 6442E-01* 77.08060**1* 15310* 2* 3609E+01* 7695E+02* 7710E+02* 4642.50
 8110E-01* 77.09342**1* 15372* 2* 3579E+01* 7696E+02* 7711E+02* 4662.40
 1021E+00* 77.10936**1* 15434* 2* 3542E+01* 7699E+02* 7714E+02* 4682.28
 1285E+00* 77.12919**1* 15496* 2* 3497E+01* 7702E+02* 7717E+02* 4702.18
 1618E+00* 77.15384**1* 15558* 2* 3440E+01* 7706E+02* 7720E+02* 4722.24
 2037E+00* 77.18446**1* 15620* 2* 3369E+01* 7710E+02* 7724E+02* 4743.02
 2565E+00* 77.22242**1* 15682* 2* 3281E+01* 7715E+02* 7729E+02* 4763.80
 3229E+00* 77.26936**1* 15744* 2* 3173E+01* 7720E+02* 7734E+02* 4784.57
 4065E+00* 77.32724**1* 15806* 2* 3040E+01* 7726E+02* 7739E+02* 4805.35
 5117E+00* 77.39841**1* 15888* 2* 2875E+01* 7732E+02* 7745E+02* 4832.84
 6442E+00* 77.48530**1* 15970* 2* 2675E+01* 7739E+02* 7751E+02* 4860.80

.8110E+00* 77.59070*1* 16052* 2* .2432E+01* .7745E+02* .7757E+02* 4889.51
 .1021E+01* 77.71747*1* 16134* 2* .2139E+01* .7752E+02* .7764E+02* 4918.25
 .1285E+01* 77.86841*1* 16216* 1* .1791E+01* .7759E+02* .7770E+02* 4946.99
 .1618E+01* 78.04573*1* 16298* 1* .1382E+01* .7766E+02* .7777E+02* 4977.08
 .2037E+01* 78.25007*1* 16380* 1* .9106E+00* .7773E+02* .7784E+02* 5007.18
 .2565E+01* 78.47962*1* 16462* 1* .3814E+00* .7780E+02* .7790E+02* 5038.25
 .3229E+01* 78.69617*1* 16544* 1* .-1176E+00* .7788E+02* .7797E+02* 5069.29
 .4065E+01* 78.72456*1* 16626* 1* .-1820E+00* .7795E+02* .7804E+02* 5100.31
 .5117E+01* 78.72773*1* 16728* 1* .-1880E+00* .7802E+02* .7811E+02* 5137.67
 .6442E+01* 78.72983*1* 16830* 1* .-1912E+00* .7810E+02* .7818E+02* 5175.03
 .8110E+01* 78.73177*1* 16932* 1* .-1936E+00* .7817E+02* .7825E+02* 5212.42
 .1021E+02* 78.73368*1* 17034* 1* .-1954E+00* .7824E+02* .7833E+02* 5248.98
 .1285E+02* 78.73568*1* 17136* 1* .-1969E+00* .7832E+02* .7840E+02* 5284.71
 .1618E+02* 78.73782*1* 17238* 1* .-1979E+00* .7839E+02* .7847E+02* 5319.74
 .2037E+02* 78.74020*1* 17340* 1* .-1986E+00* .7847E+02* .7854E+02* 5353.53
 .2565E+02* 78.74291*1* 17442* 1* .-1990E+00* .7854E+02* .7861E+02* 5386.22
 .3229E+02* 78.74613*1* 17544* 1* .-1992E+00* .7862E+02* .7868E+02* 5418.89
 .4065E+02* 78.74809*1* 17598* 1* .-1992E+00* .7869E+02* .7875E+02* 5443.08
 .5117E+02* 78.75044*0* 17657* 1* .-1992E+00* .7876E+02* .7882E+02* 5471.80
 .6442E+02* 78.75332*0* 17718* 1* .-1990E+00* .7884E+02* .7890E+02* 5500.49
 .8110E+02* 78.75682*0* 17779* 1* .-1988E+00* .7891E+02* .7897E+02* 5529.18
 .1021E+03* 78.76143*0* 17858* 1* .-1987E+00* .7899E+02* .7904E+02* 5562.21
 .1285E+03* 78.76675*2* 17973* 1* .-1983E+00* .7906E+02* .7911E+02* 5600.73
 .1618E+03* 78.77366*1* 18170* 1* .-1975E+00* .7914E+02* .7918E+02* 5661.09
 .2037E+03* 78.78447*1* 18566* 1* .-1966E+00* .7921E+02* .7925E+02* 5783.74
 .2565E+03* 78.79727*1* 18981* 1* .-1953E+00* .7928E+02* .7933E+02* 5910.25
 .3229E+03* 78.81210*0* 19443* 1* .-1940E+00* .7936E+02* .7940E+02* 6048.07
 .4065E+03* 78.82672*1* 20022* 1* .-1925E+00* .7943E+02* .7947E+02* 6229.32
 .5117E+03* 78.84407*0* 20697* 1* .-1908E+00* .7951E+02* .7954E+02* 6441.10
 .6442E+03* 78.86435*0* 21549* 1* .-1889E+00* .7958E+02* .7961E+02* 6695.08
 .8110E+03* 78.88643*1* 22640* 1* .-1866E+00* .7965E+02* .7968E+02* 7027.05
 .1021E+04* 78.91654*2* 23855* 1* .-1837E+00* .7973E+02* .7976E+02* 7396.42
 .1285E+04* 78.94909*0* 25270* 1* .-1806E+00* .7980E+02* .7983E+02* 7866.26
 .1618E+04* 78.98664*0* 26701* 1* .-1769E+00* .7987E+02* .7990E+02* 8355.75
 .2037E+04* 79.03477*1* 28463* 1* .-1722E+00* .7995E+02* .7997E+02* 8984.43
 .2565E+04* 79.08202*1* 30205* 1* .-1677E+00* .8002E+02* .8004E+02* 9629.38
 .3229E+04* 79.13226*0* 32042* 1* .-1629E+00* .8009E+02* .8011E+02* 10332.78
 .4065E+04* 79.18853*0* 34080* 1* .-1575E+00* .8017E+02* .8018E+02* 11103.29
 .5117E+04* 79.25110*1* 36482* 1* .-1516E+00* .8024E+02* .8026E+02* 11995.15
 .6442E+04* 79.32016*1* 39110* 1* .-1451E+00* .8031E+02* .8033E+02* 12946.05
 .8110E+04* 79.39547*1* 42046* 1* .-1382E+00* .8038E+02* .8040E+02* 14023.15
 .1021E+05* 79.47719*1* 45278* 1* .-1307E+00* .8046E+02* .8047E+02* 15206.34
 .1285E+05* 79.56552*1* 48845* 1* .-1228E+00* .8053E+02* .8054E+02* 16315.37
 .1618E+05* 79.65950*1* 52744* 1* .-1145E+00* .8060E+02* .8061E+02* 17960.86
 .2037E+05* 79.76446*0* 57210* 1* .-1053E+00* .8067E+02* .8068E+02* 19791.34
 .2565E+05* 79.93696*1* 63403* 1* .-9085E-01* .8074E+02* .8075E+02* 22075.74
 .3229E+05* 80.12909*1* 70565* 1* .-7561E-01* .8081E+02* .8082E+02* 24524.77
 .4065E+05* 80.30233*1* 77763* 1* .-6270E-01* .8088E+02* .8089E+02* 27008.05
 .5117E+05* 80.48828*1* 85102* 1* .-5181E-01* .8095E+02* .8096E+02* 29623.35
 .6442E+05* 80.61778*1* 93294* 1* .-4151E-01* .8102E+02* .8103E+02* 32537.02
 .8110E+05* 80.76289*1* 101434* 1* .-3293E-01* .8109E+02* .8109E+02* 35448.07
 .1021E+06* 80.89105*1* 109196* 1* .-2603E-01* .8116E+02* .8116E+02* 38271.00
 .1285E+06* 81.00219*1* 116481* 1* .-2061E-01* .8122E+02* .8122E+02* 41052.38
 .1618E+06* 81.10225*0* 123486* 1* .-1628E-01* .8128E+02* .8129E+02* 43782.64
 .2037E+06* 81.18916*0* 129907* 1* .-1285E-01* .8135E+02* .8135E+02* 46363.64
 .2565E+06* 81.26615*1* 135905* 1* .-1008E-01* .8140E+02* .8141E+02* 48853.53
 .3229E+06* 81.33407*1* 141455* 1* .-7942E-02* .8146E+02* .8146E+02* 51294.20
 .4065E+06* 81.39435*0* 146586* 1* .-6314E-02* .8151E+02* .8151E+02* 53647.77
 .5117E+06* 81.45607*0* 152002* 1* .-4809E-02* .8156E+02* .8156E+02* 56044.98
 .6442E+06* 81.51197*0* 157054* 1* .-3616E-02* .8161E+02* .8161E+02* 58363.28
 .8110E+06* 81.56218*0* 161827* 1* .-2692E-02* .8165E+02* .8165E+02* 60585.80
 1000E+07* 81.60393*0* 165891* 1* .-2036E-02* .8168E+02* .8168E+02* 62480.02

E.5 Computer Program Code Listing

- NUMERICAL SOLUTION FOR A GRAVITY DRAINAGE PROBLEM INCLUDING A FREE SURFACE AND SEEPAGE FACE IN A RADIAL FLOW INTO A WELL.
- This program is written in FORTRAN.
- LIST OF INPUT DATA AND COMMENTS
- MM = Number of grid blocks in the radial direction. Typical value still on study (suggested around 30 or 40);
- N = Number of grid blocks in the vertical direction. Typical values around 60;
- Note: the above parameter can be set in the subroutine after conclusion of the SENSITIVITY ch.
- CI = Liquid compressibility, 1/psi. Sug. for water = 3.d-06;
- CT = Total compressibility, 1/psi. Sug. for incompres. = CI
- BO = Formation volume factor;
- RHO = Liquid density, lb/ft3;
- VISC = Liquid viscosity, cp;
- POROS = Porosity, fraction;
- HO = Original liquid height in the formation, ft;
- RW = Wellbore radius, inches;
- SKIN = Skin parameter;
- PERMH = Horizontal permeability, mD;
- PERMV = Vertical permeability, mD;
- Q = Liquid production, bbl/day;
- PWMIN = Minimum wellbore pressure allowed, psi;
- TFLW = Flowing time, min;
- TBUILD = Buildup time, min;
- SWR = Residual liquid saturation, fraction.
- ITERMX = Maximum number of iterations per time-step, set = 3;
- IFLOW = Constant(0) or variable (1) flow rate, set = 0;
- NQV = Number of flow rate steps, set = 0;
- RED = External dimensionless radius, set = 1.d+06;
- EPS = Small residual number, set = 1.d-06;
- NTPR = Number of times of pressure profile outputs, set = 0;
- IPRNT1 = Output control of units of wellbore press., set = 1; (=0, dimensionless; =1, field units)
- IPRNT2 = Output control of units of head profile, set = 0;
- IPRNT3 = Output control of units of total pres. dist., set = 1;
- TDI = Initial dimensionless time-step, set = 1.d-02;
 - Note: This value rules the successive time-step sizes;
- TDPR(1) = Dimensionless time for profile outputs. Must be included in dados subroutine, just after NTPR.
- NOBS = Number of observation wells, set = 0;
- DIST(K) = Distance of obs. wells from wellbore center, ft; no data
- TETA = Time relaxation param. (between 0. and 1.), set = 0.5;
- MATMAX = Max. numb. of matrix calculation allowed, set = 100,000;
- NTDI = Initial number of time-step per log cycle, set = 10;
- MULT = Definition of MULTIMODAL radial spacing, set = 1;
 - Note: =1, Multimodal; =0, logarithmic spacing;
- RMULT = Used in Multimodal mode, set = 400,000.
 - Note: it represents the ratio between the first and the last radial grid size. WARNING: it must be reevaluated if red .ne. 10e+6.
- PART(J) = Partial penetration parameter for each block. set = 1.:
 - Note: this value (between 0. and 1.) multiplies the individual each block sandface permeability.

- Author: F.R.COURI
- Characteristics of the program:
 - 1) Main program and several subroutines.
 - 2) Variable grid block angles of flow.
 - 3) Drawdown at variable flow rates (opt.).
 - 4) Multimodal radial grid or standard logarithmic spacing options.
 - 5) Progressive grid region calculation: only blocks within a perturbed cylindric region by the production are considered.
 - 6) Calculation of potentials at any degree of implicitness in time.
 - 7) Iterative wellbore potential calculation using explicit quadratic equation;
 - 8) Minimum wellbore flow pressure allowed;
 - 9) Analytical jacobian;
 - 10) Matrix solver: CB-solver:
 - Matrix solver that calls MSILUM, which performs incomplete factorization, and MSORMN, which does ORTHOMIN iterations.
 - 11) Main output files:
 - s2 = Normalized free surface profile at tdpr;
 - s3 = Normalized iso-potential curves at

```

*           the shuttin time tfw;
*           S11 =; Pres. drawdown, buildup pres., seepage
*           length, other control parameters vs. time;
*           S12 =; Head profile of Wyckoff et al. experiment;
*           For each observation well "k" located at different
*           positions dist(k) from the center, new
*           files "k*" (where * means "k" value) containing
*           head, seepage, Ramey et al.(1989) and Theis(1932)
*           solutions are generated. The "open" statements
*           need to be incremented.
*-----*
PARAMETER(mcm=80,ncm=100)
IMPLICIT REAL*8(A-H,O-Z)

common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&      hdav(mcm+1),zdavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),al(mcm),fr(mcm),permx(mcm,ncm),
&      rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /grd/pd(mcm,ncm),grad(mcm)
common /tempo/tmpr(10)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),d2d(mcm),dzavg(mcm+1),
&      dhavg(mcm+1)
common /coef/A(mcm,ncm),B(mcm,ncm),C(mcm,ncm),D(mcm,ncm),
&      E(mcm,ncm),F(mcm,ncm)
common /jacob/DA(mcm,ncm),DB(mcm,ncm),DC(mcm,ncm),DD(mcm,ncm),
&      DE(mcm,ncm),DF(mcm,ncm),DG(mcm,ncm)
common /angle/ cosn(mcm,ncm)
common /varflw/tq(20),qvar(20),aq(20),bq(20),cq(20)
common /deriv/dww(ncm,ncm)
common /profl/dist(30)
common /logic/iwhflg,iflow,ktnct,ktnflg,lgchs,ihw
common /iprnt/ijprn1,ijprn2,ijprn3
common /pcte/qdold,tdb,tdnxt,istar
common /dado1/h0,rw,visc
common /dimiss/td,dtd,hwd,hod,hs,dzdw,dhw,qd,red
dimension w1(mcm,ncm),dzold(mcm)
common /deb/g/tmin,iter

* Declaration of output files:
open(40,file='ramey',status='unknown')
rewind(40)
open(41,file='theis',status='unknown')
rewind(41)
open(2,file='s2',status='unknown')
rewind(2)
open(3,file='s3',status='unknown')
rewind(3)
open(11,file='s11',status='unknown')
rewind(11)
open(12,file='s12',status='unknown')
rewind(12)
open(13,file='s13',status='unknown')
rewind(13)
open(14,file='s14',status='unknown')
rewind(14)
open(15,file='s15',status='unknown')
rewind(15)
open(16,file='s16',status='unknown')
rewind(16)

* *** DATA INPUT

CALL DADOS  (MM,N,ITMX,IFLOW,NQV,RED,HO,RW,POROS,VISC,RHO,BO,
&SKIN,PERMV,PERMV,Q,EPS,PWMIN,NTPR,IPRNT1,IPRNT2,IPRNT3,IPRNT4,
&TDI,TBUILD,TFLW,CL,CT,NOBS,SWR,HC,TETA,MATMAX,NTDI,MULT,RMULT)

* Allowing for a varible teta in the program (just enter
* teta < 1 in the input data)
* This is not recommended for normal runs, only useful for tests
* with the program
if (teta .gt. 1.) then
  itet = 1
  tetai = teta
else
  itet = 0
endif

* *** PRELIMINARY CALCULATIONS OF CONSTANTS AND INITIAL VALUES

CALL PRECAL  (mm,n,ho,hw,visc,rho,rw,skin,q,pwmmin,iflow,nqv,
&           hc,Bo,tdi,cl,ct,noobs,FK,DU,CTIME,HWMIN,HCD,

```

```

& HWOLD,CLD,CTD,FS,REDDU,RDU,Mult,Rmult)
* Dimensionless buildup (tdbld) and flowing (tdflw) time
* OBS.: nbuild is a logic parameter for the buildup period.
nbuild = 0
tdbld = tbuild/ctime
tdflw = tflw/ctime
if (tdbld .ge. eps) nbuild = 1

* *** INITIAL VALUES
CALL INIT      (mm,m,iter,iterh,itime,ktlist,mflag,
&                  matriz,nprnt,ntd,ntdi,dzd(1),dtd1,cl,
&                  tdi,grado,hwd1,tdflw,tq(nqv) )

* Set a parameter for the matrix solver (set the order of expansion)
* ideg = 1 gives almost the same results as ideg = 2, although
* ideg = 1 allows a much faster solution.
ideg = 1

* *** PREPARE TOPS OF INPUT DATA SUMMARY IN SOME OUTPUT FILES:
call title(mm,n,nobs,ho,rw,skin,q,cl,ct,hc,
&          visc,swr,ctime,Red,Mult,Rmult,iprnt1)

* *** START OF NEXT TIME-STEP CALCULATIONS

* *** CALCULATION OF THE FLOW RATE (WHEN VARIABLE RATE IS BEING USED)
1  if (iflow .ne. 0) call FLWRT(iflow,nqv,td,tdflw,QD)

* *** MAXIMUM NUMBER OF ITERATIONS UPDATING: IT'S A FUNCTION OF THE
* DIMENSIONLESS TIME LOG CYCLE
itermx = diag10(td)
if (itermx .lt. 3) itermx = 3
if (itermx .lt. itmx) then
  itermx = itmx
endif
if (ktlist .gt. 10*itermx) ktlist = 10*itermx
if (dabs(hwd-hwold) .le. 0.001) ktlist = 3

* *** CHECK OUTER BOUNDARY EFFECTS AND EXTEND RESERVOIR EXTERNAL RADI
* IF NECESSARY

if (mflag .eq. 1) go to 3
do j = n,1,-1
  arg = w(m,j) - wo(j)
  if (dabs(arg) .gt. 1.d-09) then
    mflag = 1
    go to 2
  endif
enddo

2  if (mflag .eq. 1) then
    if (rd(m+1) .ge. red) then
      * When the outer boundary is reached
      * if (imflg.eq.0) print *, 'OUTER BOUNDARY REACHED AT TD = ',td
      imflg = 1
    else
      * When external boundary isn't reached, increase one additional
      * cylindric annulus:
      m = m + 1
      write(*,145) M,TMIN,ITER
    endif
    mflag = 0
  endif
145  format('GRID INCREASE TO M=' ,i3, ' AT TMIN=' ,e10.4, ' ITER=' ,i3)

* *** CALCULATION OF GEOMETRIC PARAMETERS
3  CALL GEOMET  (m,n,fs,reddu,rdu,skin,eps,alfw,matriz,lgchs)

* The following parameter is a reference for the first time that
* WHGH subrt. is called
iwhgflg = 1

* *** LABEL 2000: AUTO REDUCED TIMESTEP RESTART FROM HERE AFTER TDCOR
2000 tmin = ctime " td

=====
call cputime(time0,kcode)
time0 = time0/1000000.
=====

iter = 0
iterh = 0
idh = 0

