THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

HORIZONTAL WELL PRODUCTIVITY AND RISK ASSESSMENT

by

Kristian Brekke

A dissertation submitted in partial fulfillment of

the requirements for the degree of Doctor of Philosophy

in the Discipline of Petroleum Engineering

The Graduate School

The University of Tulsa

1996

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

HORIZONTAL WELL PRODUCTIVITY AND RISK ASSESSMENT

by

Kristian Brekke

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

_____, Co-Chairperson

_____, Co-Chairperson

_____

_____

ABSTRACT

Brekke, Kristian (Doctor of Philosophy in Petroleum Engineering)

Horizontal Well Productivity and Risk Assessment

(144 pp. - Chapter VII)

Co-Directed by Dr. Leslie G. Thompson and Dr. Ovadia Shoham

(184 words)

A simplified, fast and accurate method was developed for predicting long term horizontal well performance. Heterogeneous, anisotropic geology close to the wellbore was considered in addition to pressure loss through the completion. Speed and accuracy were achieved by replacing the well and reservoir simulation with a semi-analytical network approach, and by upscaling reservoir properties for radial flow. Comparison to fine grid reservoir simulations verifies that both total well productivity and flux profile along the well are maintained for the simplified approach. Computational efficiency and comprehensive treatment of the horizontal well problem make the method suitable for complete incorporation of uncertainties connected to the completion, the near wellbore geology and formation damage. The procedure was applied to illustrate how uncertainties in geology and completion efficiency affect the distribution of the total well productivity for finite and infinite conductivity horizontal wells of different lengths. The method proved to be very efficient for this type of study, and indicated positively skewed (log-normal like) productivity distributions for short wells, normal distributions for long wells and a tendency for negative skewness of the productivity distribution from pressure loss in the wellbore.

ACKNOWLEDGEMENTS

I would like to thank the member companies of University of Tulsa Petroleum Reservoir Exploitation Projects (TUPREP) for their support. I also extend my appreciation to the Norwegian Research Council for their financial support throughout this project. I would also like to thank my co-advisors for useful discussions, comments and ideas. Finally, I would like to thank the University of Tulsa for providing the resources and a very pleasent environment.

This work is dedicated to my wife Kimberly, and to my children Andreas, Nicholas and Sarah for their unconditional support and enthusiasm throughout my studies.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Developments in drilling and completion technology have resulted in horizontal wells with longer wellbores, more complex geometry well paths and the possibility of more advanced completion designs than the traditional open hole completion. Prepacked liner completion designs[1] with built in flexibility for last minute adjustments, cemented and perforated liner completions equipped with internal tubulars, inflatable casing packers and sliding sleeves are now considered feasible options. The technology for drilling multiple laterals from the same vertical section and horizontal wells with "branches" has also emerged during the last decade. Compared to vertical wells, these wells present the engineer with a more complicated assignment of predicting well performance.

This new technology provides new tools for the engineer and offers a greater impact on the economic outcome of a project. By using completion technology that facilitates control of inflow along the well, an optimization of the inflow profile according to case specific goals may be attempted. These include suppression of water and gas coning, eliminating negative effects from friction in the wellbore, or balancing the production from different isolated zones. Thus, it is no longer only the correct definition of target location, hole size and well length that may influence the outcome of a planned well.

Information regarding the geometry of the well path or the geology close to the wellbore is a critical input for any attempt to control the inflow profile along a horizontal well. These data are unfortunately not attainable before the well is drilled. Thus, the time

available for the engineer to make last minute adjustments based on simulations is limited in some cases to a window of a few hours or days from the time a well is drilled to the final well completion. For offshore wells, where a later adjustment of the completion is not likely to be cost effective, it is critical that all available information is used to find the best possible completion design. Simple, efficient and accurate prediction methods are required to make use of all available information to maximize production.

The decision to invest capital in the drilling and completion of a well has to be made based on substantially less information than the information available after the well is finally drilled. Thus, to make this decision, it is necessary to estimate the possible outcome of the investment as quickly and accurately as possible, based on the statistical information available from all other sources.

The parameters affecting horizontal well performance involve a higher level of uncertainty as compared to vertical wells.[2] Horizontal wells are affected by geological variations in the horizontal direction besides involving a larger variation in the outcomes of the more complex drilling and completion operations. Thus, the application of long horizontal wells increases the potential both for success and failure. The potential for success can be enhanced by a better understanding of the total reservoir and wellbore interaction and flow behavior before the well is completed. Unexpected failures can be avoided by efficiently including all uncertainties in the predictions before the well is drilled.

This study focuses on the development and application of methodology for comprehensive prediction of production performance for horizontal wells. Besides providing accurate and CPU-time-efficient calculation of the entire horizontal well flow problem, the methodology is developed for the purpose of incorporating the uncertainties

connected to the near wellbore geology, formation damage and completion efficiency. The theory used to describe the horizontal well flow problem in this study is applied to three different regions as follows:

*1) **Flow through the near wellbore reservoir zone.*** Upscaling methods have been developed for radial flow in the formation close to the wellbore. The methods are based on a single phase, steady-state flow assumption. Fine grid simulations confirmed the development of a steady-state flow zone around the wellbore after a very short time. The upscaling methods incorporate the effects of convergent flow around the wellbore through heterogeneous and anisotropic formation. The theory and a computer program are developed for converting the description of a heterogeneous, anisotropic reservoir geology in Cartesian coordinates to an equivalent system in cylindrical coordinates, for upscaling.

*2) **Flow in the Outer Reservoir.*** Two methods were considered for coupling the well and near wellbore simulations to the response from the outer reservoir. The network model for the well and near wellbore reservoir can be coupled to a numerical reservoir simulator by an iterative coupling scheme. It is also shown how a semi-analytical pseudo steady-state reservoir response for non-uniform inflow and pressure profile along the well may be integrated with the well and near wellbore flow calculations through the network model. Productivities are derived by applying superposition in three dimensional space for the connection points between the near wellbore reservoir zone and the outer reservoir.

*3) **Flow through the completion.*** For the wellbore, the applied pressure loss model accounts for the friction pressure loss and the additional pressure loss due to acceleration of the radial inflow into the wellbore.

The mass and momentum balance for the entire flow system are then solved by combining the flow regions in a nonlinear network solver developed in this study.

Results from applying the developed methodology are verified by comparing them to fine grid simulations using the Eclipse[3] numerical reservoir simulator. A step by step verification of the different parts of the semi analytical network simulation approach is performed. The verification is performed for a horizontal well producing single-phase oil through a highly heterogeneous and anisotropic near wellbore formation. Both the total well productivity and the inflow profile along the horizontal well are found to correspond with less than 1 % error between the network model and fine grid numerical reservoir simulations. The effects of the total wellbore pressure loss on the well productivity are illustrated together with the magnitude of the accelerational pressure loss in the wellbore.

A large number of heterogeneous realizations for the near wellbore reservoir zone were generated from variograms and average value of permeability by applying unconditional Sequential Gaussian Simulation[4]. These realizations formed the basis for the generation of probability distributions for total well productivity. Statistical results for the total well productivity are illustrated and discussed for cases of different well lengths and different levels of pressure loss in the wellbore.

CHAPTER II

UNCERTAINTIES AFFECTING PRODUCTIVITY

The comprehensive incorporation of uncertainties in different parts of a horizontal well flow system can theoretically be performed by applying traditional reservoir simulation techniques. However, reservoir simulators have not been sufficiently flexible for incorporation of horizontal wellbore pressure loss prediction beyond the open hole or perforated liner completions. In addition, creation of a reliable probability distribution of well productivities may require several hundred simulation runs. The computer time and engineering labor combined with a large number of simulations are substantial. Probability distributions for productivity or cumulative production may in addition be required for a variety of different completion and reservoir parameters such as correlation length (variogram range), degree of anisotropy, well length and well path, skin and completion method, diameter and roughness. In this case, the problem quickly becomes unmanageable with currently available reservoir simulation programs.

Thus, a flexible, sufficiently accurate and fast method for predicting long term well performance is a critical element in a successful and practical statistical approach. Figure 2.1 illustrates the layout of the risk assessment procedure with a nonlinear network solver, indicated in the middle, as being this critical element.

Average values and information for the spatial correlation of the permeability field in the area of the planned well may be present from information gained from wells in the

same or similar reservoirs, seismic data or outcrops.

A statistical method, in this case unconditional Sequential Gaussian Simulation, was used for generation of multiple permeability fields for the near wellbore zone. If data from a pilot hole are available, conditional simulation of the permeability field may be applied to reduce the uncertainty in near wellbore geology.

For calculation efficiency, the network solver relies heavily on a reduction of the problem from the original grid resolution by the upscaling procedure indicated in Fig. 2.1. The permeability fields can be upscaled for a certain well trajectory and effective permeabilities are generated as input for the network solver.