```

```

hwd1 = hwd

* *** FIRST GUESS FOR HWD, CALCULATED WITH OLD W-VALUES
if (istar .eq. 1) then
  istar = 2
  write(*,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin
  write(11,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin
endif

* Subroutine WHGH is called for the first time in the time-step
CALL WHGH  (n,hwold,eps,hwmin,tmin,iter,
&           alfw,grado,cld,j1,hwd1,XJ,BWB,QDNW,QNW)
* if (ihw .ne. 0) go to 500
hwd1 = hwd

* START OF ITERATIVE PROCEDURE
* -----
* *** KEEP POTENTIAL VALUES TO BE COMPARED IN THE NEXT CALC. LOOP
100 do i = 1,m
    do j = 1,n
      w1(i,j) = w(i,j)
    enddo
  enddo

* *** CALCULATION OF MATRIX COEFFICIENTS
* MATCAL is a logical to skip repetition of same calculations
* of coefficients of the matrix which are not changed
* during the iteration process.
* MATCOF is the subrt. that calc. the matrix coefficients
* JACOBI is the jacobian coefficients subrt.

if (iter .eq. 0) then
  matcal = 0
else
  matcal = 1
endif
CALL MATCO2  (m,n,sk,rdu,ctd,swr,hcd,matcal)

CALL JACOBI  (m,n,j1,xj,Bwb,ctd,cld,hwold,
&           grado,lgchs,matcal,alfw,Swr,dww21)

* -----
* TEMPORARY ONLY FOR DEBUGGING
if(tmin .ge. 9.999d0 .and. tmin .le. 1.000d1) then
  if (matcl .le. 12) then
    call matdbg(m,n,iter,tmin,DA,DB,DC,DD,DE,DF,DG)
    matcl = matcl + 1
  endif
endif

* *** SOLVE THE JACOBIAN MATRIX
* -----
call cputime(timel,kcode)
timel = timel/1000000.
if (timel .gt. 180000.) then
  write(*,*) 'CPU TIME EXCEEDED THE LIMIT OF 180000 MINUTES'
  write(11,*) 'CPU TIME EXCEEDED THE LIMIT OF 180000 MINUTES'
  stop
endif

CALL SOLVE(m,n,ierr,ideg)
matriz = matriz + 1

if (ierr .ge. 3) then
  mixcnt = mixcnt + 1
  if (mixcnt .gt. 100) then
    write(*,999) cputot
    write(*,*) 'ABEND: MATRIX SOLVER IS UNABLE'
    stop
  endif
endif

* TDCOR resets the initial time-step condition
CALL TDCOR  (m,n,hwold,ktlst,tdbase)

* Check for the last time step in the flow period:
arg = (td - tdfw)/tdfw
if (dabs(arg) .gt. eps .and. lflow .eq. 1) lflow = 0
go to 2000
endif
MTXcnt = 0
* -----

```

```

call cputime(time2,kcode)
time2 = time2/1000000.
cpmat = time2 - time1
cpuot = cpuot + cpmat
*****  

c **** CALCULATE THE NEW THICKNESSES OF FREE SURFACE LAYER BLOCKS
CALL THICKN (m,n,grado,iter,alfw,tmin,eps)

* *** TIME-STEP CONTROL
*   If thickness of free surface layer has shrunked too much, restart
*   with smaller dt
500 if (idh .ne. 0 .or. ihw .ne. 0) then
    idhcnt = idhcnt +1
    if (idhcnt .gt.100) then
        write(*,999) cputot
        write(*,*) 'ABEND: NEGATIVE VALUES OF DH at i=' ,idh
        write(*,*) 'or ABEND DUE TO NEG. DELTA IN HW CALC.,IHW=' ,IHW
        stop
    endif

*   TDCOR resets the initial time-step condition
    CALL TDCOR (m,n,hwold,kilst,tdbase)

*   Check for the last time step in the flow period:
    arg = (td - tdfw)/tdfw
    if (dabs(arg) .gt. eps .and. lflow .eq. 1) lflow = 0
    go to 2000
  endif
  idhcnt = 0

* *** RECALCULATION OF HWD, WW(J) AND J1
iwhflg = 2
CALL WHGH (n,hwold,eps,hwmin,tmin,iter,
&           alfw,grado,cld,j1,hwd1,XJ,BWB,QDNW,QNW)
if (ihw .ne.0) go to 500

* *** CHECK CONVERGENCE
iflag = 0
iflgh = 0
arg = (hwd - hwd1)/hod
if (dabs(arg) .gt. eps) iflgh = 1
do i = 1,m
  do j = 1,n
    arg = w1(i,j) - w(i,j)
    if (dabs(arg) .gt. eps) then
      iflag = 1
      go to 300
    endif
  enddo
enddo

*   Check for the maximum number of matrix calculations
300 if(matriz .ge. matmax) then
    hs = hhold
    hwd = hwold
    td = td - dt
    do i = 1,m
      hdi(i)=hhold(i)
      dhi(i) = dhold(i)
      do j = 1,n
        w(i,j)=wold(i,j)
      enddo
    enddo
    CALL PRINTA (m,n,iter,matriz,istar,itermx,ntpr,iterh,tdfw,
&             fk,tdblk,j1,tdi,visc,rw,qlast,rho,ctime,skin,
&             qdnw,qnw,nobs,hwold,xj,dww21,ctd,cmat,cpuif,cpuot)
    write(11,9) matmax,time2
    write(*,9) matmax,time2
9   format('Program interrupted, matriz =',i7,' and cpu =',f9.2)
    go to 1001
  endif

*   Prepare next loop if iflag=1
  if (iflag .eq. 1) iter= iter+1
  if (iflgh .eq. 1) iterh = iterh +1
  if (iflag .eq. 1. or. iflgh .eq. 1) then
    if (iter .gt. itermx) then
      if (iter .le. 10) then
        if (kiflg .ge. 5000) then
          itermx = itermx + 1

```

```

        ktflag = 1
        endif
        endif
*      ISTAR = 1 detects the begining of constant wellbore press.
        if (istar.eq.1) then
          istar = 0
          id = tdnxt
          did = tdnxt - tdb
        endif
        CALL TDCOR (m,n,hwold,ktlist,tdbase)
        if (lflow .eq. 1) lflow = 0
        if (ktflag .ge. 1000000) then
          write(*,*) 'KTFLG = 10E+6, PROGRAM ABORTED, cput=',time2
          go to 1001
        endif
        go to 2000
      else
        hwd1 = hwd
      ****
      call cputime(time0,kcode)
      time0 = time0/1000000.
      ****
      go to 100
    endif
    endif
    if (istar .eq. 1) then .
      istar = 2
      write(*,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin
      write(11,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin
    endif
    if (td .gt. 10.) then
      * If there is any EHROR, reduce time-step
      if (idh .ne. 0) then
        idhcnt = idhcnt +1
        write(6,*) 'EHROR: hs.gt.hsold, hs=',hs,' hsold=',hsold
        CALL TDCOR (m,n,hwold,ktlist,tdbase)
        if (idhcnt .gt.1000) then
          write(*,999) cputot
          write(*,*) 'ABEND DUE TO NUMERICAL PROBLEMS IN THE IBC'
          stop
        endif
        go to 2000
      else
        idhcnt =0
        idh = 0
      endif
    endif
  **** END OF ITERATIVE PROCESS
  ****
  call cputime(time3,kcode)
  time3 = time3/1000000.
  ****

* Preparation of new wellbore flow rate in the const. press. case
  qdold = qd*bwb

* Prepare some time step control parameters
  arg = dabs(td - tdbase)
  if (arg .le. eps) then
    td = tdbase
    ktlist = ktflag
    ktflag = 1
    arg = tdbase - tdflw
    if (dabs(arg) .le. eps) then
      id = tdflw
      lflow = 1
    endif
  endif

* Numeric dispersion control of hd(i) - round-off
  do i = 1,m
    if (dabs((hd(i)-hold(i))/hd) .le. 1.d-10) hd(i) = hold(i)
    if (hd(i) .gt. hd) hd(i) = hd
  enddo

* **** PRINTOUT PROCEDURES

  arg = (tdflw - id)/tdflw
  if (dabs(arg) .le. eps) id = tdflw
  if (ktflag .eq. 1 .or. id .eq. tdflw) then

```

```

nprnt = nprnt + 1
if (nprnt .eq. ntd/10 .or. lflow .eq. 1) then
  if (lflow .eq. 1) qlast = qd
  cpudif = time3 - time0

  CALL PRINTA (m,n,iter,matrix,istar,itermx,ntpr,iterh,tdflw,
&           fk,tdbld,j1,tdi,visc,rw,qlast,rho,ctime,skin,
&           qdnw,qnw,nobs,hwold,xj,dww21,ctd,cpmat,cpudif,cpuot)
  nprnt = 0
  if (iter .eq. 1) call tetcti (hwd, td, tmin, teta, tetai)
endif
endif

* *** TIME-STEP CONTROL
if (ktflg .eq. 1) then
  CALL TIM(tdi,td,dtd,tdbld,itime,dtd1,tdflw,
&           qd,ntdi,ntd)
else
  if (istar .eq.1) then
    write(*,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN='',tmin
    write(11,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN='',tmin
    istar = 2
    dtd = tdnxt - td
    td = tdnxt
  else
    td = td + dtd
  endif
endif
arg = (tdflw - td)/tdflw
if (itime .eq. 1 .or. arg .le. eps .and. nbuild .eq. 0) then
  * Last time-step before stoping the program
  CALL PRINTA (m,n,iter,matrix,istar,itermx,ntpr,iterh,tdflw,
&           fk,tdbld,j1,tdi,visc,rw,qlast,rho,ctime,skin,
&           qdnw,qnw,nobs,hwold,xj,dww21,ctd,cpmat,cpudif,cpuot)
  Go to 1000
endif

* *** CONVERTING "OLD" INTO "NEW"
hwold = hwd

do i = 1,m
  hold(i) = hd(i)
  dzold(i) = dzd(i)
enddo

* *** VERTICAL GRID REDEFINITION
* Numerical control to avoid dispersion
if (dabs(dh(i)-dzd(i))/hod .le. 1.d-10) dh(i) = dzd(i)
* VERGRD recalculates all the potentials and grid vertical
* positions by interpolation, prior to the new time-step
CALL VERGRD(m,n,hod,dzold,tmin)

* *** PREPARATION OF ITERATIVE PROCESS FOR THE NEXT TIME-STEP
do i = 1,m
  Numerical control to avoid dispersion
  if (dabs(dzd(i)-dzold(i))/hod .le. 1.d-10) dzd(i) = dzold(i)
* Initialization of a new time-step
  dh1(i) = dh(i)
  dhold(i) = dzd(i)
  pd(i,1) = w(i,1) - grado * (zd(i,1)+dzd(i)/2.d0)
  do j = 1,n
    w1(i,j) = w(i,j)
    wold(i,j) = w(i,j)
  enddo
enddo
hs = hd(1)
hsold = hs
dhw = hs/n
dzdw = dhw

* *** INNER BOUNDARY POTENTIAL DETERMINATION (App. 1)
do j = 1,n
  zdw(j) = (n-j) * dzdw
  if (zdw(j) .lt. hwd. and. zdw(j)+dzdw .ge. hwd) j1 = j
enddo
if (lflow .eq. 2 .and. hs .lt. hwd) j1 = 1
xj = (hwd - zdw(j1))/dzdw
do 21 j = 1,n
  if (j .gt. j1 .or. hwd .gt. hs) then
    wwoold(j) = grado * hwd
    go to 21
  endif
enddo

```

```

        endif
        if (j.eq.j1) then
          wwoold(j) = grado/2.d0* ((1.d0+xj)*hwd + (1.d0-xj) *
          & (zdw(j) + dzdw))
        else
          wwoold(j) = grado * (zdw(j) + dzdw/2.d0)
        endif
21  continue
wwold(1) = wold(1,1)

*****
call cputime(time4,kcode)
time4 = time4/1000000.
dif4 = time4 - tt4
*****
*   Restart a new time-step
go to 1
999 format('PROGRAM INTERRUPTED AFTER CPUTOT=',F6.1)
1000 WRITE(*,*) 'THE END'
1001 stop
      end
*****

```

```

subroutine dados(M,N,ITMX,IFLOW,NQV,RED,HO,RW,POROS,VISC,RHO,BO
&,SKIN,PERMH,PERMV,Q,EPS,PWMIN,NTPR,IPRNT1,IPRNT2,IPRNT3,IPRNT4,
&TDI,TBUILD,TFLW,CL,CT,NOBS,SWR,HC,TETA,MATMAX,NTDI,MULT,RMULT)
PARAMETER(mcm=80,ncm=100)
* INPUT DATA READING SUBROUTINE

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

common /tempo/tdpr(10)
common /profi/dist(30)
common /varflw/tq(20),qvar(20),aq(20),bq(20),cq(20)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)

* *** Integer control parameters
read(*,1) m,n,mult,itmx,matmax,ntdi,iflow

* *** General fixed reservoir parameters
read(*,2) red,Rmult,cl,ct,ho,rw,pores,visc,rho,Bo

* *** Other reservoir properties and production data
read(*,3) skin,permh,permv,swr,hc

* *** Convergence checking and time-relaxation parameters
read(*,4) q,pwmin,epsi,teta

* *** Output print control
read(*,5) iprnt1,iprnt2,iprnt3,iprnt4

* *** Time control parameters
read(*,6) tdi,tbuild,tflw,ntpr
read(*,7) (tdpr(i),i=1,ntpr)

* *** Output profile
read(*,8) nobs
read(*,9)
read(*,10) (dist(i),i=1,nobs)

* *** Partial penetration parameters
read(*,11)
read(*,12) (part(j), j=1,n)

* *** Number of variable flow rate - nqv
read(*,13)
read(*,14) nqv

* *** Variable flow rates with time - tq(i), qvar(i)
do 100 i = 1,nqv
  read(*,15) tq(i),qvar(i)
100  continue

1  format(7(/,t60,i7))
2  format(10(/,t60,f16.7))
3  format(5(/,t60,e14.4))
4  format(4(/,t60,e14.4))
5  format(4(/,t60,i5))
6  format(3(/,t60,e12.4),/,t60,i5)
7  format(/,t60,i5)
8  format(/)

return
end
*****
```

```

SUBROUTINE INIT (mm,m,iter,iterh,itime,ktlst,mflag,
& matrix,npnrt,ntd,ntdi,dzd,dtd1,cl,
& tdi,grado,hwd1,tdflw,tq)
PARAMETER(mcm=80,ncm=100)
* SET UP INITIAL VALUES
implicit real*8 (a-h,o-z)
common /dado/poros,permh,permv,teta
common /dimls/td,did,hwd,hod,hs,dzdw,dhw,qd,red
common /logic/iwhfg,iflow,ktcnt,ktflg,lgchs,ihw
common/pcte/qdold,tdb,tdnxt,istar

* *** INITIAL VALUES
iter = 0
iterh = 0
itime = 0
istar = 0
ktcnt = 1
lflow = 0
ktflg = 1
ktlst = 1
mflag = 0
matrix = 0
npnrt = 0
lgchs = 0
ntd = ntdi
dhw = dzdw
dtd = tdi
dtd1 = tdi
grado = 1.d0 / hod
hwd1 = hwd
hs = hod
td = tdi
if (tdflw .lt. tdi) tdflw = tq
if (cl .eq. 0) then
  m = 3*mm/4
else
  m = mm/2
endif
if (red .lt. 100) m = mm

return
end

-----
SUBROUTINE PRECAL (mm,n,ho,hw,visc,rho,rw,skin,q,pwmin,iflow,nqv,
& hc,Bo,tdi,cl,ct,nobs,FK,DU,CTIME,HWMIN,HCD,
& HWOLD,CLD,CTD,FS,REDDU,RDU,Mult,Rmult)
PARAMETER(mcm=80,ncm=100)

* Subroutine for preliminary calculations
implicit real*8(a-h,o-z)

common /dimls/td,dtd,hwd,hod,hs,dzdw,dhw,qd,red
common /dado/poros,permh,permv,teta
common /alt/2d(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
& hdavg(mcm+1),zdavg(mcm+1)
common /grd/pd(mcm,ncm),grad(mcm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzdw(mcm),dzavg(mcm+1),
& dhavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),al(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /pres/wl(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /varflw/tq(20),qvar(20),aq(20),bq(20),cq(20)
common /profl/dist(30)

rw = rw / 12.d0
* Dimensionless initial free surface position hod
hod = ho / rw
* Dimensionless Fluid position in the wellbore
hwd = hod
* Dimensionless capillary fringe height
hcd = hc / rw
* Vertical to Horizontal Permeability ratio: fk
fk = permv/permh
* Dimensionless flow rate qd
qd = 2.03328d4 * q * visc / (permh * ho * rw * rho)
* Dimensionless Liquid and Total Compressibility
cld = ho * rho * cl / 144.d0
ctd = ho * rho * ct / 144.d0
* Geometric parameters

```

```

du = 1.d0/mm
reddu = red ** du
arg = (reddu*reddu+1.d0)/2.d0
rdu = dsqrt(arg)

* Time Conversion factor from min to dimensionless: t = ctime*td
ctime = 144.*60.d0*visc*poros*rw/(2.6368d-4*permh*hod*rho)
* Minimum or constant wellbore dimensionless liquid height:
hwmin = 144.d0 ** pwmin / (rw ** rho)
* Dimensionless time and variable flow rate
do 12 i = 1,nqv
    tq(i) = tq(i) / ctime
    qvar(i) = 2.03328d4 * qvar(i) * visc / (permh * ho * rw * rho)
12 continue
if (qvar(nqv).eq. 0.d0) then
    nnn = nqv -1
else
    nnn = nqv
endif
if (iflow .eq. 1) call spline(nnn,tq,qvar,az,bz,cz)

* Vertical coordinate of the lower surface of each block zd(i,j):
do 2 i = 1,mm
    zd(i) = hod / n
    zdavg(i) = (n-1) * zd(i)
    do 1 j = 1,n
        zd(i,j) = hod * j * zd(i)
        zdold(i,j) = zd(i,j)
1 continue
2 continue
zdavg(mm+1) = zdavg(mm)

* Inner Radial coordinate of each block:
CALL RADIUS (mm,mult,red,du,Rmult)

* INICIO DA ANTIGA SUBROTINA INIT
    hwd = hod
    hwdold = hod

* INITIAL DIMENSIONLESS POTENTIAL AND THICKNESS DISTRIBUTION
* -----
do 6 j = 1,n
    wo(j) = 1.d0
    ww(j) = 1.d0
    wwold(j) = 1.d0
    zdw(j) = zd(1,j)

    do 5 i = 1,mm
        wi(i,j) = 1.d0
        wold(i,j) = 1.d0
5 continue
6 continue

    do 7 i = 1,mm
        xz = zd(i)
        dh(i) = xz
        dhold(i) = xz
        dh1(i) = xz
        hd(i) = hod
        hold(i) = hod
        dhavg(i) = xz
        dzavg(i) = xz
        grad(i) = 1.d0 / hod
        pd(i,1) = w(i,1) . grad(i) * (zd(i,1)+dh(i)/2.d0)
* *** Calculation of the geometric parameter alf(i)
* *** alf(i) = 0.5d0
        alf(i+1) = (rd(i+1) . rbar(i))/(rbar(i+1) . rbar(i))

7 continue
    alf(1) = 1.d0
    alf(mm+1) = 1.d0
* Initial block thickness at the sandface
    dhavg(mm+1) = xz
    dzavg(mm+1) = xz
    zdzw = xz
    dhw = xz
    alfw = 1.d0

* *** Dimensionless radial distances for pressure profile output

```

```

      do 11 k= 1,nobs
11      dist(k) = dist(k)/rw
* *** Calculation of the radial geometric factor fr(i) (page 22)
      Ddow = dlog(Rbar(1))
      RARG = rbar(2)/rbar(1)
      Dup = dlog(RARG)
      fr(1) = Dup / Ddow
      do i = 1,mm-1
        RARG = rbar(i+2)/rbar(i+1)
        Dup = dlog(RARG)
        RARG = rbar(i+1)/rbar(i)
        Ddow = dlog(RARG)
        fr(i+1) = Dup / Ddow
        argfr = dabs(fr(i+1)-1.d0)
        if (argfr .le. 1.d-09) fr(i+1) = 1.d0
* Calculation of the areal parameter A1(i) (page 22)
      A1(i) = (rd(i+1)*rd(i+1)-rd(i)*rd(i))/2.d0*Ddow
      enddo
      A1(mm) = (rd*rd - rd(mm)*rd(mm))*Dup / 2.d0
      return
      end
*****
      subroutine flwrt(iflow,nqv,td,tdflw,QD)
      implicit real*8 (a-h,o-z)
      common /varflw/tq(20),qvar(20),aq(20),bq(20),cq(20)
      if (iflow .eq. 1) then
        qd = seval(nqv,td,tq,qvar,aq,bq,cq)
      else
        if (td .gt. tq(nqv) .and. td .lt. tdflw) then
          qd = qvar(nqv)
        else
          do 1 i = 1,nqv-1
            if (td .le. tq(i+1) .and. td .gt. tq(i)) then
              qd = qvar(i)
              return
            endif
          1      continue
        endif
      endif
      if (td .ge. tdflw) qd = 0.d0
      return
      end
*****

```

```

SUBROUTINE RADIUS(mm,mult,red,Rmult)
* RADIAL MESH DISTRIBUTION BASED ON THE MULTIMODAL METHOD BY
* TERAN and DE LA GARZA (1980)
PARAMETER(mcm=80,ncm=100)
IMPLICIT REAL*8 (A-H,O-Z)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
real*8 lambda

* Inner Radial coordinate of each block:
if (mult .eq. 0) then
  do 4 i = 1,mm
    u(i) = (i - 1)*du
    rd(i) = red ** u(i)
4  continue
  rd(mm+1) = red
  Rmult = (red - rd(mm))/(rd(2)-rd(1))
endif
lambda = Rmult**(1.0/(mm-1))
if (mult .ne. 0) then
  rd(1) = 1.0
  do i=2,mm
    rd(i) = 1.0+(red-1.0)**(lambda**(i-1))/(lambda**mm-1)
  enddo
  rd(mm+1) = red
endif
* *** Calculate rbar:
rbar(mm) = dlog(lambda) * red/(lambda-1.0)
rbar(mm+1) = red
do 1 i=mm-1,1,-1
  arg = (rd(i)*rd(i)+rd(i+1)*rd(i+1))/2.0
  rbar(i)=dsqrt(arg)
  do iter=1,100
    F = rbar(i) * rbar(i+1) + rd(i+1)**dlog(rbar(i+1)/rbar(i))
    del = F/(1.0*rd(i+1)/rbar(i))
    rbar(i) = rbar(i) - del
    if (dabs(del) .le. 1.0E-6) go to 1
  enddo
  print*, 'NO CONVERGENCE ON RBAR CALCULATIONS'
  stop
1 continue
print*, 'MULT=' ,MULT,'RMULT=' ,RMULT
do i=1,mm+1
  write(*,2) i,rd(i),rbar(i)
enddo
2 format(' rd(' ,i3,')=' ,e12.6,' rbar(' ,i3,')=' ,e12.6)
return
end

-----
SUBROUTINE GEOMET(m,n,fs,reddu,rd,skin,eps,alfw,matrix,lgch)
PARAMETER(mcm=80,ncm=100)

implicit real*8(a-h,o-z)
common /dado/poros,permh,permv,teta
common /dim/ls/td,dd,hd,hs,dzdw,dhw,qd,red
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
& hdavg(mcm+1),zdavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
& dhavg(mcm+1)
common /angle/ cosn(mcm,ncm)
dimension x(mcm+2),rold(mcm+1)

* *** Calculation of the axial permeability permx(1,j) (pages 13 and 22)
arg = 0.0
permhv = permh * permv
permh2 = permh * permh
permv2 = permv * permv
dln = dlog(rbar(1))
fs = dln/(skin + dln)
do 2 j = 1,n
  tbet = ((n-j)*1.0 + 0.5d0)**arg
  tbet2 = tbet * tbet
  cbet2 = 1.0/(1.0+tbet2)
  sbet2 = 1.0 - cbet2
  arg1 = permh2*sbet2 + permv2*cbet2
  denom = dsqrt(arg1)
  permx(1,j) = part(j) * fs * permh
2  continue

```

```

tbet = ((n-1)*1.d0 + 0.5d0)*arg
tbet2 = tbet * tbet
cbet2 = 1.d0/(1.d0+tbet2)
sbt2 = 1.d0 - cbet2
arg1 = permh2*sbt2 + permv2*cbet2
denom = dsqrt(arg1)
cbet = dsqrt(cbet2)
permx(1,1) = part(1) * cbet * fs * permh

* **** Calculation of the axial permeability permx(i,j) (pages 13 and 22)
do 4 i = 1,m-1
  arg = (dzd(i+1) - dzd(i)) / (rbar(i+1)-rbar(i))
  tbet = (1.d0*(n-1) + 0.5d0)*arg
  if (tbet .gt. 1.d0) tbet = 1.d0
  tbet2 = tbet * tbet
  cbet2 = 1.d0/(1.d0+tbet2)
  sbt2 = 1.d0 - cbet2
  arg1 = permh2*sbt2 + permv2*cbet2
  denom = dsqrt(arg1)
  cbet = dsqrt(cbet2)
  permx(i+1,1) = cbet * permhv / denom
  fx(i,1) = permx(i,1) / permx(i+1,1)
do 3 j = 1,n
  tbet = (1.d0*(n-j) + 0.5d0)*arg
  if (tbet .gt. 1.d0) tbet = 1.d0
  tbet2 = tbet * tbet
  cbet2 = 1.d0/(1.d0+tbet2)
  sbt2 = 1.d0 - cbet2
  arg1 = permh2*sbt2 + permv2*cbet2
  denom = dsqrt(arg1)
  cbet = dsqrt(cbet2)
  permx(i+1,j) = cbet * permhv / denom
  fx(i,j) = permx(i,j) / permx(i+1,j)
3   continue
4   continue

*   Calculate new thickness
if (matrix .ne. 0)
  &      call newtkn(m,n,hod,red,hs,alfw,dhw)

do 5 j = 1,n
  fx(m,j) = permx(m,j) / permh
5   continue

* *** COSINE OF EACH BLOCK SLOPE
do j = 1,n
  do i = 2,m-1
    tang = (zd(i+1,j)+dzd(i+1))/2.d0 - zd(i-1,j)-dzd(i-1)/2.d0
    &      / (rbar(i+1) - rbar(i-1))
    sec2 = 1.d0 + tang * tang
    sec = dsqrt(sec2)
    cosn(i,j) = 1.D0 / sec
  enddo
  cosn(m,j) = 1.d0
  tang = (zd(2,j)+dzd(2))/2.d0 - zdw(j) - dzdw/2.d0
  &      / (rbar(2) - 1.d0)
  sec2 = 1.d0 + tang * tang
  sec = dsqrt(sec2)
  cosn(1,j) = 1.D0 / sec
  enddo

return
end

***** subroutine newtkn(m,n,hod,red,hs,alfw,dhw,td,lgchs,rd,rbar)
PARAMETER(mcm=80,ncm=100) . .

IMPLICIT REAL*8(A-H,O-Z)

common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&      hdavg(mcm+1),zdavg(mcm+1)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&      dhavg(mcm+1)
dimension x(mcm+2),y(mcm+2),a(mcm+2),b(mcm+2),c(mcm+2)
dimension rd(mcm+1), rbar(mcm+1)

do 1 i = 1,m
  x(i+1) = rbar(i)
  y(i+1) = hd(i)
1   continue
x(1) = rd(1)

```

```

y(1) = hs
x(m+2) = red
y(m+2) = hod

do 2 i = 2,m
  dif = hd(i) - hd(i-1)
  if (dabs(dif).lt. 1.d-8) then
    xh = (hd(i) + hd(i-1))/2.d0
  else
    if (i.eq.2) then
      xh = aval(m+2,rd(i),x,y,0)
    else
      xh = aval(m+2,rd(i),x,y,1)
    endif
  endif
  dzavg(i) = xh/n
  dhavg(i) = dzavg(i)
  zdavg(i) = xh - dzavg(i)
2  continue

  xh = (hs + hd(1))/2.d0
  dzavg(1) = xh / n
  dhavg(1) = dzavg(1)
  zdavg(1) = xh - dzavg(1)

  dhw = hs / n
  dhavg(1) = dh(1)

  return
end
*****  

subroutine newtkn(m,n,hod,red,hs,alfw,dhw)
PARAMETER(mcm=80,ncm=100)

IMPLICIT REAL*8(A-H,O-Z)

common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
&          rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&          hdavg(mcm+1),zdavg(mcm+1)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&          dhavg(mcm+1)

do i=2,m
  hdavg(i) = alf(i)*hd(i) + (1.d0-alf(i))*hd(i-1)
  xh = hdavg(i)
  dzavg(i) = xh/n
  dhavg(i) = dzavg(i)
  zdavg(i) = xh - dzavg(i)
enddo
  xh = (hs + hd(1))/2.d0
  dzavg(1) = xh / n
  dhavg(1) = dzavg(1)
  zdavg(1) = xh - dzavg(1)
  dhw = hs / n
  dhavg(1) = dh(1)

  return
end
subroutine vergrd(m,n,hod,dzold,tmin)

PARAMETER(mcm=80,ncm=100)
implicit real*8 (a-h,o-z)
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&          hdavg(mcm+1),zdavg(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&          dhavg(mcm+1)
dimension xz(ncm+1),yw(ncm+1),dzold(1)
dimension alfa1(ncm+1),alfa2(ncm+1),alfa3(ncm+1)

nn = n + 1

do 20 i = 1,m
  dzd(i) = hd(i) / n
  do 10 j = 1,n
    zold(i,j) = zd(i,j)
    zd(i,j) = (n-j) * dzd(i)
10  continue
20  continue

  do 50 i = 1,m

```

```

      do 30 j = 2,n
         yw(n-j+1) = w(i,j)
         xz(n-j+1) = zold(i,j) + dzold(i)/2.d0
30      continue
         yw(n) = w(i,1)
         xz(n) = zold(i,1) + dh(i)/2.d0
         call spline(n,xz,yw,alfa1,alfa2,alfa3)
      do 40 j = 2,n
         zz = zd(i,j) + dzd(i)/2.d0
         w(i,j) = aval(n,zz,xz,yw,j-2)
40      continue
         w(i,1) = hd(i)/hod
50      continue

      return
      end
*****
SUBROUTINE THICKN (m,n,grado,iter,alfw,tmin,eps)
PARAMETER(mcm=80,ncm=100)
* Calculate hd(i) first based on the variation of w(i,1) and then
* calculate the dh(i)
IMPLICIT REAL*8(A-H,O-Z)

common /dado/poros,permh,permv,teta
common /dimiss/td,did,hwd,hod,hs,dzdw,dhw,qd,red
common /logic/iwhflg,lflow,k1cnt,k1flg,lgchs,ihw
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),zdw(ncm),
&      hdavg(mcm+1),zdavg(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wold(ncm),wo(ncm)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
&      rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /grd/pd(mcm,ncm),grad(mcm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&      dhavg(mcm+1)
dimension x(mcm+2),yh(mcm+2)
dimension ah(mcm+2),bh(mcm+2),ch(mcm+2)
dimension sz(mcm+2),bz(mcm+2),cz(mcm+2)

* Numerical control
do 2 i = m-2,m
   do 1 j = 1,n
      xcomp = w(i,j)-wold(i,j)
      if (dabs(xcomp) .le. 0.1d-10) w(i,j) = wold(i,j)
1      continue
2      continue
* *** RECALCULATE ALL HD(i) BASED ON W(i,1) AND WOLD(i,j)
do 3 i = 1,m
   hd(i) = w(i,1) * hod
   dh(i) = hd(i) - zd(i,1)
   pd(i,1) = w(i,1) * grado * (zd(i,1)+dh(i)/2.d0)
3      continue

   hsold = hs
   hs = hd(1)
   dhw = dh(1)
   x(1) = 1.d0
   yh(1) = hs
   do i=1,m
      x(i+1) = rbar(i)
      yh(i+1) = hd(i)
   enddo
   x(m+2) = rbar(m+1)
   yh(m+2) = hod

* *** CALCULATE DHAVG(I) from node 2 on.
do i=2,m
   hdavg(i) = alf(i) * hd(i) + (1.d0-alf(i))*hd(i-1)
   dhavg(i) = hdavg(i) - zdavg(i)
   arg = hd(i) - hd(i-1)
enddo
hdavg(1) = hd(1)
dhavg(1) = dh(1)
dhw = dh(1)

      return
      end
*****
SUBROUTINE SHRINK (m,n,iter,eps,w,wo,tmin)

```

```

* This subroutine shrinks the radial grid when the external
* boundary values resume the initial values during the shuttin
* period.
PARAMETER(mcm=80,ncm=100)
REAL*8 w(mcm,ncm),wo(ncm),eps
REAL*8 arg1, arg2
eps1 = 1.d-09
do j = n,1,-1
    arg2 = w(m-2,j) - wo(j)
    arg1 = w(m-1,j) - wo(j)
    if (dabs(arg2) .gt. eps1 .or. dabs(arg1) .gt. eps1 ) return
enddo
m = m - 1
write(*,1) M,TMIN,ITER
1 format('GRID DECREASE TO M=',i3,' AT TMIN=',e10.4,' ITER=',i3)
return
end
*****
```

```

SUBROUTINE WHGH (n,hwold,eps,hwmin,tmin,iter,
& alfw,grado,cld,j1,hwd1,XJ,BWB,QDNW,QNW)
PARAMETER(mcm=80,ncm=100)

* THIS SUBROUTINE CALCULATES THE WELBORE LIQUID LEVEL AND THE
* SANDFACE POTENTIAL FOR EACH BLOCK.
* THE QUADRATIC EQUATION IS CALCULATED BY THE SUBROUTINE WHEAD.

implicit real*8 (a-h,o-z)
common /dado/poras,permh,permv,teta
common /dado1/ho,rw,visc
common /dimss/td,dtd,hwd,hod,hs,dzdw,dhw,qd,red
common /logic/iwhflg,iflow,kcnt,kflg,lgch,ihw
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
& hdavg(mcm+1),zdavg(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
& dhavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),ix(mcm,ncm),part(ncm),alf(mcm+1)
common /pcte/qdold,tdb,tdnxt,istar

ifif = 0
deps = eps
tet1 = 1.d0 - teta

* *** Geometrical parameters

dr = dlog(Rbar(1))

* Extrapolated thickness in the wellbore sandface

hs = hd(1)
lgchs = 0
hdavg(1) = hd(1)
zdavg(1) = zd(1,1)
dhavg(1) = dh(1)
dhw = dh(1)
dzavg(1) = dzd(1)

* *** Initial guess of Hwd:
hwd = hwd1
hwdbf = hwd

* *** Iterative calculation of hwd:
icont = 1
itflg = 0

* Calculation of Bwb
1 Bwb = 4 d0*hod/(4 d0*hod + cld*(hwd+hwold))

* Find location of hwd
if (hwd .le. hod .and. hwd .gt. zdw(1)) then
  j1 = 1
  del = 1 d0
else
  del = 0 d0
  do 2 j = 2,n
    if (hwd .le. zdw(j-1) .and. hwd .gt. zdw(j)) j1 = j
2  continue
endif

* *** Height of liquid in the formation at the sandface

* CALCULATE THE FRACTION Xj

if (j1 .eq. 1) then
  xj = (hwd - zdw(1))/dhw
else
  xj = (hwd - zdw(j1))/dzdw
endif
if (xj .gt. 1.d0) xj = 1.d0

* *** CALCULATE THE WELBORE DIMENSIONLESS LIQUID HEIGHT
* THROUGH THE QUADRATIC EQUATION
CALL WHEAD (n,del,hwold,dr,hwd1,eps,hwmin,grado,j1,xj,bwb,
& Qdnw,Qnw,tmin)
if (ihw .ne. 0) return
if (istar .eq. 1) then
  write(*,111) iwhflg,itflg,hwdbf,hwd,dtd
  write(*,112) iter,w(1,1),w(1,n),hs,hwold
  ifif = ifif + 1