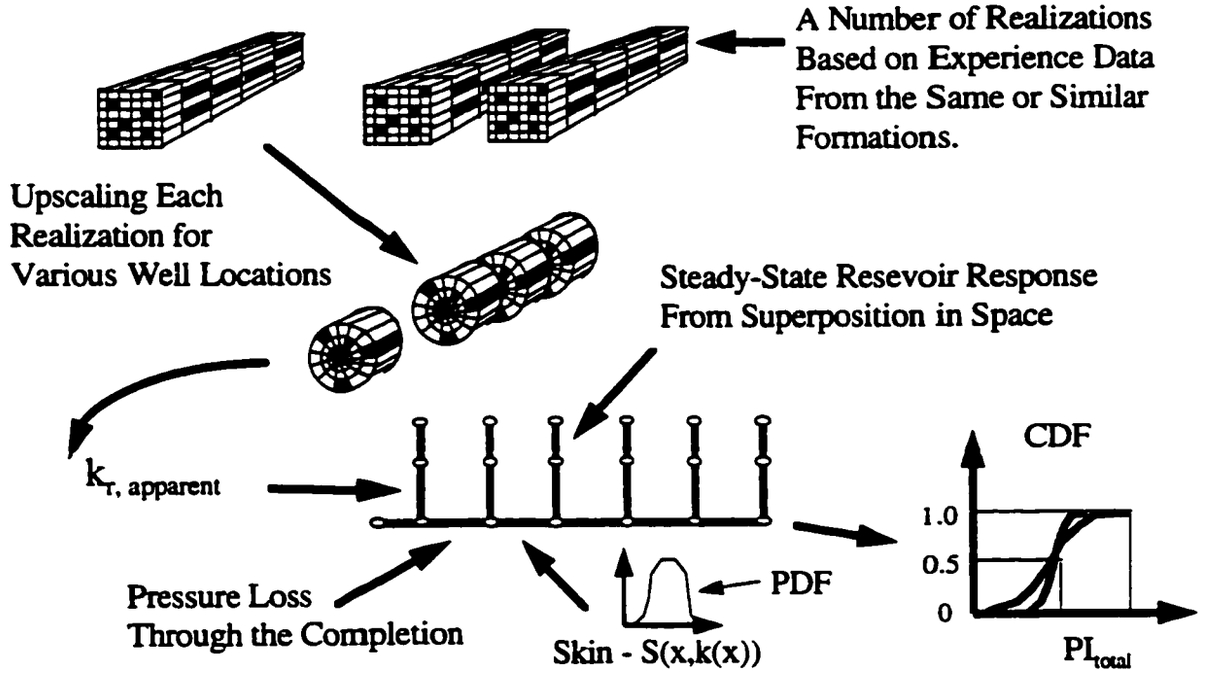

Fig. 2.1: Uncertainty Assessment Procedure - Overview

Distributions for skin and completion roughness can be drawn from previously defined distributions. These distributions may be obtained empirically and are likely to be field specific. However, the skin distribution may be linked to the formation's exposure time to mud during drilling operation[5] and to local permeability at a given location along the wellbore. For this study, the distribution for roughness was not made dependent upon other parameters, but rather a uniform distribution was used for illustration of the procedure.

The flow problem may now be simulated repeatedly with the network model for variations in the input variables, and the result is probability distributions for total well productivity. Thus, the effect on the shape of the probability distribution from changes in well configuration (length, trajectory, completion type) can be investigated. This may be of importance since, for an operation with very few wells, we may be more interested in knowing the most expected outcome (mode) of well productivity rather than the average of all possible outcomes (mean). In addition, we would like to predict the level of confidence for obtaining a certain minimum productivity.

CHAPTER III

SIMPLIFIED, SEMI-ANALYTICAL SIMULATION APPROACH

Compared to the evaluation of productivity for a vertical well, a proper evaluation of the productivity for a horizontal well often involves prediction of more complex and interdependent flow phenomena for the well, the near wellbore reservoir zone and the outer reservoir. For this study, a non-linear network solver was developed to perform efficient, steady state, comprehensive calculation of the flow through the completion and the near wellbore reservoir zone.

Two methods are proposed for implementation of the outer reservoir response. The first method is to couple the network model iteratively to a transient numerical reservoir simulator. This option has the benefit of including more general and transient behavior for the outer reservoir prediction. The other option is to include steady state relationships based on superposition in space as an integrated part of the network system. These relationships provide steady state outer reservoir response for inflow to discrete elements along the well.

The network solver is non directional, which means that flow in any geometry or any direction can be represented by a flow connection. Unlike nodal analysis programs, commonly used for pressure loss calculations in the wellbore, the main direction of flow is not required as an input, but is rather calculated as part of the procedure. Any pair of nodes in the system can be connected by arcs (flow connections) representing any flow

geometry for which a flowrate vs. pressure loss relationship is known. The two main differences between a network solver and any other discrete flow problem formulation methodology are the flexibility of defining flow connections and the automatic generation of mass and momentum equations resulting from the flow problem topology. Thus, no dimensional or geometrical restrictions apply, which is usually the case for finite difference simulators commonly applied for the reservoir and the wellbore.

3.1 Steady State Network Solver

The basic theory for network solvers has been well documented.[6] Linear network solvers are most thoroughly documented, but can be applied only for linear problems such as steady-state, single-phase flow of Newtonian fluids through porous media. For single-phase turbulent flow and multiphase flow along the well, and for multiphase flow through porous media, a nonlinear relationship exists between rate and pressure loss. One or all of these flow conditions are normally present in a horizontal well, and a nonlinear network solver[7,8] was, therefore, required. The solution to the problem has to be approximated by applying an iterative approach.

The network consists of nodes (connection points) and arcs (flow connections) as illustrated in Figure 3.1. The nodes may be configured as one of the following three types:

a) Unknown flowrate and pressure nodes - (typically used for internal nodes)

b) Specified pressure, but unknown mass flowrate - (typically used for boundary nodes)

c) Specified mass flowrate, but unknown pressure - (typically used for boundary nodes)

For the resulting problem to be well posed, at least one terminal node should be a specified pressure node.

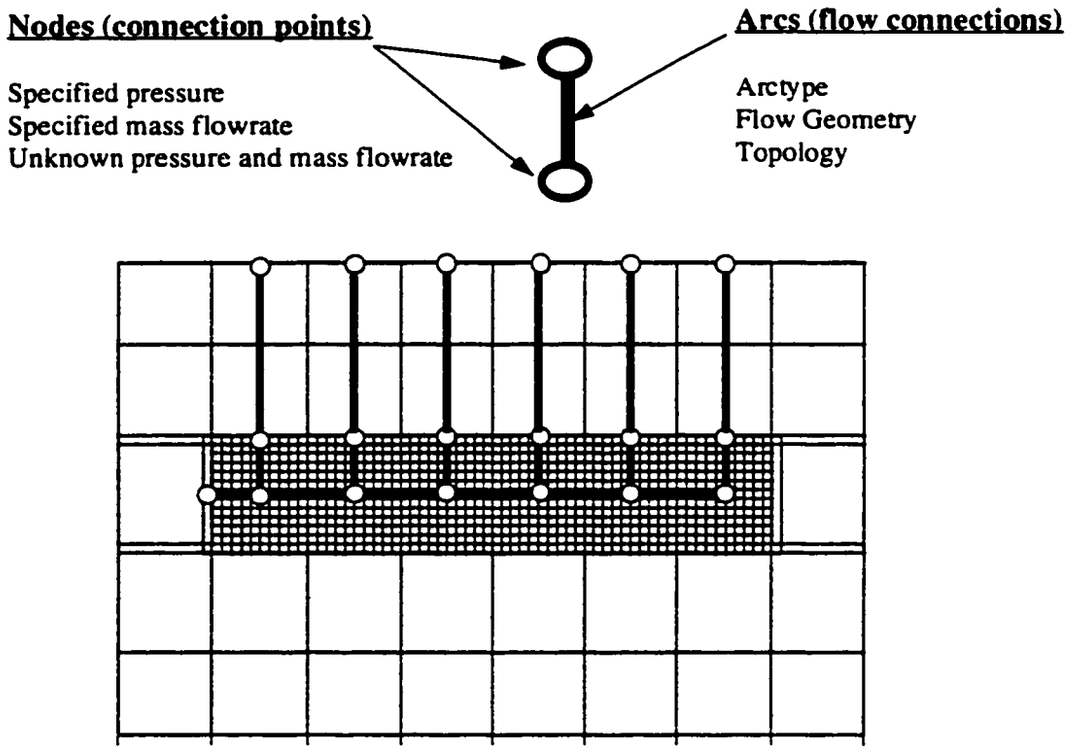

Figure 3.1: Steady State Network Solver Schematics

The network flow problem is constructed from mass and momentum balances. As no accumulation of mass is allowed anywhere in the system, a mass balance, as given by equation 3.1, can be formulated for each node accounting for all mass entering or leaving the node through the connected arcs:

$$\sum_{i=1}^{n} \dot{m}_i = 0, \tag{3.1}$$

where n is the number of arcs connected to the specific node. The mass flow into the node has a positive value while the flow out from the node is negative.

Thus, internal nodes have a net mass flow of zero. Boundary, or terminal nodes may be assigned a net incoming or outgoing mass flowrate, as shown in Equation 3.2. Production from the node is assigned a negative mass flowrate and a positive mass flowrate denotes injection. In this case the pressure of the node is to be calculated.

$$\sum_{i=1}^{n} \dot{m}_i = \dot{m}_{in/out} \tag{3.2}$$

If the pressure rather than the mass flowrate is being specified, the mass balance is omitted for this node, and an equation specifying the pressure is generated, as given in Equation 3.3.

$$p_j = p_{specified} \tag{3.3}$$

For each arc in the network a momentum balance can be defined which simply describes the pressure differential between the arc's upstream and its downstream connected node as a function of the mass flowrate. If this relationship is linear, the momentum balances are given by equation 3.4. The constant c is determined by the geometry and fluids' characteristics for the flow connection in question .

$$(p_u - p_d)_{k+1} = \Delta p_{k+1} = c \cdot \dot{m}_{k+1} \tag{3.4}$$

For flow connections with nonlinear relationships between pressure loss and mass flowrate, no explicit formulation of the pressure loss can be given. For this case, c is not a constant, but rather a function of the mass flowrate. However, an iterative solution can be applied. The pressure loss through the arc for the new iteration level (k+1) can be expressed by applying Newton's method as given in Equation 3.5 as the old iteration level

pressure loss plus the difference in mass flowrate between the two iteration levels times the derivative in pressure loss with respect to a change in mass flowrate at the old iteration level.

$$\Delta p_{k+1} = \Delta p_k + (\dot{m}_{k+1} - \dot{m}_k) \cdot \frac{\delta \Delta p_k}{\delta \dot{m}_k} \tag{3.5}$$

We may now rearrange this equation with the variables for the new iteration level on the left side and the preceding iteration level variables on the right side.

$$\Delta p_{k+1} - \frac{\delta \Delta p_k}{\delta \dot{m}_k} \cdot \dot{m}_{k+1} = \Delta p_k - \dot{m}_k \cdot \frac{\delta \Delta p_k}{\delta \dot{m}_k} \tag{3.6}$$

For each arc, the derivatives of pressure loss with respect to mass flowrate can be found numerically by perturbing the mass flowrate and performing two pressure loss calculations: For single phase flow inside a horizontal completion, this derivative would be calculated by perturbation of flowrate in Equations 3.17 through 3.19. Thus, as the derivative of the arc's pressure loss is approximated numerically, the program only requires routines to be derived for the arc's pressure loss and not the derivative.