```

```

      if (ifif .ge. 100) stop
      go to 25
    endif
111  format('iwhflg=',i1,' itflg=',i3,' hwdbf=',e14.8,' hwd=',e14.8,
     & ' dtd=',e12.6)
112  format('iter=',i2,' w(1,1)=' ,e14.8,' w(1,n)=' ,e14.8,' hs=' ,
     & '          e14.8,' hwoold=' ,e14.8)

      arg1 = arg
      arg = (hwdbf + hwd)/hod

      if (dabs(arg) .gt. deps) then
        itflg = itflg + 1
        if (itflg .gt. 99) then
          itflg = 0
          if (icont .gt. 1) then
            write(*,*) 'Too much iteration in whgh'
            call contrl(n)
            write(*,*) 'hwdbf=' ,hwdbf,' hwd=' ,hwd,' iwhflg=' ,iwhflg
            write(*,*) 'zdw(j1)=' ,zdw(j1),' j1=' ,j1,' xj=' ,xj
            stop
          endif
          icont = icont + 1
          hwd = (hwd + hwdbf)/2.
          write(*,*) 'WARNING:',icont,'st HWD ITERATION VALUE ',
          & 'ASSUMED AFTER 100 ITERATIONS IN WHGH AT TMIN=' , TMIN
          go to 1
        endif
        hwdbf = hwd
        go to 1
      endif

      if (iflow .ne. 2) then
        if (hwd .gt. hs) hwd = hs
      endif

*   Calculate wellbore pressure in front of each block (App. 1)

25  ww(1) = w(1,1)
    do 3 j = 2,n
      if (j. gt. j1 .or. hwd .gt. hs) then
        ww(j) = grado * hwd
        go to 3
      else
        if (j .eq. j1) then
          ww(j) = grado/2.d0* ((1.d0+xj)*hwd + (1.d0-xj) *
          & (zdw(j1) + dzdw))
        else
          ww(j) = grado * (zdw(j) + dzdw/2.d0)
        endif
      endif
    3  continue

    return
  end

-----
subroutine wwder(n,j1,xj,dtd,poros,grado,bwb,qqd,cld,permh,
  & hwd,hwoold,teta,alfw,DWW21)
PARAMETER(mcm=80,ncm=100)

* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF THE INNER
* BOUNDARY POTENTIALS FOR THE JACOBIAN
implicit real*8 (a-h,o-z)

common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
  & rbar(mcm+1),fx(mcm,ncm),part(ncm),alff(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /grd/pd(mcm,ncm),grad(mcm)
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hhold(mcm),zdw(ncm),
  & hdavg(mcm+1);zdavg(mcm+1)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
  & dhavg(mcm+1)
common /deriv/dww(ncm,ncm)
common /logic/iwhflg,iflow,ktcnt,ktflg,lgchs,ihw

dimension dhdw(ncm+1)

dzd1 = dzavg(1)
dhd1 = dhavg(1)

```

```

call hwder(n,j1,xj,dtd,poros,grado,bwb,qd,cld,
& permh,dzd1,dhd1,hwd,hwold,teta,alfw,lgchs,DHDW)

do 1 j = 1,n
  do 2 k = 2,n
    if (j1 .gt. j) then
      dww(j,k) = 0.d0
    else
      if (j1 .lt. j) then
        dww(j,k) = grado * dhdw(k)
      else
        dww(j,k) = xj * grado * dhdw(k)
      endif
    endif
    if(j.eq.1) dww(1,k) = 0.d0
  2 continue
  if (j .eq. 1) then
    dww(1,1) = 1.d0
  else
    dww(j,1) = 0.d0
  endif
1 continue
dww21 = 0.d0
return
end
*****
```

```

subroutine hwder(n,j1,xj,dtd,poros,grado,bwb,qd,cld,
& permh,dzd1,dhd1,hwd,hwold,teta,alfw,lgchs,DHDW)
  PARAMETER(mcm=80,ncm=100)

* WELLBORE LIQUID HEIGHT DERIVATIVE

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

common /geom/u(mcm),rd(mcm+1),al(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
dimension dhdw(1)

* FINDING PARAMETER DELJ1:
if (j1 .eq. 1) then
  delj1 = 0.d0
else
  delj1 = 1.d0
endif

* CONSTANT CALCULATIONS

denom1 = .qd*Bwb*grado*cld / (4.d0 + grado*(hwd + hwold) * cld)
denom2 = 5.d-1 /(poros * dtd)
denom3 = teta /(permh * dlog(Rbar(1)))
xlogc = lgchs
xnum1 = (delj1 + (1.d0 - delj1) * (1.d0 - xj*xj))** (1.d0-xlogc)
xnum2 = w(1,1) . ww(1)
alf1 = (1.d0 + alfw)/2.d0
xnum3 = dhd1*(1.d0-alfw/2.d0*xnum1) + alf1*xnum2/grado

Sperm = 0.d0
do 1 j = j1 + 1,n
  Sperm = Sperm + permx(1,j)
1 continue
denom4 = (1.d0 - delj1)**dhd1*permx(1,1) *xj
  + zdz1 *(Sperm + delj1 * xj *permx(1,j1))

denom = denom1 + denom2 + grado * denom3 * denom4

dhdw(1) = 0.d0

do 2 k = 2,n
  dhdw(k) = denom3 * zdz1 * permx(1,k) / denom
2 continue

xnum4 = dhd1 * xnum1 - xnum2/grado
dhdw(n+1) = 0.d0

return
end
```

```

***** SUBROUTINE WHEAD (n,del,hwold,dr,hwd1,eps,hwmin,grado,j1,xj,bwb,
      Qdnw,Qnw,tmin)
      PARAMETER(mcm=80,ncm=100)

* CALCULATION OF THE QUADRATIC INNER BOUNDARY EQUATION
      implicit real*8 (a,h,o,z)
      real*8 kxwol,kxnwol,kxw,kxnw
* The extended precision is only required when calculating
* the wellbore flow rate in a constant pressure i.b.c.
* It works only on mainframes or IBM RISC machines, not on DEC's.
      real*16 a,b,c,delta,dois,quatr,hwda,hwdb,xhmin,c2,ecor,c1

      common /dado/poros,permh,permv,teta
      common /dado1/ho,rw,visc
      common /dimlas/td,dtd,hwd,hod,hs,dzdw,dhw,qd,red
      common /logic/iwhflg,iflow,kicnt,kflg,lgchs,ihw
      common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
      &      hdavg(mcm+1),zdavg(mcm+1)
      common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
      &      rbar(mcm+1),fx(mcm,ncm),par(mcm),al(mcm+1)
      common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
      common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
      &      dhavg(mcm+1)
      common/pcte/qdold,idb,tdnxt,istar

      dois = 2.d0
      quatr = 4.d0
      hsold = zdw(1) + dzdw
      if (hsold .le. hs .and. iflow .ne. 2) then
         theta = 1.d0
      else
         theta = teta
      endif

      if (hwold .gt. zdw(1)) then
         j2 = 1
      else
         do 2 j = 2,n
            if (hwold .le. zdw(j-1) .and. hwold .gt. zdw(j)) j2 = j
         2 continue
      endif
      xj2 = (hwold - zdw(j2))/dzdw
      if (hwold .gt. hsold) xj2 = 1.d0

      sum1 = 0.d0
      sum2 = 0.d0
      sum3 = 0.d0
      sum4 = 0.d0
      sum5 = xj2 * permx(1,j2)
      sum6 = (1.d0-xj2) * permx(1,j2)

c *** CALCULATION OF SUMMATIONS.

      do 1 j = 1,n
         if (j .gt. j1) sum1 = sum1 + permx(1,j)
         if (j .ne. 1) sum2 = sum2 + permx(1,j) * w(1,j)
         if (j .lt. j1 .and. j .ne. 1) sum3 = sum3 + permx(1,j)
         sum4 = sum4 + permx(1,j) * wold(1,j)
         if (j .gt. j2) sum5 = sum5 + permx(1,j)
         if (j .lt. j2) sum6 = sum6 + permx(1,j)
      1 continue

* Calculation of average permeabilities at the sandface

      if (hwold .le. eps) then
         kxwol = 0.d0
      else
         xx = 1.d0*(n - j2) + xj2
         kxwol = sum5 / xx
      endif
      arg = hsold - hwold
      if (arg .le. eps) then
         kxnwol = 0.d0
      else
         kxnwol = sum6 / (j2 * 1.d0 - xj2 )
      endif
      if (hwd .eq. 0.d0) then
         kxw = 0.d0
      else
         if (hwd .lt. hs) then
            kxw = ( dzdw * sum1 + xj * (del*dhw + (1.d0-del) *

```

```

&      dzdw) * permx(1,j1)) / hwd
else
  kxw = (dzdw * sum1 + dhw * permx(1,1)) / hs
endif
endif
arg = (hs - hwd) * grado
if (arg .le. 0.0) then
  kxnw = 0.0d0
else
  kxnw = ((1.0d0-xj)*permx(1,j1)*(del*dhw + (1.0d0-del)*
&      dzdw) + dzdw*sum3 + (1.0d0-del)*dhw*
&      permx(1,1)) / (hs - hwd)
endif
sum2 = dzavg(1) * sum2 + dhavg(1) * permx(1,1) * w(1,1)

* *** FLOW RATE FROM THE SEEPAGE FACE
call qseep  (n,j1,j2,teta,xj,xj2,dhavg(1),dzavg(1),
&      permh,dr,Qdnw,Qnw)

* *** COEFFICIENTS OF THE QUADRATIC EQUATION
A = grado * theta / (dr * permh) * (kxw - kxnw/2.0d0)
B = 5.0-1 / (poros * dt)
C = qd * bwb * B * hwo - 1.0d0 / (dr * permh) *
  (theta * (sum2 - kxnw * hs * hs * grado / 2.0d0) +
  (1.0d0 - theta) * (sum4 * dzavg(1) * (hwo - hwo * hwo * kxwo) +
  + (hsold*hsold - hwo*hwo)/2.0d0 * kxnwo) * grado) )

* *** CALCULATE NEW FLOW RATE FOR A CONSTANT WELBORE PRESSURE
if (istar .eq. 2) then
  hwd = hwmn
  if (hwmn .le. 1.0d-6) then
    xhmin = 0.0d0
    C1 = - 1.0d0 / (dr * permh) *
      (theta * (sum2 - kxnw * hs * hs * grado / 2.0d0) +
      (1.0d0 - theta) * (sum4 * dzavg(1) * hsold*hsold/2.0d0 * kxnwo) * grado)
  else
    xhmin = hwmn
    ecor = xhmin - hwmn
    xhmin = xhmin - ecor
    C1 = - B * xhmin * 1.0d0 / (dr * permh) *
      (theta * (sum2 - kxnw * hs * hs * grado / 2.0d0) +
      (1.0d0 - theta) * (sum4 * dzavg(1) * (xhmin * xhmin * kxwo) +
      + hsold*hsold * xhmin*xhmin/2.0d0 * kxnwo) * grado)
  endif
  qD = (-A*xhmin*xhmin + B*xhmin * C1)/Bwb
  return
endif

* *** POSITIVE ROOT OF THE QUADRATIC EQUATION
*      A.Hwd^2 + B.Hwd + C = 0
delta = B * B + quatr * A * C
arg = delta
ihw = 0
if (dabs(arg) .le. eps) delta = 0.0d0
if (delta .lt. 0.0d0) then
  write(*,*) 'SQRRT OF NEGATIVE VALUE IN whead, AT TD=',TD
  ihw = 111
  write(*,*) 'delta= ',delta
  return
endif
hwda = (- B + dsqrt(arg)) / (dois * A)
hwdb = (- B - dsqrt(arg)) / (dois * A)

if (hwda .lt. hwmn) then
  if (hwdb .gt. hwmn .and. hwdb .lt. hod) then
    hwd = hwdb
  else
    100    arg1 = (hwdb - hwmn)/hod
    101    arg2 = (hwda - hwmn)/hod
    102    if (dabs(arg1) .le. eps .or. dabs(arg2) .le. eps) then
    103      istar = 2
      write(*,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin
      write(11,*) 'START CONSTANT PRESSURE I.B.C. AT TMIN=',tmin

```

```

104      hwd = hwmin
105      return
106      else
107          print*, 'CONSTANT PRESSURE T-STEP CORRECTION, TMIN=',tmin
108          call ctpre(n,dtd,td,a,b,c,hwmin,hwold,poros,qd)
109          hwd = hwmin
110      endif
111      endif
112      continue
113  else
114      if (hwdb .le. hwmin) then
115          if (hwda .gt. hs .and. lflow .ne. 2) then
116              if (hwda .gt. hod) then
117                  call ctpre(n,dtd,td,a,b,c,hwmin,hwold,poros,qd)
118                  hwd = hwmin
119              endif
120          else
121              print*, 'hwda='',hwda,' hwdb='',hwdb
122              print*, 'hwmin='',hwmin,' hs='',hs
123              print*, '*****'
124              print*, 'WARNING: INSTABIL. IN THE WELLBORE HEAD CALC.'
125              endif
126          else
127              arga = hwd1 - hwda
128              argb = hwd1 - hwdb
129              xa = dabs(arga)
130              xb = dabs(argb)
131              if (xa .gt. xb) then
132                  hwd = hwdb
133              else
134                  hwd = hwda
135              endif
136          endif
137      else
138          arga = hwd1 - hwda
139          argb = hwd1 - hwdb
140          xa = dabs(arga)
141          xb = dabs(argb)
142          if (xa .gt. xb) then
143              hwd = hwdb
144          else
145              hwd = hwda
146          endif
147      endif
148  endif
149  return
150 end
151
152 subroutine hacntl(hs,hd1,hsold,1DH,hod,eps,tmin)
153
154      ! THIS IS A SUBROUTINE TO CONTROL THE TIME-STEP SIZE, ACCORDING
155      ! TO THE VARIATION OF THE SEEPAGE TOP
156      ! It is not in operation
157      implicit real*8 (a-h,o-z)
158      common /logic/iwhflg,lflow,ktcnt,kflg,lgchs,ihw
159      arg = rhsold - hs)/hod
160      lgchs = 0
161      if (lflow .eq.0 .or. lflow .eq. 1) then
162          if (arg .lt. -eps) then
163              lgchs = 1
164              idh = 10
165          else
166              idh = 0
167              lgchs = 0
168          endif
169      endif
170  return
171 end
172
173 subroutine ctpre(n,dtd,td,a,b,c,hwmin,hwold,poros,qd)
174
175      ! THIS SUBROUTINE IS USED TO SET AN ACCURATE TIME WHEN THE CONSTANT
176      ! FLOW RATE INNER CONDITION BECOMES A CONSTANT PRESSURE CONDITION.
177
178      real*8 qdold,tdb,tdnxt,dtd,td,hwmin,hwold,poros,qd
179      real*8 aux,dtds,tdst
180      ! EXTENDED PRECISION ONLY FEASABLE IN IBM RISC MACHINES
181      real*16 a,b,c
182      real*8 a,b,c
183      common/pcte/qdold,tdb,tdnxt,istar
184
185      aux = a*hwmin*hwmin + c + hwold/(2.d0 * poros * dtd)
186      dtds = (hwold - hwmin) / (2.d0 * poros * aux)
187      tdst = td + dtd + dtds
188      if (istar .eq. 0) then