A simple horizontal well flow problem is given in Figure 3.2 to illustrate the procedure. The upstream nodes (N1 and N2) have pressure specified, and the downstream node (N5) has a specified mass flowrate. Thus, the pressure for nodes N3 and N4, and the mass flowrates for arcs A1, A2, and A3 are to be calculated. The following three mass balances can be prepared for the nodes without specified pressure (N3, N4 and N5):

$$\dot{m}_1 + \dot{m}_3 = 0 \tag{3.7}$$

$$\dot{m}_2 + \dot{m}_3 + \dot{m}_4 = 0 \ . \tag{3.8}$$

$$\dot{m}_4 = \dot{m}_{5,specified} \tag{3.9}$$

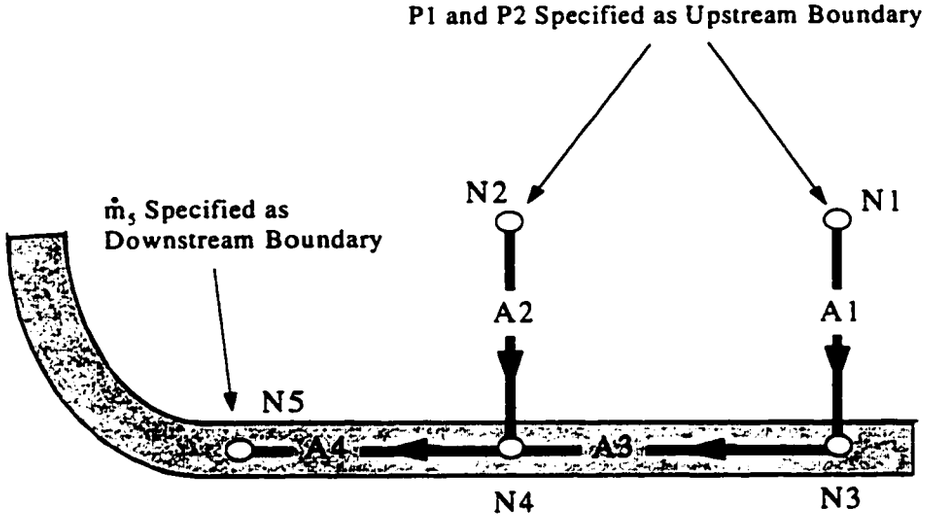

Figure 3.2: Example Network Problem - Simple Horizontal Well Problem

For the two upstream boundary nodes, N1 and N2, the two boundary conditions are:

$$p_1 = p_{1,specified} \tag{3.10}$$

$$p_2 = p_{2,specified} \cdot \tag{3.11}$$

Finally, the momentum balance for arcs A1 through A4 can be formulated, respectively, as:

$$(p_3 - p_1)_{k+1} - (\frac{\delta(p_3 - p_1)}{\delta \dot{m}_1})_k \cdot \dot{m}_{1,k+1} = (p_3 - p_1)_k - \dot{m}_{1,k} \cdot \frac{\delta(p_3 - p_1)_k}{\delta \dot{m}_{1,k}} \tag{3.12}$$

, where $-(\frac{\delta(p_3 - p_1)}{\delta \dot{m}_1})_k$ is defined as c1 ,

and $(p_3 - p_1)_k - \dot{m}_{1,k} \cdot \frac{\delta(p_3 - p_1)_k}{\delta \dot{m}_{1,k}}$ is defined as b1 for simplicity in further derivations.

$$(p_4 - p_2)_{k+1} - (\frac{\delta(p_4 - p_2)}{\delta \dot{m}_2})_k \cdot \dot{m}_{2,k+1} = (p_4 - p_2)_k - \dot{m}_{2,k} \cdot \frac{\delta(p_4 - p_2)_k}{\delta \dot{m}_{2,k}} , \qquad (3.13)$$

where $-(\frac{\delta(p_4 - p_2)}{\delta \dot{m}_2})_k$ is defined as c2 and $(p_4 - p_2)_k - \dot{m}_{2,k} \cdot \frac{\delta(p_4 - p_2)_k}{\delta \dot{m}_{2,k}}$ as b2,

$$(p_4 - p_3)_{k+1} - (\frac{\delta(p_4 - p_3)}{\delta \dot{m}_3})_k \cdot \dot{m}_{3,k+1} = (p_4 - p_3)_k - \dot{m}_{3,k} \cdot \frac{\delta(p_4 - p_3)_k}{\delta \dot{m}_{3,k}} , \qquad (3.14)$$

where $-(\frac{\delta(p_4 - p_3)}{\delta \dot{m}_3})_k$ is defined as c3 and $(p_4 - p_3)_k - \dot{m}_{3,k} \cdot \frac{\delta(p_4 - p_3)_k}{\delta \dot{m}_{3,k}}$ as b3,

$$(p_5 - p_4)_{k+1} - (\frac{\delta(p_5 - p_4)}{\delta \dot{m}_4})_k \cdot \dot{m}_{4,k+1} = (p_5 - p_4)_k - \dot{m}_{4,k} \cdot \frac{\delta(p_5 - p_4)_k}{\delta \dot{m}_{4,k}} , \qquad (3.15)$$

where $-(\frac{\delta(p_5 - p_4)}{\delta \dot{m}_4})_k$ is defined as c4 and $(p_5 - p_4)_k - \dot{m}_{4,k} \cdot \frac{\delta(p_5 - p_4)_k}{\delta \dot{m}_{4,k}}$ as b4.

The resulting set of momentum and mass balances can now be arranged as the matrix problem given by equation 3.16 and solved by a sparse matrix solver.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -c1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -c2 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -c3 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -c4 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \\ \dot{m}_4 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{m}_{5,specified} \\ p_{1,specified} \\ p_{2,specified} \\ b1 \\ b2 \\ b3 \\ b4 \end{bmatrix} \qquad (3.16)$$

The coefficients c1..c4 and the right hand side b1..b4 contain information obtained from the last iteration level. Thus, these value of these coefficients are updated as the system is repeatedly solved until convergence is obtained.

3.2 Network Solver Applied in the Wellbore

The flow inside a horizontal well differs from regular pipe flow because of the non-uniform flowrate along the well due to the influx from the reservoir. The fluids are continuously entering the wellbore from the upstream (toe) part of the well towards the downstream (heel) part of the well. An approximate solution can be obtained by dividing the wellbore into flow connections where the main flow connections represent the wellbore, and the branched flow connections relate to the flow entering the wellbore through segments of the reservoir. The network solver can be used in this way to divide the wellbore into elements with increasing flowrates towards the heel part of the well. For turbulent, single-phase liquid flow through a wellbore arc, the pressure loss was modeled by using the equation for single phase flow frictional pressure gradient given in Equation 3.17 and a Moody type friction factor.

$$\frac{dp}{dl}_{frict.} = f\frac{\rho v^2}{2d_i} \tag{3.17}$$

The rough pipe Moody friction factor was calculated from Hålands formula given by Equation 3.18:

$$f = \frac{1.0}{(1.8 \cdot \log_{10}(6.9 / N_{\mathrm{Re}} + (e_D / 3.7)^{1.11}))^2} . \tag{3.18}$$

In addition to the frictional pressure gradient, an acceleration pressure loss is present along the wellbore. The flow configuration causing this pressure loss is illustrated in Figure 3.3.

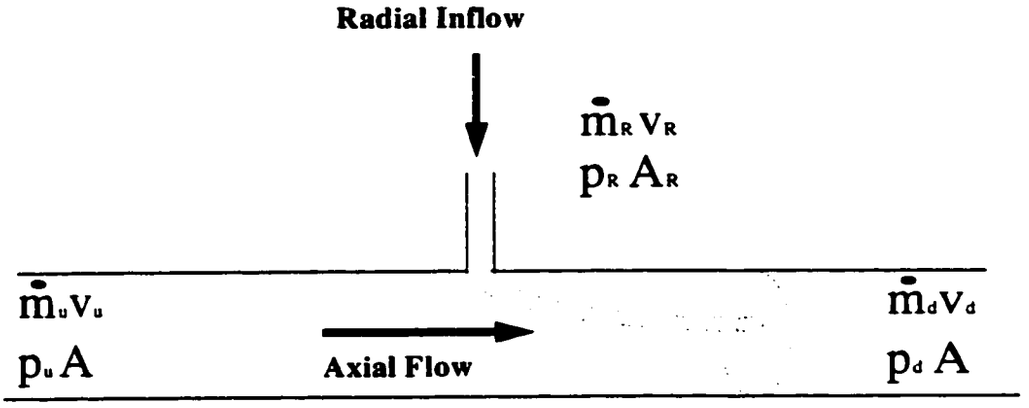

Figure 3.3: Acceleration Pressure Loss Along the Wellbore due to Radial Inflow

Both the fluids entering a wellbore arc from the upstream part of the wellbore (subscript u) and radially (subscript r) from the reservoir along the arc are being accelerated to their combined axial velocity (subscript d) at the downstream end of the arc. Several studies have been performed investigating this additional pressure loss. Aasheim derived a relationship[15] for the acceleration pressure loss due to radial inflow given by:

$$\Delta p_{acc.} = \frac{(\dot{m}_d v_d - \dot{m}_u v_u)}{A} . \qquad (3.19)$$

This equation is based on a simple momentum balance along the pipe past a perforation and assumes that the radial inflow is entering the completion with a zero velocity. Thus, no energy is assumed to be gained from the entrance velocity of the radial inflow. The

model was in good agreement with experimental data[16] obtained from a large diameter test facility for axial flow with radial inflow.

3.3 Network Solver Applied in the Near Wellbore Zone

In the previous section it was shown how the network simulator could be applied to calculate the pressure and flow behavior in the wellbore. As the network model relies on the assumption of steady-state flow, its applicability for reservoir predictions had to be further investigated. If we assume that the first few feet of the reservoir, close to the wellbore, attain steady state flow conditions after a very short production time, the network simulator may be used to model the pressure and flow behavior in the near wellbore zone while ignoring only negligible transient effects. Verification of the growth of a steady flow zone near the well is presented in Section 5.2.

Efficient use of the network solver in the near wellbore zone is, however, dependent on some prior knowledge of the main flow direction in the reservoir with respect to the well. Far away from the well, where the pressure gradient towards the well is likely to be small, the direction of the flow is strongly dependent on the geology and, thus, hard to predict. However, close to the well, where the pressure gradient in the direction of the well is significant, and the pressure gradients in the θ (as defined in Figure 3.4) and well direction are small, we may assume that radial flow perpendicular to the well is the dominating flow geometry.

Thus, the radial flow assumption together with the steady-state flow assumption facilitate upscaling of the permeabilities from a large number of blocks around the well

into one apparent permeability for a much larger well block. This operation should be performed in a way that preserves the physics of the flow problem of the "real" fine grid, heterogeneous case.

If a fine resolution heterogeneous description of reservoir properties is present from applying geostatistical methods, this description is likely to be presented in a Cartesian grid. Figure 3.4 illustrates how the near wellbore zone can be implemented in the network solver by the introduction of flow connections used to model the flow behavior of the near wellbore reservoir zone.

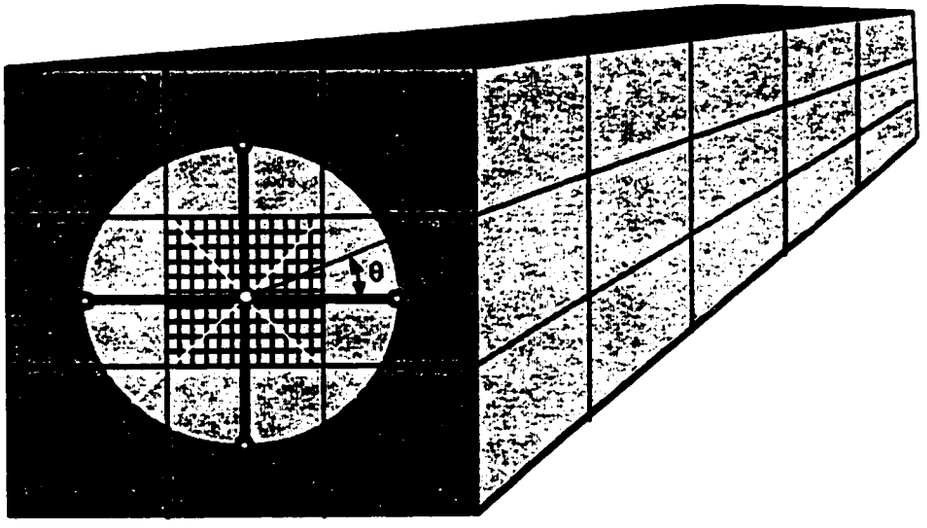

Figure 3.4: Schematics - Network Solver Applied in the Near Wellbore Reservoir Zone

If we overlay the Cartesian grid with a cylindrical grid symmetrically arranged around the well, each cylindrical grid block will have an apparent directional permeability towards the well. The apparent directional permeability towards the well can be calculated by applying especially derived upscaling methods for radial flow, presented in Chapter 4. One apparent permeability may be obtained for a full cylindrical block, or if a definition of

inflow in the θ direction is required, the cylinder can be divided into "slices" with each having an upscaled permeability.

In cases with anisotropic formation where the pressure loss through the near wellbore zone is a strong function of θ, at least four slices are required to capture the variability in influx in the theta direction. For an extension of this method to multiphase coning cases where a segregation of fluid occurrs around the wellbore, an even finer division in the θ direction may be required.

Block permeabilities may also be upscaled in the direction of the well without directly affecting the total well productivity significantly. However, a lumping of blocks in the well direction will reduce the resolution of the inflow profile along the well, and may indirectly have an impact on the total well productivity through the calculated wellbore pressure loss.

3.4 Network Solver Applied to the Reservoir Outside the Near Wellbore Zone

The objective for applying the network simulator to the near wellbore zone is to reduce the computational problem by significantly reducing the number of grid blocks involved in a numerical simulation of the reservoir and well. However, the well and the near wellbore zone can not be considered as an isolated flow problem uncoupled from the outer reservoir. The shape of the inflow profile has a profound impact on the productivity of the reservoir towards the boundary of the near wellbore zone. Thus, methodology for including the outer reservoir performance in the network model were investigated.

Since the scope of this work was to investigate long term productivities, we are interested in the pseudo-steady state or steady-state flow performance more than the early transient reservoir response. Several late time performance relationships for horizontal wells are available in the literature. However, all of these methods make the simplifying assumptions of either uniform flux or uniform pressure along the well. Models considering a random inflow profile or pressure profile along the well are not readily available. For the problem of modeling the late time performance response from the reservoir outside a heterogeneous near wellbore zone, a non-uniform inflow and pressure profile model is required. The pressure loss inside the well will cause a non-uniform well pressure, and the pressure loss and rate through the near wellbore zone will vary with location along the well.

Similar to the near wellbore zone, the response from the remainder of the reservoir has to be divided into discrete sections along the well for inclusion in the network simulator. Unlike the near wellbore zone, the cross flow in the reservoir can not be neglected for this case. The outer reservoir productivity for a connection point is highly dependent on the flowrate of the connection point under consideration. The connection productivity is also dependent on the production rates for the neighboring connection because the connection points are competing for drainage volume. The larger the rate for a connection point, the larger its drainage volume becomes and, as a result, the larger is its productivity. Also, small production rates from neighboring connections favor a more efficient shape of the drainage volume for the connection point under consideration. For these cases, the productivity is, therefore, improved.

Thus, the interdependency of the connection points must be taken into account when calculating the productivities. Muskat[14] developed steady-state theory for multiple wells inside a circular drainage area. He assumed a two dimensional, steady-state flow, and his theory was based on superposition in space. The method, originally presented for multiple vertical well systems, was extended to three dimensions in this study. Rather than considering multiple vertical wells, the superposition was performed for all the connection points between the reservoir and the near wellbore reservoir zone along a horizontal well.

We assumed that the near wellbore zone elements could be regarded as small cavities in the shape of spheres and that the reservoir itself has the shape of a sphere. A schematic illustration of the configuration is given in Figure 3.5, where r_{wi} denotes the radius of each producing sphere, $d_{i,j}$ is the distance between two producing spheres and r_e is the radius of the reservoir sphere. The pressure distribution in the reservoir can be found by integration of Darcy's law for spherical flow. For each connection point in a three dimensional space, the pressure distribution can then be represented by the following equation:

$$p = C - \frac{q\mu}{4\pi k}\left(\frac{1}{r}\right). \tag{3.20}$$

The average pressure on the spherical reservoir boundary can be determined by adding the effects from each connection point using the average distance from the connection point to the boundary. The average distance to the boundary from a well at any location in a two dimensional circular reservoir was, according to Muskat[14], set to be equivalent to the radius of the bounded region. For the three dimensional, spherical reservoir case, the average boundary pressure is then given by:

$$p_e = c - \sum_{i=1}^{n} \frac{q_i \mu}{4\pi k}\left(\frac{1}{r_e}\right). \tag{3.21}$$

This is an approximation of the average distance to the spherical boundary which is only true for a drainage point located in the center of the reservoir. An eccentric location of the drainage point may for the worst case (where the drainage point is located on the boundary) have up to a 33% longer average distance to the boundary. However, the error made by approximating the average radius from the drainage points was found to be negligible in this study.

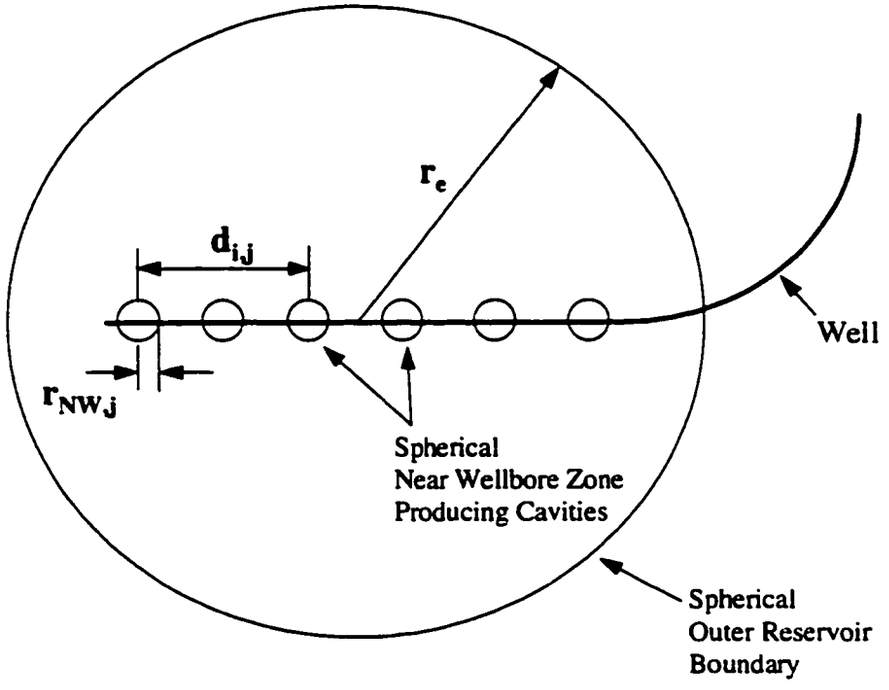

Figure 3.5: Reservoir Productivities From Superposition - Schematics

For true steady-state flow, c in Equation 3.21 is constant. For pseudo-steady state flow, c is a function of time and must be determined to ensure that the material balance is preserved. This given value can be found either from performing an analytical material balance calculation or from a reservoir simulation.