```

```

tdb = td + dtd
tdnxt = td
istar = 1
endif
dtd = dtds
td = tdst
return
end
*****  

subroutine qseep (n,j1,j2,teta,xj1,xj2,dhavg,dzavg,
& permh,dr,Qdnw,Qnw)  

* CALCULATION OF THE FLOW RATES IN THE SEEPAGE FACE AND IN THE
* WET FACE OF THE BOREHOLE.
* IT IS USED ONLY AS OUTPUT INFORMATIONS.
PARAMETER(mcm=80,ncm=100)
implicit real*8 (a-h,o-z)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),a1f(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /dado1/ho,rw,visc
if (j1 .eq. 1) then
  Qdnw = part(1)/(permh*dr) * (teta * (1.d0-xj1) * dhavg *
& permx(1,1) * (W(1,1) - Ww(1)) +
& (1.d0-teta) * (1.d0 - xj2) * dzavg * permx(1,1) *
& (Wold(1,1)-Wwold(1)))
  Qnw = permh * ho * rw * Qdnw /(2.03328d4 * visc)
else
  sum1 = part(1) * dhavg * permx(1,1) * (W(1,1) - Ww(1)) +
& part(j1) * (1.d0 - xj1) * dzavg * (W(1,j1) - Ww(j1))
  sum2 = part(1) * dzavg * permx(1,1) * (Wold(1,1) - Wwold(1)) +
& + part(j2) * (1.d0 - xj2) * dzavg * (W(1,j2) - Ww(j2))
  do j = 2,j1-1
    sum1 = sum1 + part(j)*dzavg * permx(1,j) * (W(1,j) - Ww(j))
  enddo
  do j = 2,j2-1
    sum2 = sum2 + part(j) * dzavg * permx(1,j) *
& (Wold(1,j) - Wwold(j))
  enddo
  Qdnw = 1.d0/(permh*dr) * (teta * sum1 + (1.d0-teta) * sum2)
  Qnw = permh * ho * rw * Qdnw /(2.03328d4 * visc)
endif
return
end
*****
```

```

SUBROUTINE MATCO2  (m,n,sk,rdu,ctd,swr,hcd,matcal)
PARAMETER(mcm=80,ncm=100)
* THIS SUBR. CALCULATES THE COEFFICIENTS OF THE BLOCK PRESSURES.

implicit real*8(a-h u-z)

common /dato/poros,permh,permv,teta
common /dimiss/tid,tdt,hwd,hod,hs,dzdw,dhw,qd,red
common /alt/2d(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&          dhavg(mcm+1),zdavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
&          rbar(mcm+1),fx(mcm,ncm),part(ncm),al(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),zdavg(mcm+1),
&          dhavg(mcm+1)
common /coef/A(mcm,ncm),B(mcm,ncm),C(mcm,ncm),D(mcm,ncm),
&          E(mcm,ncm),F(mcm,ncm)
common /grd/pd(mcm,ncm),grad(mcm)
common /angle/cosn(mcm,ncm)
dimension fval(mcm),altd(mcm),cdterm(mcm),cdtrml(mcm)
dimension aux4(mcm), aux5(mcm), aux6(mcm), aux7(mcm)

* Calculation of group of constants for optimization and vectorization
  sg = 1.d0 - swr
  tet1 = 1.d0 - teta
  tet2 = 2.d0 * teta
  aux1 = permh / dtd
  aux2 = ctd / 2.d0
  aux3 = teta * permv
  do i = 2,m-1
    fval(i) = permv * A1(i) / permx(i+1,1)
    altd(i) = aux1 / permx(i+1,1) * A1(i) / cosn(i,1)
    cdtrml(i) = aux2 * altd(i) * (dzd(i)+dh(i))
  enddo

* *** Free Surface Equation
  do i = 2, m-1
    B(i,1) = 0.d0
    C(i,1) = tet2 * fval(i) / (dh(i) + dzd(i))
    D(i,1) = teta * fx(i,1) * fr(i) * (hcd + dhavg(i))
    E(i,1) = teta*(hcd + dhavg(i+1))
    F(i,1) = altd(i)*(dh(i)-dzd(i))*sg - cdtrml(i) * wold(i,1)
    & - tet1*((hcd+dzavg(i+1))*(wold(i+1,1)-wold(i,1)) +
    & fx(i,1)*fr(i)*(hcd+dzavg(i))*(wold(i-1,1)-wold(i,1)) +
    & + fval(i)/dzd(i)*(wold(i,2)-wold(i,1)))
  enddo
  do i = 2,m-1
    A(i,1) = - E(i,1) * D(i,1) - C(i,1) * cdtrml(i)
  enddo

* *** Second Layer Equation
  do i = 2, m-1
    altd(i) = aux1 / permx(i+1,2) * A1(i) / cosn(i,2)
    fval(i) = permv * A1(i) / permxi+1,2)
    cdterm(i) = altd(i) * dzd(i) * ctd
  enddo
  do i = 2, m-1
    B(i,2) = tet2 * fval(i) / (dzd(i)+dh(i))
    C(i,2) = teta * fval(i) / dzd(i)
    D(i,2) = teta*fx(i,2)*fr(i)*dzavg(i)
    E(i,2) = teta*dzavg(i+1)
    F(i,2) = - cdterm(i) * wold(i,2) - tet1 * ( dzavg(i+1) *
    & wold(i+1,2)-wold(i,2)) - fx(i,2)*fr(i)*dzavg(i) *
    & (wold(i,2)-wold(i-1,2)) + fval(i)/dzd(i) * (wold(i,1) +
    & 2.d0*wold(i,2) + wold(i,3)) )
  enddo
  do i = 2, m-1
    A(i,2) = -B(i,2) - C(i,2) - D(i,2) - E(i,2) * cdterm(i)
  enddo

  if (matcal .eq. 0) then
* *** Lower Boundary Equation
    do i = 2,m-1
      altd(i) = aux1 / permx(i+1,n) * A1(i) / cosn(i,n)
      fval(i) = permv * A1(i) / permx(i+1,n)
      cdterm(i) = altd(i) * dzd(i) * ctd
    enddo
    do i = 2,m-1
      B(i,n) = teta * fval(i) / dzd(i)
      C(i,n) = 0.d0
      D(i,n) = teta*fx(i,n)*fr(i) * dzavg(i)
      E(i,n) = teta*dzavg(i+1)
    enddo
  end if

```

```

F(i,n) = . cdterm(i) * wold(i,n) * tet1 * (dzavg(i+1) *
& (wold(i+1,n) - wold(i,n)) * fx(i,n)*fr(i)* dzavg(i)
& * (wold(i,n) - wold(i-1,n)) + fval(i)/dzd(i) *
& (wold(i,n-1) - wold(i,n)))
enddo
do i = 2,m-1
  A(i,n) = - B(i,n) - D(i,n) - E(i,n) - cdterm(i)
enddo

* *** Normal Layer Equations
do i = 1,m-1
  aux4(i) = aux3 = A1(i) / dzd(i)
  aux5(i) = teta * fr(i) * dzavg(i)
  aux6(i) = teta * dzavg(i+1)
  aux7(i) = fr(i) * dzavg(i)
enddo
do i = 1,m-1
  do j = 3, n-1
    B(i,j) = aux4(i) / permx(i+1,j)
    D(i,j) = aux5(i) * fx(i,j)
    E(i,j) = aux6(i)
  enddo
enddo

do i = 2, m-1
  do j = 3, n-1
    altd(i) = aux1 * A1(i) / permx(i+1,j) /cosn(i,j)
    cdterm(i) = aux1 * A1(i) / permx(i+1,j) * dzd(i) * ctd
    fval(i) = permv * A1(i) / permx(i+1,j)
    C(i,j) = B(i,j)
    A(i,j) = -2.d0*B(i,j) - D(i,j) - cdterm(i)
    F(i,j) = - cdterm(i) * wold(i,j) - tet1 * ( dzavg(i+1) *
& (wold(i+1,j)-wold(i,j)) - fx(i,j) * aux7(i) *
& (wold(i,j)-wold(i-1,j)) + fval(i)/dzd(i) *
& (wold(i,j-1)- 2.d0*wold(i,j) + wold(i,j+1) ) )
  enddo
  enddo
endif

* *** OUTER BOUNDARY LAYERS

fk1 = fk * A1(m)
altd(m) = A1(m) / dtd
cdterm(m) = altd(m) * dzd(m) * ctd
cdtrml(m) = aux2 * altd(m) * (dzd(m)+dh(m))

* *** Free Surface Equation

B(m,1) = 0.d0
C(m,1) = 2.d0*teta * fk1 / (dzd(m) + dh(m))
D(m,1) = teta*fx(m,1) * fr(m) * (hcd + dhavg(m))
E(m,1) = 0.d0
A(m,1) = -teta*dhavg(m+1) - C(m,1) - D(m,1) - cdtrml(m)
F(m,1) = altd(m)*(dh(m)-dzd(m))*sg*cdtrml(m)*wold(m,1)-dzavg(m+1)
& * (1.d0*tet1*wold(m,1)) - tet1 * ( -fx(m,1)*fr(m) *
& (hcd + dhavg(m)) *
& (wold(m,1)-wold(m-1,1)) - fk1/dzd(m) *
& (wold(m,1) - wold(m,2)) )

* *** Second Layer Equation

B(m,2) = tet2 * fk1/(dh(m)+dzd(m))
C(m,2) = teta*fk1/dzd(m)
D(m,2) = teta * fx(m,2) * fr(m) * dzavg(m)
E(m,2) = 0.d0
A(m,2) = -teta*dzavg(m+1) * B(m,2) - C(m,2) - D(m,2) - cdterm(m)
F(m,2) = - cdterm(m) * wold(m,1) * dzavg(m+1)*(1.d0 - tet1 *
& wold(m,2)) - tet1 * (-fx(m,2)*fr(m)*dzavg(m) *
& (wold(m,2)-wold(m-1,2)) + fk1/dzd(m)*(wold(m,1) *
& - 2.d0*wold(m,2) + wold(m,3) ) )

if (matcal.eq. 0) then
* *** Lower Boundary Equation
  B(m,n) = teta*fk1 /dzd(m)
  D(m,n) = teta * fx(m,n) * fr(m) * dzavg(m)
  A(m,n) = -teta*dzavg(m+1) * B(m,n) - D(m,n) - cdterm(m)
  C(m,n) = 0.d0
  E(m,n) = 0.d0
  F(m,n) = - cdterm(m) * wold(m,n) * dzavg(m+1)*(1.d0-tet1 *
& wold(m,n)) - tet1 * (-fx(m,n)*fr(m)*dzavg(m) *
& (wold(m,n)-wold(m-1,n)) + fk1/dzd(m)*(wold(m,n-1) *
& wold(m,n) ) )

```

```

* *** Normal Layer O.B.C. Equations
do j = 3,n-1
  B(m,j) = teta* fka1/dzd(m)
  D(m,j) = teta * fx(m,j)*fr(m)*dzavg(m)
  E(m,j) = 0.d0
  F(m,j) = - cdterm(m) * wold(m,j)- dzavg(m+1)*(1.d0-tet1*
    &           wold(m,j)) - tet1*((-fx(m,j)*fr(m)*dzavg(m)*
  &           (wold(m,j)-wold(m-1)))+fka1/dzd(m)*(wold(m,j-1)*
  &           2.d0*wold(m,j) + wold(m,j+1) ) )
  &
  enddo
  do j = 3,n-1
    C(m,j) = B(m,j)
    A(m,j) = - teta*dzavg(m+1)*B(m,j)-C(m,j)-D(m,j)-cdterm(m)
  enddo
  endif

* *** SANDFACE (INNER) BLOCKS

* *** Free Surface Equation

  altd(1) = aux1 / permx(2,1) * A1(1) / cosn(1,1)
  fval(1) = permv / permx(2,1) * A1(1)
  cdterm1(1) = aux2 * altd(1) * (dzd(1)+dh(1))

  B(1,1) = 0.d0
  C(1,1) = tet2 = fval(1) / (dh(1) + dzd(1))
  D(1,1) = 0.d0
  E(1,1) = teta*(hcd + dhavg(2))
  A(1,1) = - C(1,1)- E(1,1)- cdterm1(1)

  F(1,1) = altd(1)*(dh(1) - dzd(1))*sg - cdterm1(1) * wold(1,1)
  &           - tet1*((hcd + dzavg(2)) * (wold(2,1)-
  &           wold(1,1))-fval(1) * dzd(1)*(wold(1,1)-wold(1,2)))
  &

  c *** Second Layer Equation

  altd(1) = aux1 / permx(2,2) * A1(1) / cosn(1,2)
  fval(1) = permv / permx(2,2) * A1(1)
  cdterm(1) = altd(1) * dzd(1) * ctd

  B(1,2) = tet2 = fval(1) / (dh(1) + dzd(1))
  C(1,2) = teta *fval(1)/dzd(1)
  D(1,2) = 0.d0
  E(1,2) = teta*dzavg(2)
  A(1,2) = - teta *fx(1,2)*fr(1)*dzavg(1) - B(1,2) - C(1,2) - E(1,2)
  &           - cdterm(1)
  F(1,2) = -fx(1,2)*fr(1)*dzavg(1)*(teta*ww(2)-tet1*(wold(1,2)-
  &           . wwoold(2)) ) - cdterm(1) * wold(1,2)
  &           - tet1*((dzavg(2)*(wold(2,2)-wold(1,2))+fval(1)/dzd(1) *-
  &           (wold(1,1) - 2.d0*wold(1,2) + wold(1,3) ) ) )

  c *** Inner Boundary Inner Layer Equations

  do j = 2,n-1
    altd(1) = aux1 / permx(2,j) * A1(1) / cosn(1,j)
    fval(1) = permv / permx(2,j) * A1(1)
    cdterm(1) = altd(1) * dzd(1) * ctd

    B(1,j) = teta*fval(1)/dzd(1)
    D(1,j) = 0.d0
    E(1,j) = teta * dzavg(2)
    F(1,j) = -fx(1,j)*fr(1)*dzavg(j)*(teta*ww(j)-tet1*(wold(1,j)-
    &           . wwoold(j)) ) - cdterm(1) * wold(1,j)
    &           - tet1*((dzavg(2)*(wold(2,j)-wold(1,j))+fval(1)/dzd(1)*
    &           (wold(1,j-1) - 2.d0*wold(1,j) + wold(1,j+1) ) ) )
    &

    enddo
    do j = 3,n-1
      C(1,j) = B(1,j)
      A(1,j) = - teta *fx(1,j)*fr(1)*dzavg(j) - 2.d0*B(1,j) * E(1,j)
      &           - cdterm(1)
    enddo

  c *** Lower Inner Boundary Equation

  altd(1) = aux1 / permx(2,n) * A1(1) / cosn(1,n)
  fval(1) = permv / permx(2,n) * A1(1)
  cdterm(1) = altd(1) * dzd(1) * ctd

  B(1,n) = teta*fval(1)/dzd(1)
  C(1,n) = 0.d0
  D(1,n) = 0.d0

```

```

E(1,n) = teta*dzavg(2)
A(1,n) = -teta*fx(1,n)*fr(1)*dzavg(1)- B(1,n)-cdterm(1)

F(1,n) = -fx(1,n)*fr(1)*dzavg(1)*(teta*ww(n)-tet1*(wold(1,n)
& - wold(n)) ) - cdterm(1) * wold(1,n)
& - tet1*(dzavg(2)*(wold(2,n)-wold(1,n))+fval(1)/dzd(1)*
& (wold(1,n-1) - wold(1,n) ) )

return
end
=====
SUBROUTINE JACOBI (m,n,j1,xj,Bwb,ctd,cld,hwold,
& grado,lgchs,matcal,alfw,Swr,dww21)
* PARAMETER(mcm=80,ncm=100)
* JACOBIAN MATRIX CALCULATION

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

common /dato/poros,permh,permv,teta
common /dimlas/dt,dtb,hwd,hod,hs,dzdw,dhw,qd,red
common /alt/zd/mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
& hdavg(mcm+1),zdavg(mcm+1)
common /geom/u(mcm),rd(mcm+1),a1(mcm),fr(mcm),permx(mcm,ncm),
& rbar(mcm+1),fx(mcm,ncm),part(ncm),alf(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),ddz(mcm),dzavg(mcm+1),
& dhavg(mcm+1)
common /coef/A(mcm,ncm),B(mcm,ncm),C(mcm,ncm),D(mcm,ncm),
& E(mcm,ncm),F(mcm,ncm)
common /jacob/D(A(mcm,ncm),DB(mcm,ncm),DC(mcm,ncm),DD(mcm,ncm),
& DE(mcm,ncm),DF(mcm,ncm),DG(mcm,ncm))
common /grd/pd(mcm,ncm),grad(mcm)
common /deriv/dww(ncm,ncm)
common /angle/ cosn(mcm,ncm)

Sg = 1.d0 - Swr

if (lgchs .eq. 1) then
  if(ikkk .eq. 0 .and. td .ge. 10.) then
    print*, 'primeira vez que lgchs =1 apos td=10. '
  ikkk = 1
  endif
  alf1 = 0.5d0
else
  alf1 = alf(1)
endif
call wwder(n,j1,xj,dtb,poros,grado,bwb,qd,cld,permh,
& hwd,hwold,teta,alfw,DWW21)

c *** Calculation L.H.S. and R.H.S. vectors

do 2 j = 3,n-1
  do 1 i = 2,m-1

    DF(i,j) = A(i,j)*w(i,j)+B(i,j)*w(i,j-1)+C(i,j)*w(i,j+1)+
    & D(i,j)*w(i-1,j)+E(i,j)*w(i+1,j)-F(i,j)
    if (matcal .eq. 1) go to 1
    DA(i,j) = A(i,j)
    DB(i,j) = B(i,j)
    DC(i,j) = C(i,j)
    DD(i,j) = D(i,j)
    DE(i,j) = E(i,j)

1   continue
2   continue

* Free Surface
do 3 i = 2,m-1

  altd = hod * permh / permx(i+1,1) * A1(i) / (dtb*cosn(i,1))
  cdterm = altd * Ctd / 2.d0
  Cterm = hod * C(i,1)/(dh(i) + dzd(i))
  Cterm = 0.d0
  Dterm = teta*alf(i)*fx(i,1)*fr(i)*hod
  Eterm = teta * hod * (1.d0 - alf(i+1))

  DF(i,1) = A(i,1)*w(i,1)+C(i,1)*w(i,2)+
  & D(i,1)*w(i-1,1)+E(i,1)*w(i+1,1)-F(i,1)

  DA(i,1) = A(i,1) - (Dterm + Eterm + Cterm + cdterm) * w(i,1)
  & - Cterm * w(i,2) + Dterm * w(i-1,1) + Eterm* w(i+1,1)