For the more likely case where the shape of the reservoir is not spherical, the average radius from the well to the boundary is found as the radius of an equivalent sphere where pore volume is equal to the reservoir pore volume. The pressure at any of the involved connection points between the reservoir and the near wellbore reservoir zone can now be determined by adding the contributions from all connection points. The connection pressures can be determined by the expression given in Equation 3.22:

$$p_{NW,j} = c - \frac{q_j \cdot \mu_j}{4 \cdot \pi \cdot k}\left(\frac{1}{r_{NW,j}}\right) - \sum_{i=1, i \neq j}^{n} \frac{q_i \cdot \mu_i}{4 \cdot \pi \cdot k}\left(\frac{1}{d_{ij}}\right). \tag{3.22}$$

The value of **c** is in this equation the same as for Equation 3.21. An equation for the outer reservoir productivity can now be obtained by combining Equations 3.21 and 3.22:

$$PI_j = \frac{q_j}{(p_e - p_{NW,j})} = \frac{q_j}{\left(\sum_{i=1}^{n} \frac{q_i \mu}{4\pi k}\left(\frac{1}{r_e}\right) + \frac{q_j \cdot \mu_j}{4 \cdot \pi \cdot k}\left(\frac{1}{r_{NW,j}}\right) + \sum_{i=1, i \neq j}^{n} \frac{q_i \cdot \mu_i}{4 \cdot \pi \cdot k}\left(\frac{1}{d_{ij}}\right)\right)}. \tag{3.23}$$

The only remaining unknown parameter is the apparent wellbore radius for the sections of the near wellbore zone. This value can be found by matching the network model outer reservoir connection productivities to the productivities obtained from a reservoir simulation of the uniform wellbore pressure case. Thus, only one reservoir simulation is necessary to match the reservoir part of the network simulator and the network simulator will now work for any non-uniform inflow profile resulting from pressure loss along the wellbore or heterogeneous near wellbore geology.

If the network model is to be applied without the assistance of a reservoir simulator, the apparent radius of the spherical wellbore used in the superposition can be found by equating the surface areas of the apparent spherical and the actual cylindrical near wellbore reservoir region.

It should be noted that the well is assumed to be located in the center of the reservoir. The inflow profile resulting from applying this method is not influenced by the reservoir boundaries being closer to the ends of the well. Thus, the method can not be used for eccentrically located wells without the implementation of image wells or shape factors for prediction of boundary effects.

The method also assumes an isotropic, homogeneous formation in the outer reservoir. An upscaling procedure based on spherical flow geometry would be required for implementing the effects of anisotropy and heterogeneities in the semi-analytical scheme for prediction of outer reservoir response. The exact reservoir geometry can be rigorously accounted for by the use of image well. However, we have not pursued this aspect further in this work.

3.5 Network Model Coupled to a Transient Reservoir Simulator

For cases with multiphase flow or where transient effects dominates in the outer reservoir during long production times, the simulation of the outer reservoir can not be handled by the proposed method based on superposition in space. The network solver can, in this case, be interfaced with a numerical reservoir simulator when used exclusively for the wellbore calculations. The transient response from the outer parts of the reservoir can then be obtained by an iterative coupling to the reservoir simulator.

The feasibility of this procedure has been previously shown[8]. However, for completeness, the methodology is described here as an alternative option to the superposition approach. A schematic illustration of the concept is given in Figure 3.6.

Connection flowrates are provided to the network simulator indirectly by the use of transfer variables. These variables provide sufficient information to construct inflow performance relationships (IPRs) for the connection points between the well and the reservoir.

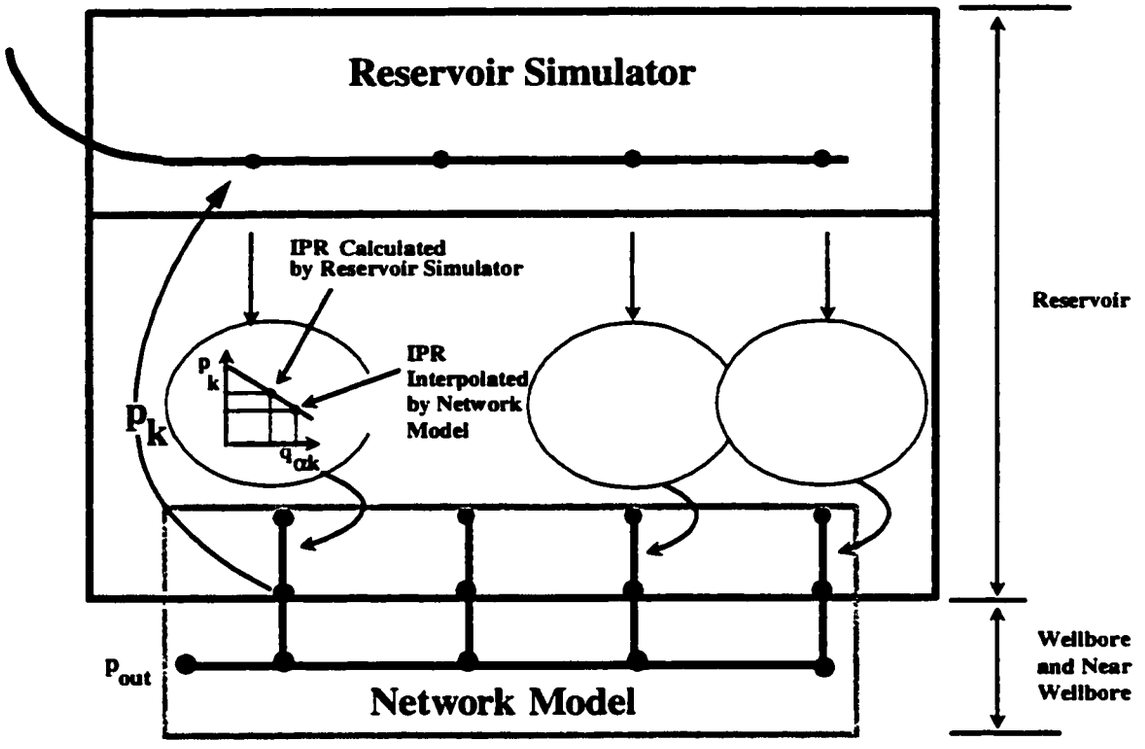

Figure 3.6: Network Model Coupled to a Transient Reservoir Simulator - Schematics

The IPRs provide the network simulator with representative relationships between time, sand face pressure and reservoir deliverability along the well. Thus, an updated simple reservoir model, consisting of straight line inflow performance relationships, is used in the network model to indirectly provide the boundary conditions necessary to calculate the pressure profile along the well.

3.5.1 The Interface Network Model / Reservoir Simulator

The downstream boundary condition (during production) in the network model is either the total flowrate q_t or the pressure p_{out}. The upstream boundary conditions to be used in the network model are reference pressures ($p_{ref,k}$) provided indirectly from the reservoir simulator through transfer variables. The reference pressures are obtained by extrapolating the connection pressures (p_k) linearly to obtain zero flowrates for the connection points, as illustrated in Figure 3.7.

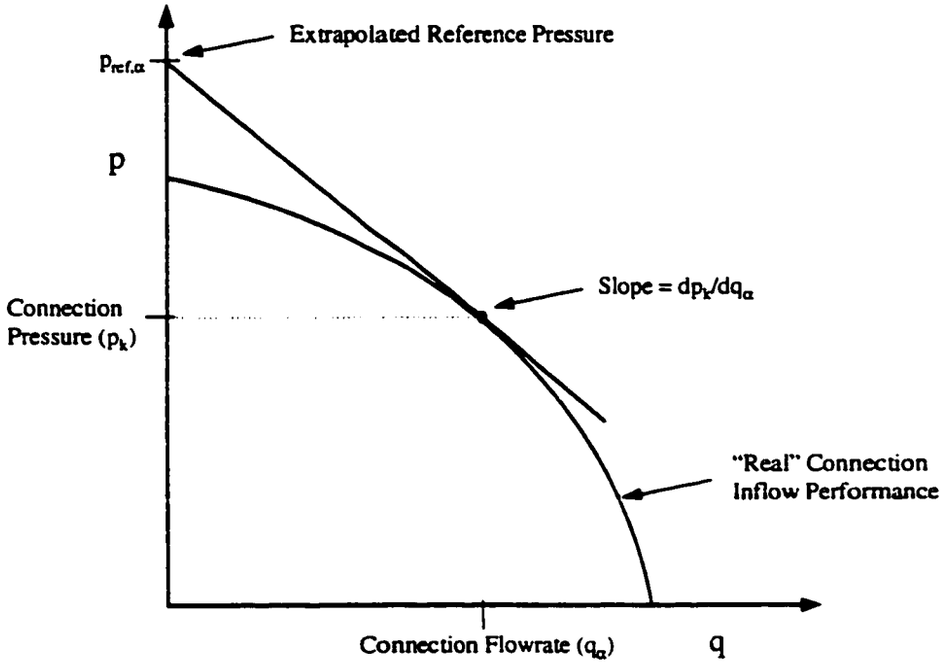

Figure 3.7: Extrapolation of Reference Pressure - Simulator Coupling

To extrapolate these reference pressures in the network model, it is necessary to obtain the flowrates ($q_{\alpha k}$), the connection pressure (p_k) and the derivatives of the flowrates with

respect to pressure ($\delta q_{\alpha k}/\delta p_k$) for each specific connection point. These variables are dependent on time and also on the production for neighboring connection points.

Based on this information and the assumption of steady-state flow in the wellbore, the network model calculates the corrected connection pressures p_k as transfer variables back to Eclipse in addition to the corresponding flowrates ($q_{NW,\alpha k}$). By comparing the flowrates obtained from the network model to the flowrates obtained from the reservoir simulation, the conservation of mass between the two simulators can be verified.

The connection flowrates and pressures are passed between the reservoir simulator and the network model in an iterative procedure. However, the partial derivatives ($\delta q_{\alpha_k}/\delta p_k$) must be updated for each iteration. Below we first discuss a general scheme for coupling the network model to the reservoir simulator, where we assume known values for the partial derivatives ($\delta q_{\alpha_k}/\delta p_k$) when calling the network model. Thereafter, several possible approaches for generation of these partial derivatives are discussed.

3.5.2 General Iterative Coupling Scheme

Let $(p_k, S_{\alpha ij}^{old}, P_{\alpha ij}^{old})$ denote wellbore pressures and phase saturations and pressure at each reservoir simulation grid block for the preceding time level. If we advance the simulation by one time step, let $(p_{\alpha ij}^{new}, S_{\alpha ij}^{new})$ denote the new values of grid block phase pressure and saturations, and let $(q_{\alpha k})$ and $(\delta q_{\alpha k}/\delta p_k)$ denote the resulting phase rate at the wellbore and derivative of phase rate with respect to wellbore pressure, respectively. Similarly, let $(q_{NW,\alpha k})$ denote the phase rate and (p_k) the wellbore pressures computed using the

network simulator. The input for the network simulator are the reservoir simulator's computed values of $(q_{\alpha k})$ and$(\delta q_{\alpha k}/\delta p_k)$ found by applying earlier iteration level values for wellbore pressures (p_k) and a specified outlet boundary condition, which may be either pressure (P^n_{out}) or total flowrate ($q^n_{NW,t}$).

In addition, the network simulator requires upstream boundary pressures for the problem to be well posed. From (p_k), $(q_{\alpha k})$ and$(\delta q_{\alpha k}/\delta p_k)$, and assuming a constant phase flowrate derivative with respect to wellbore pressure, reference pressures at $(q_{\alpha k} = 0)$ can be extrapolated for each connection point to find the boundary pressures. The following relationship applies:

$$p_{\alpha,ref.} = p_k - \frac{q_{\alpha k}}{\delta q_{\alpha k}/\delta p_k} \quad . \tag{3.24}$$

Both connection rate and connection pressure are in this way implicitly implemented as boundary conditions by constructing the inflow performance relationships for each phase and at each connection point as below:

$$q_{\alpha k} = (-\frac{\delta q_{\alpha k}}{\delta p_k})(p_{\alpha,ref.} - p_k) \quad . \tag{3.25}$$

Adjusted wellbore connection pressures (p_k) together with adjusted phase rates ($q_{NW,\alpha k}$) at each connection point are output from the network model.

To advance the coupled simulation from one time level (n) to the next time level (n+1), an iterative procedure is used with (m) as the iteration index. The procedure can be described as follows:

(1) $m := 0; p_k^{n+1,m} := p_k^n; P_{\alpha ij}^{n+1,m} := P_{\alpha ij}^n; S_{\alpha ij}^{n+1,m} := S_{\alpha ij}^n; q_{NW,\alpha k}^{n+1,m} := q_{NW,\alpha k}^n; q_{\alpha k}^{n+1,m} := q_{\alpha k}^n$

(2) Reservoir Simulator $(p_k^{n+1,m}, S_{\alpha ij}^{n+1,m}, P_{\alpha ij}^{n+1,m}) \Rightarrow P_{\alpha ij}^{n+1,m+1}, S_{\alpha ij}^{n+1,m+1}, q_{\alpha k}^{n+1,m+1}, (\frac{\delta q_{\alpha k}}{\delta p_k})^{n+1,m+1}$

(3) Network Model $(p_k^{n+1,m}, q_{\alpha k}^{n+1,m+1}, (\frac{\delta q_{\alpha k}}{\delta p_k})^{n+1,m+1}) \Rightarrow p_k^{n+1,m+1}, q_{NW,\alpha k}^{n+1,m+1}$

(4) $IF: \sum_k \sum_\alpha [q_{NW,\alpha k}^{n+1m+1} - q_{NW\alpha k}^{n+1,m}]^2 \leq \varepsilon; n := n+1; GOTO(1); ELSE: m := m+1: GOTO(2)$.

A variety of termination criteria may replace the one given in (4) above.

3.5.3 Evaluation of The Partial Derivatives

We next consider an approach based on productivity indices for evaluation of the partial derivatives appearing in the network model call in (3) above. Fluids inflow performance in a section of a well may be described with analytical steady state solutions for drainage from infinite volumes. Specifically, in such cases where the inflow behaves as

$$q_{\alpha k} = (PI)_{\alpha k} (p_{\alpha \infty} - p_k), \tag{3.26}$$

where ($p_{\alpha \infty}$) is a (phase) reservoir pressure at a sufficiently remote point, such that ($p_{\alpha \infty}$) is constant with respect to wellbore pressure and where (k) represents a connection in the section of the well associated with ($p_{\alpha \infty}$). With reference to the notation used in the iterative procedure, a productivity for each phase and connection point can be derived numerically:

$$(-\frac{\delta q_{\alpha k}}{\delta p_k})^{n+1,m+1} \approx PI = \frac{q_{\alpha k}^{n+1,m+1}}{P_{\alpha \infty}^{n+1,m+1} - p_k^{n+1,m}} \quad . \tag{3.27}$$

However, the convergence rate for the approximation $-(\delta q_{\alpha k}^{n+1,m+1}/\delta p_k) = (PI)_{\alpha k}$ is rather poor, and improved methods are required for efficient convergence to the correct answer.

One alteration to this procedure, which significantly improves the convergence rate, is an evaluation of partial derivatives based on the iteration variables. The procedure is illustrated in Figure 3.8.

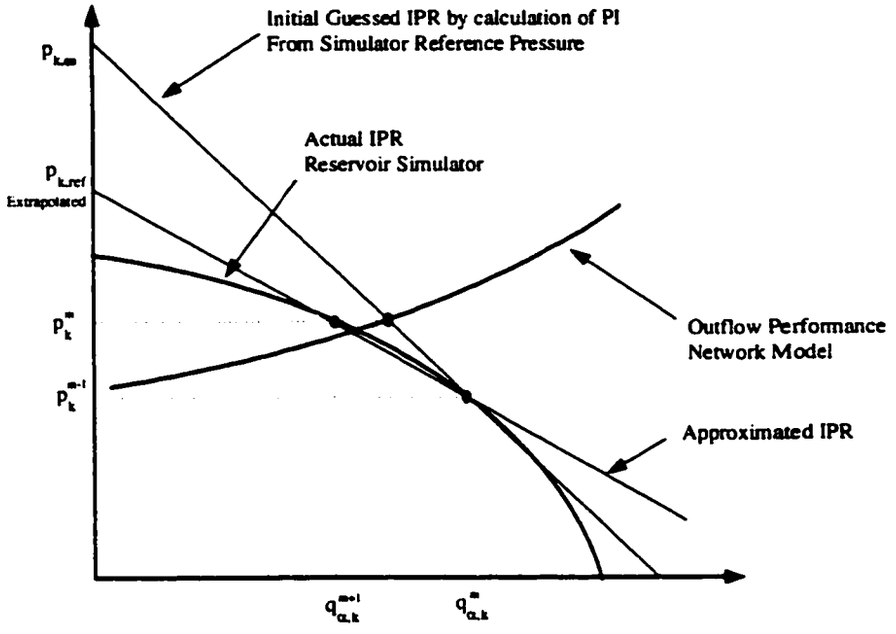

Figure 3.8: Approximation of Partial Derivative - Simulator Coupling

Estimation of the derivatives based on internal iteration variables can be defined as follows: For n>0 and m=1, take $(\delta q_{\alpha k} / \delta p_k)^{n+1,m+1}$ from the previous time step. (If n=0, use the productivity factor in formula (3.26)). For n>0 and m > 1, use

$$\left(\frac{\delta q_{\alpha k}}{\delta p_k}\right)^{n+1,m+1} = \frac{q_{\alpha k}^{n+1,m+1} - q_{\alpha k}^{n+1,m}}{p_k^{n+1,m} - p_k^{n+1,m-1}} \quad . \tag{3.28}$$

The described coupling scheme has been used for simulation of water and gas coning to horizontal wells with friction in the wellbore[8]. The coupling was confirmed to be stable, and converged in less than two iterations on the average.

CHAPTER IV

PERMEABILITY UPSCALING FOR THE NEAR WELLBORE RESERVOIR ZONE

When upscaling a heterogeneous permeability formation close to the wellbore, we require that the relationship between the total well flowrate and the drawdown between cases with upscaled formation and the case with the original, fine scaled near wellbore formation, correspond as closely as possible. For a horizontal well, in upscaling of permeabilities in the reservoir around the well, we also have some additional requirements to satisfy. The following additional physics of the flow problem must be maintained:

1) As the pressure loss along the wellbore may affect the well productivity, the inflow profile along the wellbore should be preserved through the upscaling process.
2) The pressure loss through an anisotropic near wellbore zone may be a strong function of θ. Thus, the method should allow for generation of upscaled near wellbore segments for at least four discrete portions of the near wellbore cylinder (top, bottom and sides).

The upscaling method should handle variations the original, Cartesian grid block shape and resolution along the well. To effectively upscale for undulating wells, the upscaling procedure should also handle any horizontal well location within a near wellbore zone element.

Two approaches for upscaling small scale permeability variations into a larger scale apparent permeability are the "No Cross Flow" and the "Vertical Equilibrium" upscaling

methods[10,11]. For linear flow upscaling, these methods have been combined, by performing an arithmetic averaging of the two, into what has been called the "Incomplete Layer" method.

The "No Cross Flow" and the "Vertical Equilibrium" methods are applied here for a cylindrical grid located symmetrically around the wellbore, under the assumption that the flow in the upscaled zone is predominantly radial and has reached steady-state. Thus, the flow through the near wellbore zone can be described with the steady-state equation for radial flow through porous media, given by Equation 4.1:

$$q = \frac{\alpha \cdot h \cdot k}{\mu \cdot \ln(\frac{r_{e,NWZ}}{r_w})} \cdot \Delta p \,. \tag{4.1}$$

In this equation, α is the angle of the sector ($0..2\pi$) in the θ direction; h is the length of the zone in the direction of the wellbore and $r_{e,NWZ}$ is the radius to the outer boundary of the near wellbore zone.

4.1 No-Cross Flow Upscaling

The near wellbore zone of interest is divided into separate flow sectors, as illustrated in Figure 4.1. These sectors correspond to the cylindrical grid resolution in the θ direction. The sectors are considered as isolated flow zones with no cross flow allowed in between the sectors. Thus, the radial flowrate along a sector is constant and independent of distance from the wellbore.