```

```

&           - altd * Sg + cdterm * wold(i,1)

DB(i,1) = 0.d0
DC(i,1) = C(i,1)
DD(i,1) = D(i,1) + teta * hod * (1.d0 - alf(i)) * fx(i,1) *
&           fr(i) * (w(i-1,1) - w(i,1))
&           DE(i,1) = E(i,1) + hod * teta * (w(i+1,1) - w(i,1))

c *** Second Layer blocks

DF(i,2) = A(i,2)*w(i,2)+B(i,2)*w(i,1)+C(i,2)*w(i,3)+
&           D(i,2)*w(i-1,2)+E(i,2)*w(i+1,2)-F(i,2)
&           DA(i,2) = A(i,2)
DB(i,2) = B(i,2)*(1.d0 - hod * (w(i,1)-w(i,2))/(dzd(i)+dh(i)))
DC(i,2) = C(i,2)
DD(i,2) = D(i,2)
DE(i,2) = E(i,2)

3 continue

c *** Free Surface Inner Block

cte = 2.d0 * poros * dtd

altd = hod * permh / permx(2,1) * A1(1) / (dtd*cosn(1,1))
cdterm = altd * Ctd / 2.d0
Cterm = hod * C(1,1)/(dh(1) + dzd(1))
Dterm = teta*alf(1)*fx(1,1)*fr(1)*hod
Eterm = teta * hod * (1.d0 - alf(2))

DF(1,1) = A(1,1)*w(1,1)+C(1,1)*w(1,2)+E(1,1)*w(2,1)-F(1,1)
DA(1,1) = A(1,1) - (Eterm - Cterm + cdterm) * w(1,1)
&           - Cterm * w(1,2) + Eterm * w(2,1) - altd * Sg +
&           cdterm * wold(1,1)
DB(1,1) = 0.d0
DC(1,1) = C(1,1)
DD(1,1) = 0.d0
DE(1,1) = E(1,1) * teta * hod * alf(2) * (w(1,1) - w(2,1))

* Second Layer IBC

par = fx(1,2)*fr(1)*teta / 2.d0
DF(1,2) = A(1,2)*w(1,2)+B(1,2)*w(1,1)+C(1,2)*w(1,3)+E(1,2)*w(2,2)
&           - F(1,2)

DA(1,2) = A(1,2) + teta * fx(1,2)*fr(1) * dzavg(1) * dww(2,2)
DB(1,2) = B(1,2) * (1.d0 - hod * (w(1,1)-w(1,2))/(dzd(1)+dh(1)) )
&           + teta * fx(1,2)*fr(1) * dzavg(1) * dww(2,1)
DC(1,2) = C(1,2) + teta * fx(1,2)*fr(1) * dzavg(1) * dww(2,3)
DD(1,2) = 0.d0
DE(1,2) = E(1,2)

* Free Surface OBC

altd = hod * A1(m) / dtd
cdterm = altd * Ctd / 2.d0
Cterm = hod * C(m,1) / (dh(m) + dzd(m))
Dterm = teta * hod * fx(m,1) * fr(m)
Eterm = teta * hod * (1.d0 - alf(m+1))

DF(m,1) = A(m,1)*w(m,1)+C(m,1)*w(m,2)+D(m,1)*w(m-1,1)-F(m,1)
DA(m,1) = A(m,1) - (Dterm + Eterm - Cterm + cdterm) * w(m,1)
&           - Cterm * w(m,2) + Dterm * w(m-1,1) + Eterm
&           - altd * Sg + cdterm * wold(m,1)

DB(m,1) = 0.d0
DC(m,1) = C(m,1)
DD(m,1) = D(m,1) + teta * hod * (1.d0 - alf(m)) * fx(m,1)*fr(m) *
&           (w(m-1,1)-w(m,1))
DE(m,1) = 0.d0

* Second Layer OBC

DF(m,2) = A(m,2)*w(m,2)+B(m,2)*w(m,1)+C(m,2)*w(m,3)+
&           D(m,2)*w(m-1,2)-F(m,2)
DA(m,2) = A(m,2)
DB(m,2) = B(m,2) * (1.d0 - hod*(w(m,1)-w(m,2))/(dzd(m)+dh(m)))
DC(m,2) = C(m,2)

```

```

DD(m,2) = D(m,2)
DE(m,2) = 0.d0
do 4 j = 3,n-1
*      Saturated Region IBC
DF(1,j) = A(1,j)*w(1,j)+B(1,j)*w(1,j-1)+C(1,j)*w(1,j+1)+E(1,j)
&      *w(2,j)-F(1,j)
par = teta * fx(1,j) * fr(1) * dzavg(1)
DA(1,j) = A(1,j) + par*dww(j,j)
DB(1,j) = B(1,j) + par*dww(j,j-1)
DC(1,j) = C(1,j) + par*dww(j,j+1)
DD(1,j) = 0.d0
DE(1,j) = E(1,j)
*      OBC Saturated Region
DF(m,j) = A(m,j)*w(m,j)+B(m,j)*w(m,j-1)+C(m,j)*w(m,j+1)+D(m,j)
&      *w(m-1,j)-F(m,j)
if (matcal.eq. 1) go to 4
DA(m,j) = A(m,j)
DB(m,j) = B(m,j)
DC(m,j) = C(m,j)
DD(m,j) = D(m,j)
DE(m,j) = 0.d0
4  continue
*      IBC Layer N
par = teta * fx(1,n) * fr(1) * dzavg(1)
DA(1,n) = A(1,n) + par * dww(n,n)
DB(1,n) = B(1,n) + par * dww(n,n-1)
DC(1,n) = 0.d0
DD(1,n) = 0.d0
DE(1,n) = E(1,n)
DF(1,n) = A(1,n)*w(1,n)+B(1,n)*w(1,n-1)+E(1,n)*w(2,n)-F(1,n)
if (matcal.eq. 1) go to 400
*      Layer N
do 5 i = 2,m
DA(i,n) = A(i,n)
DB(i,n) = B(i,n)
DC(i,n) = 0.d0
DD(i,n) = D(i,n)
DE(i,n) = E(i,n)
5  continue
400 DF(m,n) = A(m,n)*w(m,n)+B(m,n)*w(m,n-1)+D(m,n)*w(m-1,n)-F(m,n)
do 6 i = 2,m-1
DF(i,n) = A(i,n)*w(i,n)+B(i,n)*w(i,n-1)+D(i,n)*w(i-1,n)+E(i,n)*
&      w(i+1,n)-F(i,n)
6  continue
*      Derivatives of Ww(j) w.r.t. w(1,k), to fill up the matrix.
do 7 j = 1,n
if (j.eq. 1) then
par = teta * fx(1,1)*fr(1) * dhavg(1)
else
par = teta * fx(1,j)*fr(1) * dzavg(1)
endif
do 7 k = 2,n
if (j.lt. j1 ) then
DG(j,k) = 0.d0
else
DG(j,k) = par * dww(j,k)
endif
7  continue
if (j.lt. j1) then
DG(j,1) = 0.d0
else
DG(j,1) = par * dww(j,1)
endif
8  continue
DG(1,1) = 0.d0
return
end

```



```

subroutine solve(m,n,ier,ideg)
=====
===== Subroutine SOLVE provides the interface between the simulator =====
===== and a linear solver package. =====
=====
===== It puts the matrix coefficients in a compressed storage scheme, =====
===== allocates workspace for the iterative solver and calls MSILUM, =====
===== which performs incomplete factorization, and MSORMN, which does =====
===== ORTHOMIN iterations. =====
===== Clements Brand, Aug 1991 =====
=====
The matrix is" apart from modifications at the boundaries" a
five-diagonal banded matrix. For each block, the coefficients are
stored in the following way
" " " " "
DD --- DB - DA - DC --- DE
" " " " "
DG are coefficients in the first NxN submatrix, DF is the right-
hand side.
Compressed storage mode:
A( ) holds all nonzero matrix coefficients in consecutive order,
column index runs first, row index second.
IA( ) holds the column index of the corresponding element in A( )
NA( ) points to the first element of a row. For example, the
nonzero coefficients of row K would be stored on
A(I), I=NA(K),...,NA(K+1)-1
and their column indices would be stored on
A22,...,A2N,A2 N+2,A31,A32,...,A3N,A3 N
with A11, then
=====
PARAMETER(mcm=80,ncm=100)
implicit real*8(a-h,o-z)
=====
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /jacob/DA(mcm,ncm),DB(mcm,ncm),DC(mcm,ncm),DD(mcm,ncm),
& DE(mcm,ncm),DF(mcm,ncm),DG(ncm,ncm)
=====
parameter ( kadim = ncm*ncm+5*ncm*mcm
1      , neqdim = ncm*mcm
2      , iwkdim = 3*kadim+15*neqdim)
=====
.....make sure that kadim and neqdim are even numbers!
common/solwrk/ a(kadim),ia(kadim),na(neqdim+2),work(iwkdim)
1      , r(neqdim), x(neqdim)
=====
data iswi,nold,mold/1,0,0/
save iswi,nold,mold
=====
additional input parameters for the iterative solver
degree of factorization, number of orthogonalisations.
stopping criterion, maximum number of iterations
ideg = 1
norio = 5
errlim = 1 e-7
errlim = 1 e-6
itmax = 30
iout = 0
neq = n*m
=====
=====
put matrix elements into sparse matrix storage scheme
=====
irow = 0
icnt = 0
=====
rows 1 to n
do 100 j=1,n
  irow = irow+1
  na(irow) = icnt+1
  idiag = icnt+j
  do 110 k=1,n
    icnt=icnt+1
    a(icnt) = dg(j,k)
    ia(icnt) = k
110  continue
    if(j.ne 1)
1      a(idiag-1) = db(1,j)
      a(idiag) = da(1,j)
      if(j.ne n)
1      a(idiag+1) = dc(1,j)
  icnt = icnt+1
100

```

```

      a(icnt)      = de(1,j)
      ia(icnt) = irow+n
      **** right-hand-side
      r(irow)      = df(1,j)
      100 continue
      **** rows n+1 to m=n
      do 200 i=2,m
      do 200 j=1,n
         irow = irow+1
         na(irow) = icnt+1
         icnt = icnt+1
         a(icnt)      = dd(i,j)
         ia(icnt)      = irow-n
         if(j.ne.1) then
            icnt = icnt+1
            a(icnt)      = db(i,j)
            ia(icnt)      = irow-1
         end if
         icnt = icnt+1
         a(icnt)      = da(i,j)
         ia(icnt)      = irow
         if(j.ne.n) then
            icnt = icnt+1
            a(icnt)      = dc(i,j)
            ia(icnt)      = irow+1
         end if
         if(i.ne.m) then
            icnt = icnt+1
            a(icnt)      = de(i,j)
            ia(icnt)      = irow+n
         end if
      **** right-hand-side
      r(irow)      = df(i,j)
      200 continue
      na(irow+1)=icnt+1
      write(6,'(10e8.1)') ((dg(i,j),j=1,70),i=69,70)
      do i=1,3
      write(6,'(5e16.5)') (dd(i,j),db(i,j),da(i,j),
      1           dc(i,j),de(i,j),j=1,n)
      end do
      **** allocate the workspace and check the dimensions
      ****
      **** number of nonzero elements in the lower and upper triangular factors
      ludim = ideg*icnt/2
      this formula is a rather coarse estimate- e.g. for ndeg=1,
      ludim = (n*n-n)/2+(2*n-1)*(m-1)
      should be the exact value
      **** pointers for the arrays storing the L and U factors
      - u - i - iu - il - nu - nl - du -
      i4  i5  i6  i7  i8  i9  i10 i11
      i4 = 1
      i5 = i4 + ludim
      i6 = i5 + ludim
      i7 = i6 + (ludim/2 + 2)
      i8 = i7 + (ludim/2 + 2)
      i9 = i8 + neq/2+2
      i10= i9 + neq/2+2
      i11= i10+ neq
      **** pointers for work arrays for the ILU part
      - xhelp - ihelp - nohelp - iuloc - ieli -
      i11  i12  i13  i14  i15  i16
      i12 = i11 + neq
      i13 = i12 + neq/2+1
      i14 = i13 + ideg*(neq/2+1)
      i15 = i14 + neq/2+1
      i16 = i15 + neq/2+1
      **** pointers for the ORTHOMIN work arrays
      they share space with some of the solver arrays
      - adx - dx - adxor - dxor -
      k1  k2  k3  k4  k5

```

```

*      k1 = i11
*      k2 = k1 + neq
*      k3 = k2 + neq
*      k4 = k3 + neq*(nerto-1)
*      k5 = k4 + neq*(nerto-1)
*
*  *** diagnostic messages
*      iwk = max(i16,k5)
*      if(iwk.ge.iwkdim) then
*          write(*,*) ' NOT ENOUGH WORKSPACE FOR THE SOLVER.'
*          write(*,*) ' REQUIRED: iwk, ACTUAL VALUE IWKDIM: ', iwkdim
*          write(*,*) ' INCREASE DIMENSION OF WORK(IWKDIM)'
*          stop
*      end if
*
*  *** select incomplete factorization routine
*  *** iawi = 1 MSILUM...ILU factorization including symbolic fact.
*  *** iawi = 2 MSILUP...ILU factorization using previous symbolic fact.
*
*      iawi = 1
*      if(n.eq.nold .and. m.eq.mold) iawi=2
*      nold = n
*      mold = m
*
*      if(iawi.eq.1) then
*          call msilum( neq
*          1      , a      , ia      , na      , kdim
*          2      , work(i5), work(i7), work(i9), ludim
*          ***   1      il      nl      kldim
*          3      , work(i4), work(i6), work(i8), work(i10), ludim
*          ***   u      iu      nu      du      kudim
*          4      , ideg
*          ***   norder
*          5      , work(i11),work(i12),work(i13),work(i14),work(i15)
*          ***   xhelp  ihelp  nohelp  iulo"  ieli
*          6      , ierr)
*
*      else if(iawi.eq.2) then
*          call msilup( neq
*          1      , a      , ia      , na      , kdim
*          2      , work(i5), work(i7), work(i9), ludim
*          ***   1      il      nl      kldim
*          3      , work(i4), work(i6), work(i8), work(i10), ludim
*          ***   u      iu      nu      du      kudim
*          4      , ideg
*          ***   norder
*          5      , work(i11),work(i12),work(i13),work(i14),work(i15)
*          ***   xhelp  ihelp  nohelp  iulo"  ieli
*          6      , ierr)
*
*      end if
*
*  *** diagnostic messages
*      call esgetdim(neq,na,work(i8),work(i9),iad,iud,iid)
*      if(iud.gt.ludim .or. iid.gt.ludim) ierr=2
*      write(*,*) neq,' equations.'
*      write(*,*) 'Requested dimensions: a ', iad
*      write(*,*) '                                u ', iud
*      write(*,*) '                                l ', iid
*
*      if(ierr.eq.1) then
*          write(*,*) ' WARNING - NDEG exceeds LOCDI1 in routine MSILUN'
*          write(*,*) ' NDEG is reduced to ',ideg
*      else if(ierr.eq.2 .or. ierr.eq.3) then
*          write(*,*) ' FATAL - not enough workspace for the '
*          write(*,*) ' preconditioning matrix.'
*          write(*,*) ' Increase LUDIM; actual value = ',ludim
*          write(*,*) ' required ',max(iud,iid)
*      end if
*      ierr = 0
*
*  *** perform the ORTHOMIN iterations
*  *** -----
*
*      do 401 i=1,neq
*          x(i)=0
* 401      continue
*
*      — adx — dx — adxor — dxor —

```

```

*   k1   k2   k3   k4   k5
*
*   call msormn( neq
*   1      , a      , ia      , na      , kdim
*   2      , work(i5), work(i7), work(i9), ludim
*   ***   , l      , il      , nl      , kldim
*   3      , work(i4), work(i6), work(i8), work(i10), ludim
*   ***   , u      , iu      , nu      , du      , kudim
*   4      , r      , x      , work(k1), work(k2), work(k3), work(k4)
*   ***   , r      , x      , adx      , dx      , adxor      , dxor
*   5      , norto, errlim, itmax, nitr, iout
*   6      , ierr )
*
*   if(ierr.ge.3) then
*      write(7,*) 'WARNING - no convergence in ORTHOMIN after '
*      write(7,*) '           ,itmax,' iterations.'
*   end if
*
*   call prinzm(neq,work(i2),na,work(i7),work(i9),
*   1           work(i8),work(i8),work(i10) )
*
*   **** update the solution ****
*
*   irow = 0
*   do 500 i = 1,m
*   do 500 j = 1,n
*      irow = irow+1
*      w(i,j) = w(i,j) - x(irow)
* 500 continue
*
*   return
*   end
*
*   subroutine prinzm(neq, ia, na, il, nl, iu, nu, help)
*   dimension ia(*), na(*), il(*), nl(*), iu(*), nu(*), help(*)
*   open (15,form='formatted',file='iludat')
*   write(15,*)
*   do 201 j=1,neq
*      help(j) = 0
*      do 202 i=1,neq
*         do 203 k=nu(i),nu(i+1)-1
*            help(iu(k)) = 0.5
*            do 204 k=nl(i),nl(i+1)-1
*               help(il(k)) = 0.5
*               do 205 k=na(i),na(i+1)-1
*                  help(ia(k)) = 1.
*                  do 206 j=1,neq
*                     write(15,*) help(j)
*                  help(j) = 0
* 206   continue
* 202 continue
*
*   close(15)
*   return
*   end
*
*   C
*   C
*   subroutine msilum( neq
*   1      , a, ia, na, kdim
*   2      , l, il, nl, kldim
*   3      , u, iu, nu, du, kudim
*   4      , norder
*   5      , xhelp, ihelp, nohelp, iuloc, ieli
*   6      , ierr)
*
*   **** I T E R A T I V E   M A T R I X - S O L V E R ****
*
*   Routine MSILUM: incomplete LU-factorization of order NORDER
*
*   For a sparse matrix A, an approximate decomposition into a
*   product L.U is constructed, where L is a lower and U is an
*   upper triangular matrix. The sparsity pattern of L and U is
*   controlled by the order of admissible fill-in terms.
*
*   As opposed to routine MSILUN, this routine assumes that the
*   elements for each matrix row are stored in increasing order
*   w.r.t column index
*
*   Input (unchanged on exit):

```

```

    NEQ
    number of equations and unknowns.

    KADIM, KUDIM, KLDIM
    dimensions of the corresponding arrays.

    A(KADIM), IA(KADIM), NA(NEQ+1)
    non-zero matrix elements of A in compressed storage format
    (Yale sparse-matrix format).

    Norder
    maximum order of admissible fill-in terms.

    -----
    Input (scratch arrays)
    -----
    XHELP(NEQ), IHELP(NEQ), NOHELP(NORDER,NEQ), IULOC(NEQ), IELI(NEQ)

    -----
    Output
    -----
    L(KLDIM), IL(KLDIM), NL(NEQ+1)
    matrix elements of L in a compressed storage format
    (negative values of non-zero matrix elements below the
    main diagonal of L in compressed storage format).

    U(KUDIM), IU(KUDIM), NU(NEQ), DU(NEQ)
    matrix elements of U in a compressed storage format
    (reciprocal diagonal elements are stored in DU, negative values of
    non-zero elements above the main diagonal are stored on U, IU, NU).

    IERR
    0 . . . normal successful completion
    1 . . . NORDER exceeds LOCIDL (warning).
    Program re-sets NORDER to LOCIDL and continues. Be aware,
    it doesn't make much sense to choose a too great value of
    NORDER. It is, however, possible to increase LOCIDL by
    changing the PARAMETER statement below.
    2 . . . KLDIM exceeded (fatal). Increase array size for L and IL
    or try a lower NORDER.
    3 . . . KUDIM exceeded (fatal). Increase array size for U and IU
    or try a lower NORDER.

    -----
    implicit double precision (a-h,o-z)
    dimension akadim, ia(kadim), na(neq+1)
    1 . . . l(kldim), il(kldim), nl(neq+1)
    2 . . . u(kudim), iu(kudim), nu(neq), du(neq)
    real l
    double precision l

    *** scratch arrays
    dimension xhelp(neq), ihelp(neq), nohelp(norder,neq)
    1 . . . iuloc(neq), ieli(neq)

    *** small local arrays
    parameter( locdil = 10)
    dimension ndgloc(locdil), iii(locdil)

    ***
    *** check and initialize
    ***

    ierr = 0

    *** check local array dimensions
    if( norder.gt.locdil) then
        ierr = 1
        norder = locdil
    end if

    *** initialize ihelp
    do 100 i=1,neq
        ihelp(i) = 99
100    continue

```

```

* *** set output control flag (for debugging purposes only)
cc"      ichk = 0
cc"      if(ierr.eq.-1) ichk = 1
*
* ***
* *** special treatment of first row
* ***
*
* *** lower matrix: no elements stored for first row
    nl(1)=1
    ilcnt = 0
*
* *** upper matrix: elements of A transferred to U
    du(1) = 1./a(na(1))
    nu(1) = 1
    do 200 j=na(1)+1,na(2)-1
        u(j-1) = -a(j)
        iu(j-1) = ia(j)
200  continue
    iucnt = na(2)-na(1)-1
*
* *** pointers referring to the degree of fill-in terms
    nohelp(1,1) = na(2)-na(1)
    do 250 j=2,norder
        nohelp(j,1) = 0
250  continue
*
* ***
* *** loop for rows 2,...,NEQ of the matrix
* ***
*
* ***
* do 300 ilin = 2,neq
*
*
* ***
* *** set counters and pointers
* ***
    neli = 0
    nulli = 0
    jelii = 0
    nu(ilin) = iucnt+1
*
    do 305 i = 1,norder
        ndgloc(i)=0
305  continue
*
* ***
* *** transfer matrix coefficients from compressed to full storage mode
* ***
*
    do 310 j=na(ilin),na(ilin+1)-1
        k = ia(j)
        xhelp(k) = a(j)
        ihelp(k) = 1
*
        if(k.lt.ilin) then
* *** count elements left of the diagonal, store their position
            neli = neli+1
            ieli(neli) = k
        else if(k.gt.ilin) then
* *** count elements right from the diagonal, store their position
            nulli = nulli+1
            ndgloc(1) = ndgloc(1)+1
            iuloc(nulli) = k
        end if
310  continue
*
*
* if(ichk.eq.1) then
*     write(6,*) 'IHELP '
*     write(6,97) (ihelp(j),j=1,neq)
c97  format(40i2)
* end if
*
* ***
* *** eliminate elements left to the diagonal element
* ***
*
* do while (jeli.le.neli) ;** coded as a goto construct
2000 continue

```

```

jeli = jeli+1
if(jeli.gt.neli) goto 2999

* *** get and store pivot value
jpivot = ieli(jeli)
jucnt = nu(jpivot)-1
xpiv = xhelp(jpivot)*du(jpivot)
xhelp(jpivot) = -xpiv

* ***
* **** control incomplete factorization
* ****
* ***
* degree of element to be eliminated
jdeg = ihelp(jpivot)

* ***
* *** first case: degree of element in pivot row plus degree of
* *** element to be eliminated does not exceed norder
* ***

do 330 id = jdeg+1,norder
nd = nohelp(id-jdeg,jpivot)
cc*   write(6,*) 'jucnt, nd', jucnt, nd
cc*   write(6,*) '(u(k),k=jucnt+1,jucnt+nd)'
do 331 k=jucnt+1,jucnt+nd

  kk = iu(k)
  ihkk = ihelp(kk)
  if(ihkk.le.id) then
    * not new entry, no change in degree
    xhelp(kk) = xhelp(kk) + xpiv*u(k)
  else if(ihkk.gt.norder) then
    * new entry
    xhelp(kk) = xpiv*u(k)
    ihelp(kk) = id
    if(kk.lt.ilin) then
      insert kk at the correct position in ieli
      do 3311 j=neli,jeli+1,-1
      if(kk.le.eli(j)) then
        ieli(j+1) = ieli(j)
      else
        ieli(j+1)=kk
        goto 3319
      end if
    3311  continue
    ieli(jeli+1)=kk
  3319  neli = neli+1
  else
    nuli = nuli+1
    iuloc(nuli) = kk
    ndgloc(id) = ndgloc(id)+1
  end if

  else
    * not a new entry, but the degree changes
    xhelp(kk) = xhelp(kk) + xpiv*u(k)
    if(kk.gt.ilin) then
      ndgloc(ihkk) = ndgloc(ihkk)-1
      ndgloc(id) = ndgloc(id)+1
    end if
    ihelp(kk) = id
  end if

331  continue
jucnt = jucnt+nd
330  continue

* ***
* *** second case: degree of element in pivot row plus degree of
* *** element to be eliminated exceeds norder
* ***

do 340 kk=jucnt+1,nu(jpivot+1)-1
  kk = iu(k)
  xhelp(kk) = xhelp(kk) + xpiv*u(k)
  (only VALID entries will be updated after completion)
340  continue

goto 2000
2999 continue

```

```

      *
      *-----.
      * *** sparse elimination in row ilin now complete. collect coefficients
      *-----.
      *
      * *** scan IELI: reset IHHELP and store L-matrix in condensed form
      *
      nl(ilin) = ilcnt+1
      do 350 j=1,neli
         jj = ieli(j)
         ihelp(jj) = 99
         ilcnt = ilcnt+1
         l(ilcnt) = xhelp(jj)
         il(ilcnt) = jj
      350  continue
      *
      * *** scan IULOC
      *
      c**  set pointers to coefficients w.r.t. order
      iii(1) = iucnt
      do 355 j=2,norder
         iii(j) = iii(j-1)+ndgloc(j-1)
      355  continue
      *
      if(abs(xhelp(ilin)) .lt. 1.e-20) xhelp(ilin) = 1.e-20
      du(ilin) = 1./xhelp(ilin)
      ihelp(ilin) = 99
      *
      c**  fill up arrays u and iu
      do 360 j=1,nuli
         jj = iuloc(j)
         ihjj = ihelp(jj)
         iii(ihjj) = iii(ihjj)+1
         u(iii(ihjj)) = -xhelp(jj)
         iu(iii(ihjj)) = iuloc(j)
      c**      xhelp(jj) = 0
      ihelp(jj) = 99
      360  continue
      iucnt = iucnt+nuli
      *
      do 365 j=1,norder
         nohelp(j,ilin) = ndgloc(j)
      365  continue
      *
      300  continue
      *
      *-----.
      * *** loop ilin=2,neq now complete
      *-----.
      *
      *
      nl(neq+1) = ilcnt+1
      *
      write(*,*) na(neq+1), nl(neq+1), nu(neq)
      if(ichk.eq.1) then
         write(6,*)' die u-matrix'
         j0 = nu(1)
         do i=1,neq-1
            j1 = nu(i+1)
            write(6,99) 1./du(i), (u(j),j=j0,j1-1)
            write(6,98) i,(iu(j),j=j0,j1-1)
            j0 = j1
         end do
         i = neq
         write(6,99) 1./du(i)
         write(6,98) i
      end if
      *
      if(ichk.eq.1) then
         write(6,*)' die l-matrix '
         j0 = nl(1)
         do i=1,neq
            j1 = nl(i+1)
            write(6,99) (l(j),j=j0,j1-1)
            write(6,98) (il(j),j=j0,j1-1)
            j0 = j1
         end do
      end if
      c99  format(1h ,10e10.3)
      c98  format(1h ,10i10)
      *
      99999 continue

```

```

      return
    end
  subroutine mailup( neq
  1      , a, ia, na, kdim
  2      , l, il, nl, kldim
  3      , u, iu, nu, du, kudim
  4      , norder
  5      , xhelp, ihelp, nohelp, iuloc, ieli
  6      , ierr)
  -----
*   I T E R A T I V E   M A T R I X - S O L V E R
  -----
*   Routine MSILUP: incomplete LU-factorization of order NORDER
*   Clemens Brand, November 1990
*
*   For a sparse matrix A, an approximate decomposition into a
*   product L.U is constructed, where L is a lower and U is an
*   upper triangular matrix. The sparsity pattern of L and U is
*   prescribed in advance, using information from a previous call
*   to the solver
*
*   As opposed to routine MSILUN, this routine assumes that the
*   fill-in pattern is known in advance and given by the arrays
*   il, nl, iu and nu
  -----
*   unnecessary arrays which are kept in the parameterlist for
*   compatibility are ihelp, nohelp, iuloc and ieli
  -----
*
      implicit double precision (a,h,o,z)
      dimension a(kdim), ia(kdim), na(neq+1)
  1      , l(kldim), il(kldim), nl(neq+1)
  2      , u(kudim), iu(kudim), nu(neq), du(neq)
      real l
      double precision l
      dimension xhelp(neq)
  -----
*   *** check and initialize
  -----
*
      ierr = 0
*
*   *** special treatment of first row
*   *** lower matrix: no elements stored for first row
*   *** upper matrix: elements of A transferred to U
      du(1) = 1./a(1)
      do 200 j=na(1)+1,na(2)-1
      u(j-1) = -a(j)
  200  continue
  -----
*   *** loop for rows 2,...,NEQ of the matrix
  -----
      do 300 ilin = 2,neq
*
*   *** clear array xhelp and
*   *** transfer matrix coefficients from compressed to full storage mode
      do 310 j=nl(ilin),nl(ilin+1)-1
      jj = il(j)
      xhelp(jj)=0
  310  continue
      if(ilin.lt.neq) then
      do 320 j=nu(ilin),nu(ilin+1)-1
      jj = iu(j)
      xhelp(jj) = 0
  320  continue
      end if
*
      do 330 j=na(ilin),na(ilin+1)-1
      k = ia(j)
      xhelp(k) = a(j)
  330  continue
*
*   *** eliminate elements left to the diagonal element.
*   ***
      do 2999 jeli=nl(ilin),nl(ilin+1)-1
*   *** get and store pivot value
      jpiv = il(jeli)
      xpiv = xhelp(jpiv)*du(jpiv)

```

```

        xhelp(jpiv) = -xpiv
      *
      * *** control incomplete factorization
      *
      juent = nu(jpiv)
      do 340 k=juent,nu(jpiv+1)-1
        kk = iu(k)
        xhelp(kk) = xhelp(kk) + xpiv*u(k)
      * (only VALID entries will be updated after completion)
      340  continue
      *
      2999  continue
      *
      * *** sparse elimination in row ilin now complete. collect coefficients
      *
      * *** scan IL,IU and store L- and U-matrix in condensed form
      *
      do 350 j=nl(ilin),nl(ilin+1)-1
        jj = il(j)
        l(j) = xhelp(jj)
      350  continue
      *
      du(il,n) = 1./xhelp(ilin)
      if(ilin.lt.neg) then
        do 360 j=nu(ilin),nu(ilin+1)-1
          jj = iu(j)
          u(j) = -xhelp(jj)
        360  continue
        end if
      *
      300  continue
      *
      * *** loop ilin=2,neq now complete.
      *
      *
      * if(ichk.eq.1) then
      *   write(6,*) ' die u-matrix'
      *   j0 = nu(1)
      *   do i=1,neq-1
      *     j1 = nu(i+1)
      *     write(6,99) 1./du(i), (u(j),j=j0,j1-1)
      *     write(6,98) i,(iu(j),j=j0,j1-1)
      *   j0 = j1
      *   end do
      *   i = neq
      *   write(6,99) 1./du(i)
      *   write(6,98) i
      *   end if
      *
      * if(ichk eq 1) then
      *   write(6,*) ' die l-matrix '
      *   j0 = nl(1)
      *   do i=1,neq
      *     j1 = nl(i+1)
      *     write(6,99) (l(j),j=j0,j1-1)
      *     write(6,98) (il(i),j=j0,j1-1)
      *   j0 = j1
      *   end do
      *   end if
      c99  format(1h ,10e10.3)
      c98  format(1h ,10i10)
      *
      99999 continue
      return
      end
      subroutine msormn( neq
      1      , a, ia, na, kadm
      2      , l, il, nl, kldim
      3      , u, iu, nu, du, kudim
      4      , r, x, adx, dx, adxor, dxor
      5      , norto, errlim, itmax, nitr, iout
      6      , ierr)
      *
      * iterative solver
      *
      * iteration with orthomin convergence acceleration
      *

```

```

=====
*
* iterative solution of a linear system
*
*      a . x = r
*
* using the generalized conjugate gradient technique orthomin.
*
* as a preconditioning, the incomplete factorization of a into
* a product l . u , is used. the set-up of the triangular matrices
* l and u is done in subroutine orilu.
*
=====
*
* EHRORcode ierr
*      ierr=0 means normal successful completion
*      ierr=2 norto, number of orthogonalizations, exceeds
*             local array dimension locd1
*      ierr=3 maximum number of iterations exceeded
*      ierr=4 cnorm almost 0, no search direction
*             can be found. Improve preconditioning
*             and start again
*      ierr=5 maximum number of iterations exceeded, however,
*             a moderate reduction of the initial residual
*             has been obtained
*
=====
*
* implicit double precision (a-h, o-z)
* double precision l
* dimension a(kadim), ia(kadim), na(neq+1)
* 1      , il(kldim), il(kldim), nl(neq+1)
* 2      , u(kudim), iu(kudim), nu(neq), du(neq)
* real l
*
* dimension r(neq), x(neq)
*
* *** scratch arrays
*
* dimension adx(neq), dx(neq)
* 1      , adxor(neq,*), dxor(neq,*)
*
* *** local arrays.
*
* parameter (locd1=20)
* dimension cnorm(locd1)
*
=====
*
* ierr=0
* *** check norto (norto-1 is the maximum number of previous residual
* *** vectors that are stored and used for orthogonalization)
*
*      if(norto.gt.locd1) then
*         ierr = 2
*         norto = locd1
*      end if
*
* *** initialize residual vector, compute initial residual
* ***
*
*      call msmvml(neq, a, ia, na, x, adx)
*      resini = 0
*      do 300 i=1,neq
*         r(i) = r(i) - adx(i)
*         resini = resini + r(i)*r(i)
* 300   continue
*      resini = sqrt(resini)
*
*      if(iout.gt.0)
* 1      write(iout,'(r8.4)') 'rms init ', real(resini)
*
* ***
* =====
*      start of the orthomin iteration loop
* =====
*
*      nor = 1
*      icnt = 1
* 400   continue
*

```

```

      .
      .
      .
      " solve the system L.U * DX = R
      .
      .
      .
      " call mslusv(neq, l, ll, nl, kldim, u, iu, nu, du, kudim, dx, r)
      .
      .
      .
      " matrix multiplication: ADX = A*DX
      .
      .
      .
      " call msmvml(neq, a, ia, na, dx, adx)
      .
      xerr = 0
      do 409 i=1,neq
      xerr = max(xerr, abs(dx(i)))
409  continue
      .
      .
      .
      " orthogonalization part. ADX is orthogonalized regarding (NOR-1)
      " previously computed vectors (which are stored on ADXOR)
      .
      do 410 ior=1,nor-1
      .
      .
      .
      " compute inner product
      .
      sum = 0
      do 411 i=1,neq
      sum = sum + adxor(i,ior)*adx(i)
411  continue
      sum = -sum/cnorm(ior)
      .
      .
      .
      " add correction for orthogonalization
      .
      do 412 i=1,neq
      adx(i) = adx(i) + sum*adx(i,ior)
      dx(i) = dx(i) + sum*dx(i,ior)
412  continue
410  continue
      .
      .
      .
      " minimization constants
      .
      .
      .
      cnorm(nor) = 0
      omega = 0
      .
      do 430 i=1,neq
      cnorm(nor) = cnorm(nor)+adx(i)*adx(i)
      omega = omega + r(i)*adx(i)
430  continue
      if(cnorm(nor).lt.1.e-30) then
      if(omega.lt.10.e-30) then
      .... not really a problem, just dealing with small numbers
      omega = 1.
      else
      .... the new search direction is bad
      ierr = 4
      return
      end if
      else
      .... this is the regular case
      omega = omega/cnorm(nor)
      end if
      .
      .
      .
      " update solution and residual
      .
      .
      .
      rnorm = 0.d0
      oma = max(1.d0, abs(omega))
      do 440 i=1,neq
      .
      .
      .