This assumption allows the application of Darcy's law used for flow through each sector. Each sector is then divided into serially connected annular regions (segments) from the wellbore to the outer boundary of the sector, as illustrated in Figure 4.1.

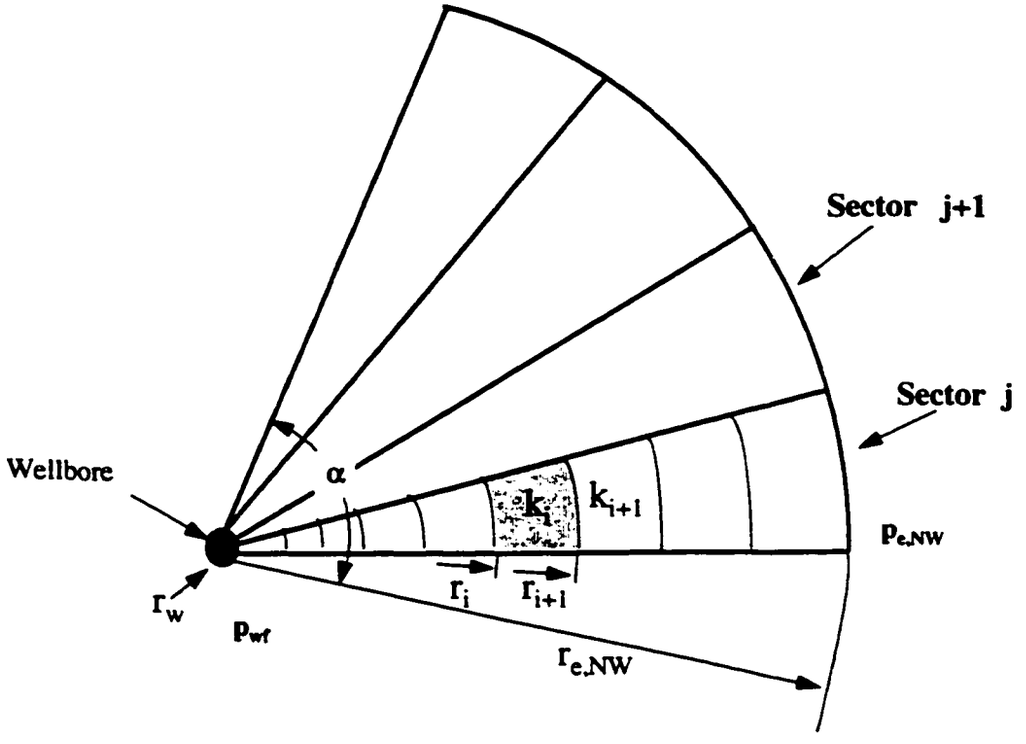

Figure 4.1 No Cross Flow Upscaling For Radial Flow

The sum of the pressure losses through the serially connected segments then add up to be the total pressure loss along the segment, as described in Equation 4.2:

$$\Delta p_{tot} = p_{e,NW} - p_{wf} = \sum_{i=1}^{n} \Delta p_i = \sum_{i=1}^{n} (p_{i+1} - p_i) \ . \qquad (4.2)$$

This relationship can now be used to determine the apparent permeability for each of the sectors. Rearranging Equation 4.1 we have:

$$\Delta p_{tot} = \frac{q\mu \ln(\frac{r_{e,NW}}{r_w})}{\alpha\, h\, k_{app.}}, \tag{4.3}$$

and

$$\Delta p_i = \frac{q\mu \ln(\frac{r_{i+1}}{r_i})}{\alpha\, h\, k_i}. \tag{4.4}$$

So by using Equation 4.2 ,

$$\frac{\ln(\frac{r_{e,NW}}{r_w})}{k_{j,app.}} = \sum_{i=1}^{n} \frac{\ln(\frac{r_{i+1}}{r_i})}{k_i} \tag{4.5}$$

we can derive an apparent sector permeability as given by the harmonic average of all segments permeabilities. The harmonic average is corrected for radial flow and is given by the following equation:

$$k_{j,app.} = \frac{\ln(r_{e,NW} \,/\, r_w)}{\sum_{i=1}^{n} \frac{1}{k_i} \cdot \ln(r_{i+1} \,/\, r_i)}. \tag{4.6}$$

The other major assumption applied in the "No-Cross Flow" upscaling method is a uniform outer boundary pressure for the volume of investigation. The total pressure loss along all segments are, thus, set to be equal. The flowrates for all individual segments add up to be the total flowrate from the near wellbore zone of interest.

The constant boundary pressure and the flowrate relationships can be used to determine the apparent permeability for an entire cylindrical element of the near wellbore zone or for a particular angular slice of the near wellbore zone element. If we use Equation 4.1 to describe the addition of sector flowrates, the following relationship gives the apparent permeability of the upscaled near wellbore zone element:

$$k_{app} = \frac{1}{m} \sum_{j=1}^{m} k_j \quad . \tag{4.7}$$

By combining equations 4.6 and 4.7 we arrive at the following final expression for the apparent no cross flow permeability:

$$k_{app} = \frac{1}{m} \sum_{j=1}^{m} \frac{\ln(r_e / r_w)}{\sum_{i=1}^{n} \frac{1}{k_i} \ln(r_{i+1} / r_i)} \quad . \tag{4.8}$$

However, the application of this method requires that the permeability field can be described by a cylindrical grid. This is facilitated by applying the method described in Section 4.3.

4.2 Vertical Equilibrium (Full Cross Flow) Upscaling

The previously described "No Cross Flow" upscaling method constituted the limiting case where flow only occurs in the radial direction and no flow occurs in the θ and well direction. The "Vertical Equilibrium" (Full Cross Flow) method is valid for the other extreme flow conditions where maximum amount of possible cross flow in the θ direction is allowed to occur.

This method assumes that we have circular iso-potentials at any distance away from the well. A schematic illustration of the methodology is given in Figure 4.2.

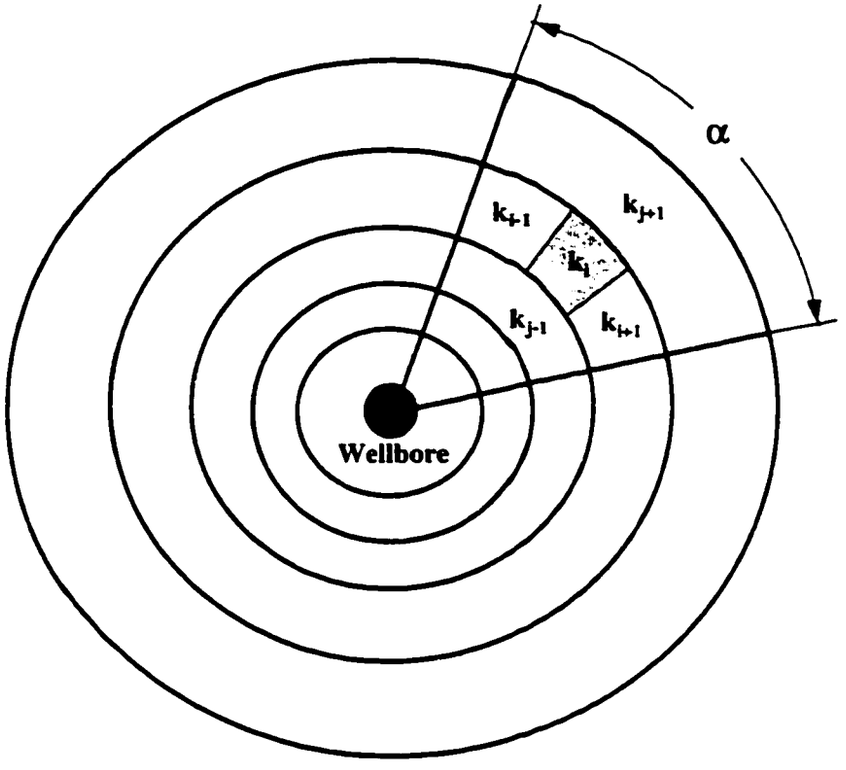

Figure 4.2 Vertical Equilibrium (Full Cross Flow) Upscaling For Radial Flow

The near wellbore zone is divided into annular regions from the wellbore to the outer boundary. Each annular band has a constant pressure loss with respect to θ. Then, each annulus is divided into smaller segments in the θ direction, and the sum of the flowrates for all the segments add up to the total flowrate through the annulus, as given by the following equation:

$$q_{j,tot} = \sum_{i=1}^{n} q_i \ . \tag{4.9}$$

Thus, the apparent permeability of an annular space in the permeability field can be found by combining Equations 4.1 and 4.9. The apparent annulus permeability is a volume weighted arithmetic average of the segment permeabilities within the annulus, as given by Equation 4.10:

$$k_j = \frac{1}{n} \cdot \sum_{i=1}^{n} k_i \ . \tag{4.10}$$

We now assume steady state flow throughout the near wellbore zone and that the sum of all the pressure losses through the serially connected annuli adds up to the total pressure loss from the wellbore to the near wellbore zone boundary. Thus, the apparent permeability for either the whole or for a particular angular slice of the near wellbore zone can be derived to be the harmonic average of all the apparent annulus permeabilities as given by the following equation:

$$k_{app} = \frac{\ln(r_e / r_w)}{\sum_{j=1}^{m} \frac{1}{\frac{1}{n} \cdot \sum_{i=1}^{n} k_i} \cdot \ln(r_{j+1} / r_j)} \ . \tag{4.11}$$

4.3 Transformation From a Cartesian to a Radial Grid System

As presented in the previous sections, the upscaling for radial flow in the near wellbore reservoir results in a straight forward and simple procedure as long as we operate on a cylindrical grid system. However, the permeability field we have to work with is more than likely to be presented in a Cartesian coordinate system. (The theoretical tools for reservoir characterization and simulation have been developed for a Cartesian system.) Thus, in order to perform upscaling for the convergent flow around the wellbore, it was necessary

to develop theory for transformation of the Cartesian based permeability field to a cylindrical coordinate system.