```

```

x(i) = x(i)+omega*dx(i)
r(i) = r(i)-omega*adx(i)
rnorm = rnorm+r(i)*r(i)
tmp = abs( omega*dx(i) )
xerr = max(xerr,tmp)
rnorm = max(rnorm,abs(r(i)))
440 continue
rnorm = sqrt(rnorm)
*
*
cc"  xnorm = 0
cc"  errmax = 0
cc"  do 460 i=1,nqe
cc"  errmax = max(abs(real(r(i))),errmax)
cc"  xnorm = xnorm+x(i)*x(i)
460 continue
*
if(iout.gt.0)
1  write(iout,"(i1,real(omega),real(rnorm),
2           ,real(rnorm/resini),real(xerr))
*
*-----*
* convergence check
*-----*
*----*
*----* check convergence condition; return (successful completion)
*----*
*----* if(rnorm .lt. errlim*resini
*----*      ) then
*----*      1 .or. xerr.lt.1.d-6) then
*----*          nitr = icnt
*----*          return
*----*      end if
*----*
*----* check for maximum number of iterations; return (ERROR status).
*----*
if(icnt .ge. itmax) then
  nitr = icnt
  ierr = 3
  if(rnorm.lt.sqrt(errlim)*resini
  1 .and. xerr.lt.10) ierr=5
  return
end if
*
*-----*
* go and give it another try
*-----*
*
*----*
*----* increment iteration counter
*----*
icnt = icnt+1
*
*----*
*----* restart the orthogonalization sequence or add new vector
*----* to the orthogonal system
*----*
if (nor.eq.norto) then
  nor = 1
else
  do 490 i=1,nqe
    dxor(i,nor) = dx(i)
    adxor(i,nor) = adx(i)
490 continue
  nor = nor+1
end if
*
goto 400
end
subroutine msmvml(neq, a, ia, na, x, ax)
*-----*
*-----*
*-----* M S M V M L
*-----*
*-----* performs matrix-vector-multiplication
*-----*      ax = a.x
*-----*      == == ==
*-----*
implicit double precision (a-h, o-z)
dimension a(*), ia(*), na(neq+1)
dimension x(neq), ax(neq)

```

```

j1 = 1
do 1 i=1,neq
j0 = j1
j1 = na(i+1)
ax(i) = 0
1 do 1 j=j0,j1-1
  ax(i) = ax(i) + a(j)*x(ia(j))
return
end
subroutine msluav(n, l,il,nl,kldim, u,iu,nu,du,kudim, dx,r)
*-----*
*-----*
*-----* m s l u s v *-----*
*-----* matrix solver - solve a system L.U dx = r *-----*
*-----*
*-----* forward-backward substitution to solve *-----*
*-----* L U dx = r *-----*
*-----* = ** *-----*
*-----* L, U : lower and upper triangular matrices from (incomplete) *-----*
*-----* = = *-----*
*-----* gaussian elimination *-----*
*-----* r: residuum. right hand side of equations. *-----*
*-----* *-----*
*-----* dx: solution vector delta x *-----*
*-----* = ** *-----*
*-----* implicit double precision (a-h, o-z)
double precision l
dimension l(kldim), il(kldim), nl(n+1)
1  , u(kudim), iu(kudim), nu(n), du(n)
2  , ax(n), r(n)
*-----*
real l
*-----* *** forward sweep
*-----*
j0 = 1
do 100 i=1,n
j1 = nl(i+1)
dx(i) = r(i)
do 110 j=j0,j1-1
jj = il(j)
dx(i) = dx(i) + l(j)*dx(jj)
110 continue
j0 = j1
100 continue
*-----* *** backward sweep
*-----*
i = n
dx(i) = dx(i)*du(i)
iu1 = nu(n)
do 300 i=n-1,1,-1
iu0 = nu(i)
do 310 j=iu0,iu1-1
dx(i) = dx(i) + u(j)*dx(iu(j))
310 continue
iu1 = iu0
dx(i) = dx(i)*du(i)
300 continue
*-----*
return
end
subroutine esgetdm(neq,na,nu,nl,iad,iud,ild)
dimension na(neq+1), nl(neq+1), nu(neq)
iad = na(neq+1)
iud = nu(neq)
ild = nl(neq+1)
return
end
*-----*

```

```

SUBROUTINE TIM(tdi,td,dtd,tdbld,itime,tdt1,tdflw,
&qd,ndi,ntd)

* TIME STEP CONTROL SUBROUTINE

* lflow=0 =; flow period
* lflow=1 =; last time-step in the flow period
* lflow=2 =; shuttin period

implicit real*8(a-h,o-z)
common /logic/iwhflg,lflow,ktcnt,kflg,lgchs,ihw
common/pcte/qdold,tdb,tdnxt,istar

td1 = td
if (lflow .eq. 1) then
  print*, 'START OF BUILDUP PERIOD'
  td = tdi
  dtd = tdi
  qd = 0.0
  lflow = 2
  ktcnt = 1
  ntd = ndi
  dtd1 = dtd
  go to 1
endif

dtd1 = dtd
if (ktcnt .gt. ntd) then
  ktcnt = 1
  ntd = ntd + 100
endif
ktcnt = ktcnt + 1
td = tdi * 10.0**((1.0/ndi))
dtd = td - tdi

if (lflow .eq. 0 .and. td .gt. tdfw) then
  dtd = tdfw - tdi
  td = tdfw
  write(*,*) 'LAST TIME-STEP IN THE FLOW PERIOD'
  lflow = 1
endif

1  if (lflow .eq. 2. and .td .gt. tdbld) then
    itime = 1
    print *, 'End of program at td + tdfw = ',td-dtd+tdflw
  endif
100 return
end

-----
SUBROUTINE TDCOR (m,n,hwold,ktlst,tdbase)
PARAMETER(mcm=80,ncm=100)

* Subroutine used to reduce the time-step to the test, since the
* pressure drawdown in the free-surface stream-layer is large enough
* compared to the volume of any block.
* kflg is a parameter that controls time step fractioning, and
* equals 1 always when the new td value is generated in the
* subroutine TIME. Otherwise, its value is a whole number
* used to divide the time step previously, and is gradually
* increased, since convergence is not reached because of
* time-step is still large.
* ktlst is the recorded last kflg used previously, and is used to
* avoid waste of computer time. Thus, if, for example, the
* need of auto-reduction of the time-step was 5 times in the
* last period, then, when starting to divide the new time-step
* kflg will assume 5 initially.

IMPLICIT REAL*8 (A-H,O-Z)

common /dimiss/td,dtd,hwd,hs,dzdw,dhw,qd,red
common /logic/iwhflg,lflow,ktcnt,kflg,lgchs,ihw
common /alt/zd(mcm,ncm),zdold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&          hdavg(mcm+1),zdavg(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&          dhavg(mcm+1)

if (kflg .eq.1) then
  tdbase = td
  dtd1 = dtd
  if (ktlst .gt. 3) then
    kflg = ktlst - 2

```

```

    else
        ktflag = ktlet
    endif
endif
ktflag = ktflag + 1
dtd = dtd1 / ktflag
td = tdbase + dtd1 + dtd

do 2 j=1,n
    ww(j) = wold(j)
    do 1 i=1,m
        w(i,j) = wold(i,j)
1    continue
2    continue

    do 3 i=1,m
        hd(i) = hold(i)
c        dh(i) = dhi(i)
        dh(i) = dzd(i)
        dhavg(i) = dzavg(i)
3    continue
    dhavg(m+1) = dzavg(m+1)

hs = hd(1)
hwd = hwdold
hwd1 = hwdold
dhw = dzd(1)
dhavg(1) = dzd(1)
iwhflag = 1
lgchs = 0

return
end
=====
* subroutine tetct1 (hwd, td, tmin, teta, tetai)
* VARIABLE TETA DEFINITION (NORMALLY NOT USED)
* implicit real*8 (a-h,o-z)
* real*4 tt, der, tet
*
td3 = td2
td2 = td1
td1 = td
hwd3 = hwd2
hwd2 = hwd1
hwd1 = hwd
der1a = (hwd1 * hwd2) / (td1 - td2)
der1b = (hwd2 * hwd3) / (td2 - td3)
der2old = der2
der2 = 2.d0 * (der1a - der1b) / (td1 - td3)
prod = der2 * der2old

tt = tmin
der = der2

if (dabs(der2) .le. 0.001 .or. prod .lt. 0.) then
    teta = 1.d0
else
    teta = tetai
endif
tet = teta
print ", 'tmin=' ,tt, ' der2=' ,der, ' teta=' ,tet

return
end
=====

```

```

*
*      START subroutine E'1
*
* This subroutine calculates the exponential integral E1(x), where x
* is a real number larger than zero.
* It was kindly furnished by Jan Aasen
*
* For x less than 1.35, series expansion is used.
*
* For x greater or equal to 1.35, the 15-point Gauss-Laguerre
* integration scheme is used.
*
*
* Input:  x = argument [DBL PREC]
* Output: ei = E1(x) [DBL PREC]
*
*
* Valid range of input parameter: x = [0,  $\infty$ )
*
*
* Supporting subroutines: fac(n,n'fac)
*                         lag'e'1(x,ei)
*
*
double precision function ei(x)
double precision x
double precision n'fac, tol, euler, total, sigma
integer n
data tol, euler/1.e-8, .5772156649/
total = 0
if (x .lt. 1.35) then
  do 10 n = 1,15
    call fac(n,n'fac)
    sigma = (-x)**n/(n*n'fac)
    total = total + sigma
    if (abs(sigma/total) .lt. tol) then
      ei = -euler - log(x) - total
      return
    endif
  10  continue
else
  call lag'e'1(x,ei)
  return
endif
end
*
*      END subroutine E'1
*
*
*      START subroutine fac
*
* Input : n = zero or positive [INTEGER]
* Output: n'fac = n! = faculty of n [DBL PREC]
*
*
* Valid range of input parameter: n = 0,1,...,34
*
*
subroutine fac(n,n'fac)
integer n
double precision n'fac
integer i
n'fac = 1.
do 10 i = 1, n - 1
  n'fac = (i + 1.)*n'fac
10  continue
return
end
*
*      END subroutine fac
*
*
*      START subroutine lag'e'1
*

```

```

* This subroutine evaluates the exponential integral E1(x) using the
* 15-point Gauss-Laguerre quadrature. The roots of the Laguerre
* polynomial and the weight factors are from Table 2.4, page 113,
* Carnahan et al: 'Applied Numerical Methods'. The integral is
* approximated using Eq.2.109 on page 115.
*
*
*
* Input : x = lower integration limit [DBL PREC]
* Output : ei = the value of the integral [DBL PREC]
*
*
* subroutine lag'e1(x,ei)
double precision x, ei
double precision z(15), w(15), sigma
integer i
data z/.093307812017, .492691740302, 1.215595412071,
+2.268949528204, 3.667622721751, 5.425336627414,
+7.565916226613, 10.12028568019, 13.130282482176,
+16.654407708330, 20.776478899449, 25.623894226729,
+31.407519169754, 38.530683306486, 48.026085572686/
data w/.218234885940, .342210177923, .263027577942,
+.126425818106, .0402068649210, .00856387780361,
+.00121243614721, .000111674392344, .00000645992676202,
+.000000222631690710, .00000000422743038498,
+.0000000000392189726704, .000000000000145651526407,
+.000000000000000148302705111, .0000000000000000160059490621/
ei = .0
do 10 i = 1,15
  sigma = w(i)/(z(i) + x)
  ei = ei + sigma
10  continue
ei = ei*exp(-x)
return
end
*
*          END subroutine lag'e1
*
*-----*
* subroutine spline(n,x,y,b,c,d)
* Original subroutine in Fligelman (1981)
integer n
double precision x(n),y(n),b(n),c(n),d(n)

INTEGER NM1, IB, I
DOUBLE PRECISION T

NM1=N-1
IF(N LT 2) THEN
RETURN
ENDIF
IF(N LT 3) THEN
GO TO 50
ENDIF

D(1)=X(2)-X(1)
C(2)=(Y(2)-Y(1))/D(1)
DO 10 I=2,NM1
D(I)=X(I+1)-X(I)
B(I)=2.*"(D(I-1)+D(I))
C(I+1)=(Y(I+1)-Y(I))/D(I)
C(I)=C(I+1)-C(I)
10  CONTINUE

B(1)=-D(1)
B(N)=-D(N-1)
C(1)=0.
C(N)=0.
IF(N EQ 3) THEN
GO TO 15
ENDIF
C(1)=C(3)/(X(4)/X(2))-C(2)/(X(3)-X(1))
C(N)=C(N-1)/(X(N)-X(N-2))-C(N-2)/(X(N-1)-X(N-3))
C(1)=C(1)*D(1)**2/(X(4)-X(1))
C(N)=C(N)*D(N-1)**2/(X(N)-X(N-3))

* FORWARD ELIMINATION

15 DO 20 I=2,N
T=D(I-1)/B(I-1)
B(I)=B(I)-T*D(I-1)

```

```

C(I)=C(I)-T*C(I-1)
20 CONTINUE

* BACK SUBSTITUTION
C(N)=C(N)/B(N)
DO 30 IB=1,NM1
I=N-IB
C(I)=(C(I)-D(I)*C(I+1))/B(I)
30 CONTINUE

* COMPUTE POLYNOMIAL COEFFICIENTS
B(N)=(Y(N)-Y(NM1))/D(NM1)+D(NM1)*(C(NM1)+2.*C(N))
DO 40 I=1,NM1
B(I)=(Y(I+1)-Y(I))/D(I)-D(I)*(C(I+1)+2.*C(I))
D(I)=(C(I+1)-C(I))/D(I)
C(I)=3.*C(I)
40 CONTINUE
C(N)=3.*C(N)
D(N)=D(N-1)
RETURN

50 B(1)=(Y(2)-Y(1))/(X(2)-X(1))
C(1)=0.
D(1)=0.
RETURN
END

-----
DOUBLE PRECISION FUNCTION SEVAL(N,U,X,Y,B,C,D)
INTEGER N
DOUBLE PRECISION U,X(N),Y(N),B(N),C(N),D(N)

* ---- THIS SUBROUTINE EVALUATES THE CUBIC SPLINE FUNCTION
* Original subroutine in Fligelman (1981)

INTEGER I,J,K
DOUBLE PRECISION DX
DATA I/1/

IF(I.GE.N) THEN
I=1
ENDIF
IF(U.LT.X(I)) THEN
GO TO 10
ENDIF
IF(U.LE.X(I+1)) THEN
GO TO 30
ENDIF
10 I=1
J=N+1
20 K=(I+J)/2
IF(U.LT.X(K)) THEN
J=K
ENDIF
IF(U.GE.X(K)) THEN
I=K
ENDIF
IF(J.GT.I+1) THEN
GO TO 20
ENDIF
30 DX=U-X(I)
IF (DABS(DX) .LE. 1.D-18) DX = 0.D0
IF (DABS(B(I)) .LE. 1.D-18) B(I) = 0.D0
IF (DABS(C(I)) .LE. 1.D-18) C(I) = 0.D0
IF (DABS(D(I)) .LE. 1.D-18) D(I) = 0.D0
SEVAL=Y(I)+DX*(B(I)+DX*(C(I)+DX*D(I)))
RETURN
END

-----
double precision function aval(n,u,x,y,isort)
PARAMETER(mcm=80,ncm=100)

* n - number of vectors dimension
* u - abscissa value to be interpolated in x(i)
* x - abscissa vector
* y - ordinate vector
* isort - if aval is called for the first time for a given pair of
vector x & y, isort=0. Otherwise, isort .ne. 0

```

```

implicit real*8 (a-h,o-z)
double precision x(n),y(n),x1(ncm),y1(ncm)

* *** Sorting the input vector
if (isort .eq. 0) call sort(n,x,y,x1,y1)

* *** Interpolation
do 1 k = 1,n-1
  if (u .ge. x1(K) .and. u .lt. x1(K+1)) then
    aval = y1(k) + (y1(k+1) - y1(k))/(x1(k+1)-x1(k))*(u-x1(k))
    return
  endif
1 continue
if (u .lt. x1(1)) then
  aval = y1(1) + (y1(2) - y1(1))/(x1(2)-x1(1))*(u-x1(1))
  return
endif
if (u .gt. x1(n))
&  aval = y1(n) + (y1(n) - y1(n-1))/(x1(n)-x1(n-1))*(u-x1(n))
return
end

SUBROUTINE SORT(N,x,y,rx,ry)
implicit real*8 (a-h,o-z)
real*8 x(n),y(n),rx(N),ry(n)
do 1 k=1,n
  rx(k) = x(k)
  ry(k) = y(k)
1 continue
L=N/2+1
IR=N

10  IF(L.GT.1)THEN
    L=L-1
    Rrx=rx(L)
    Rry=ry(L)
  ELSE
    Rrx=rx(IR)
    Rry=ry(IR)
    rx(IR)=rx(1)
    ry(IR)=ry(1)
    IR=IR-1
  ENDIF
  IF(IR.EQ.1)THEN
    rx(1)=Rrx
    ry(1)=Rry
    RETURN
  ENDIF
  ENDIF
  I=L
  J=L+L
20  IF(J.LE.IR)THEN
    IF(J.LT.IR)THEN
      IF(rx(J).LT.rx(J+1))J=J+1
    ENDIF
    IF(Rrx.LT.rx(J))THEN
      rx(I)=rx(J)
      ry(I)=ry(J)
      I=J
      J=J+J
    ELSE
      J=IR+1
    ENDIF
    GO TO 20
  ENDIF
  rx(I)=Rrx
  ry(I)=Rry
  GO TO 10
END
-----
Double Precision Function WAVG(i,n)
PARAMETER(mcm=80,ncm=100)
implicit real*8 (a-h,o-z)
common /alt/zd(mcm,ncm),zold(mcm,ncm),hd(mcm),hold(mcm),zdw(ncm),
&      hdavg(mcm+1),zdavg(mcm+1)
common /pres/w(mcm,ncm),wold(mcm,ncm),ww(ncm),wwold(ncm),wo(ncm)
common /thick/dh(mcm),dh1(mcm),dhold(mcm),dzd(mcm),dzavg(mcm+1),
&      dhavg(mcm+1)
sum = 0.
do j = 2,n

```

```

    sum = sum + w(i,j)
enddo

wavg = (sum*dzd(i) + dh(i)*w(i,1)) / hd(i)
if (wavg .gt. 1.) then
    wavg = 1.d0
endif
return
end

=====
Double Precision Function WWAvg(n,ww,dzdw,dhw,hs)
implicit real*8 (a-h,o-z)
dimension ww(1)
sum = 0.
do j =2,n
    sum = sum + ww(j)
enddo
wwavg = (sum*dzdw + dhw*ww(1)) / hs
return
end

=====
SUBROUTINE CPUTIME(CPUSEC,k)

* OUTPUT:
*   CPUSEC   CPU TIME USED SO FAR.
* THIS SUBROUTINE WORKS ONLY ON ABQAIQ AND PANGEA.

REAL*4 TARRAY(2),ETIME
REAL*8 CPUSEC

CPUSEC = ETIME(TARRAY) = 1000000.

RETURN
END
=====
```

**DATE
FILMED**

12/17/93

END