The basis for the transformation is to develop an apparent permeability for the cylindrical grid block which is made up of various heterogeneous Cartesian grid blocks. If we observe an arbitrary radial grid block, as illustrated in Figure 4.3, it is apparent that a radial block can be sliced into a variety of smaller, odd shaped volumes by the Cartesian grid system. The geometric shapes and volumes of the Cartesian block contributions will depend on the rotation of the cylindrical grid block with respect to the Cartesian system and the relative resolution of the Cartesian to the radial grid.

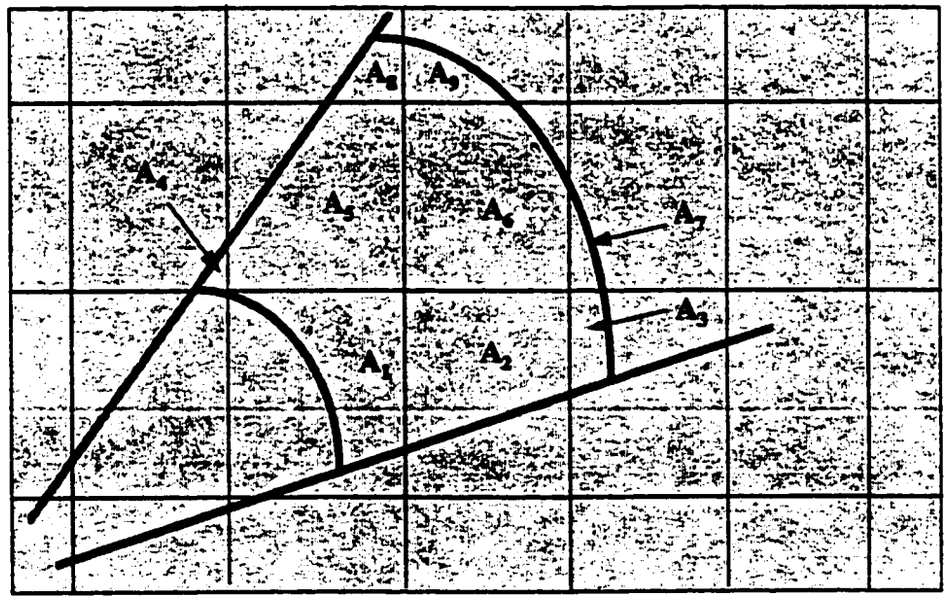

Figure 4.3 Radial Block Permeability Volumetric Averaging

As the pressure gradient is the largest close to the wellbore, the cylindrical grid block size is increased exponentially when moving away from the wellbore in the radial direction. Thus, even if the Cartesian grid block size may be constant, the radial grid block size varies throughout the entire near wellbore zone, and a wide range of relative block sizes are encountered.

The transformation routine attempts to calculate the volume of each Cartesian grid block contributing to the radial block in question and we apply these volumes as weighting factors in the averaging procedure. The contributing Cartesian blocks and the cylindrical blocks have the same length in the direction of the well. Thus, an area weighted average for the apparent radial block permeability given by the following expression applies:

$$k_r = (1 / A_r) \cdot (k_1 A_1 + k_2 A_2 + \ldots\ldots + k_n A_n), \tag{4.12}$$

where A_r is the area of the cylindrical element of the sector.

The calculation of areas in this conversion is the most CPU time demanding operation in the upscaling process, if performed for each of the several thousand cylindrical blocks usually present in the near wellbore zone. However, the area calculation may be reduced if both the Cartesian grid, cylindrical grid and well location are constant along the horizontal well. If this is the case, the areas should only have to be calculated for only one block length in the well direction. Thus, the CPU time can be reduced by a factor equal to the number of blocks in the well direction. In addition, when several realizations for near wellbore geology are upscaled simultaneously, the area calculation may be performed only once.

A complete description of this method is given in Appendix A, and FORTRAN source code programmed to perform the conversion is given in Appendix B.

4.4 Upscaling for Anisotropic Formation

A critical factor determining the productivity of a horizontal well is the degree of anisotropy present. If the communication in the vertical direction is poor, the horizontal well productivity will be severely reduced.

The large scale anisotropy caused by layers of different permeability is automatically taken into account through the upscaling process when calculating an apparent permeability from the fine grid heterogeneous permeability field.

However, anisotropy on a smaller scale has to be accounted for by considering the directional permeabilities throughout the near wellbore zone. Thus, a procedure for upscaling the near wellbore formation with different ratios of vertical to horizontal permeabilities was implemented in the radial upscaling scheme[10]. A schematic illustration of the anisotropic permeability problem is given in Figure 4.4.

The starting point for the anisotropic upscaling is the directional permeabilities for each Cartesian grid block. When applying the area based averaging procedure to determine the radial grid block permeabilities, a directional radial permeability can be calculated for the radial grid block from the x and y Cartesian system block permeabilities. The radial grid block orientation with respect to the wellbore is now, unlike for the isotropic case, of great importance as each Cartesian block contribution to the radial permeability is dependent on its direction from the wellbore. For the upscaling of radial flow around the wellbore, only the permeability in the vertical direction (z) and the horizontal direction

perpendicular to the wellbore (x) is used. The horizontal permeability parallel to the well (y) is not applied in this process unless upscaling also is performed in the well direction.

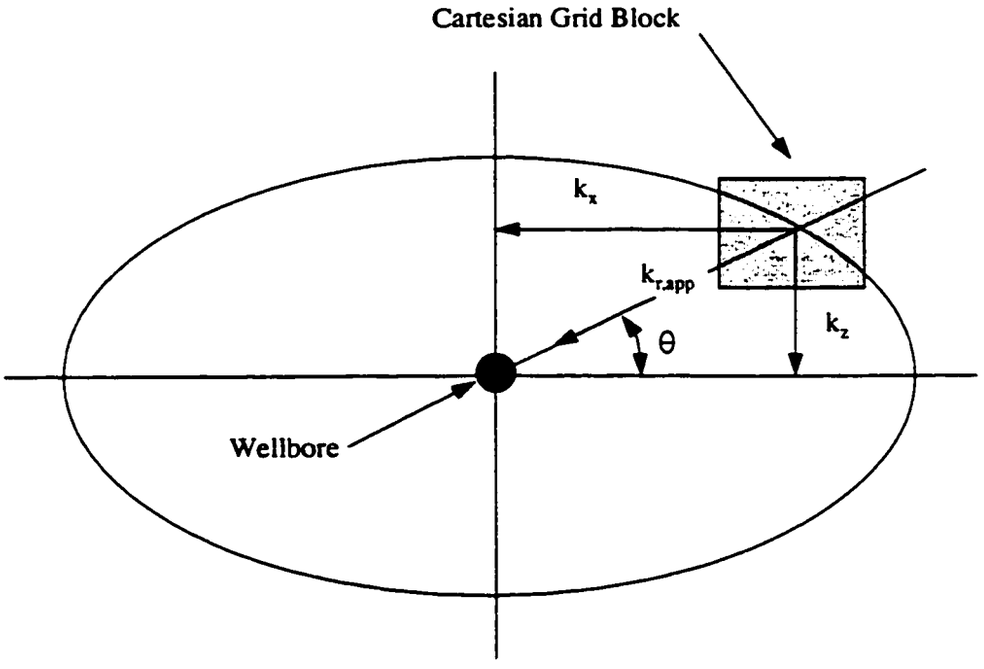

Figure 4.4 Apparent Anisotropic, Radial Permeability , Schematics

Rotation of the coordinate system can now be applied. The diagonal tensor elements for the permeability in the x, z coordinate system perpendicular to the wellbore are given by Equations 4.13 and 4.14:

$$k_{xx} = k_x \cdot \sin^2(\theta) + k_z \cdot \cos^2(\theta), \tag{4.13}$$

$$k_{zz} = k_z \cdot \sin^2(\theta) + k_x \cdot \cos^2(\theta). \tag{4.14}$$

Since, for our case, the principal directions of anisotrophy are othogonal, the off-diagonal tensor elements are equal and given by Equation 4.15 :

$$k_{xz} = k_{zx} = \sin(\theta) \cdot \cos(\theta) \cdot (k_z - k_x) \tag{4.15}$$

The directional, apparent permeability towards the wellbore can now be found by combining the tensor elements according to the following equation[13]:

$$k_{r,app} = k_{zz} - \frac{k_{xz} \cdot k_{zx}}{k_{xx}}. \tag{4.16}$$

By applying Equations 4.13 - 4.15 in Equation 4.16 and rearranging, the final relationship for the permeability in the direction of the wellbore is given by Equation 4.17:

$$k_{r,app} = \frac{k_z \cdot k_x}{k_x - k_x \cdot \cos^2(\alpha) + k_z \cdot \cos^2(\alpha)}. \tag{4.17}$$

The angle θ is measured from the Cartesian x axis to the line drawn from the grid block of interest towards the wellbore. The apparent radial permeability given in Equation 4.17 is valid only for principle directions of anisotropy corresponding to the Cartesian x and z directions.

CHAPTER V

MODEL VERIFICATION

The different modules of the developed semi-analytical network model were systematically verified against results from laboratory flow experiments and against fine grid reservoir simulations using the Eclipse numerical reservoir simulator. The verification was aimed at the following particularly critical elements:

a) Verification of the acceleration pressure loss term used in the wellbore pressure loss model by comparison to experimental data.
b) Verification of steady-state flow assumption in the near wellbore zone.
c) Verification of the radial flow upscaling procedure used for single phase flow in the near wellbore reservoir zone.
d) Investigation of the radial flow upscaling procedure applicability for multiphase flow in the near wellbore reservoir zone
e) Verification of the semi-analytical outer reservoir response approach.

5.1 Experimental Verification of Acceleration Pressure Loss for Single Phase Flow

In this study the acceleration pressure loss was investigated through single phase water flow experiments. The experiments were performed in an outdoor flow loop at Norsk

Hydro's research facilities in Porsgrunn, Norway. In the water and air flow loop, the system temperature was 7 °C, and the maximum absolute system pressure was 600 kPa Two parallel flow sections were arranged with common supply lines for water and air. A schematic illustration of the test loop is given in Figure 5.1.

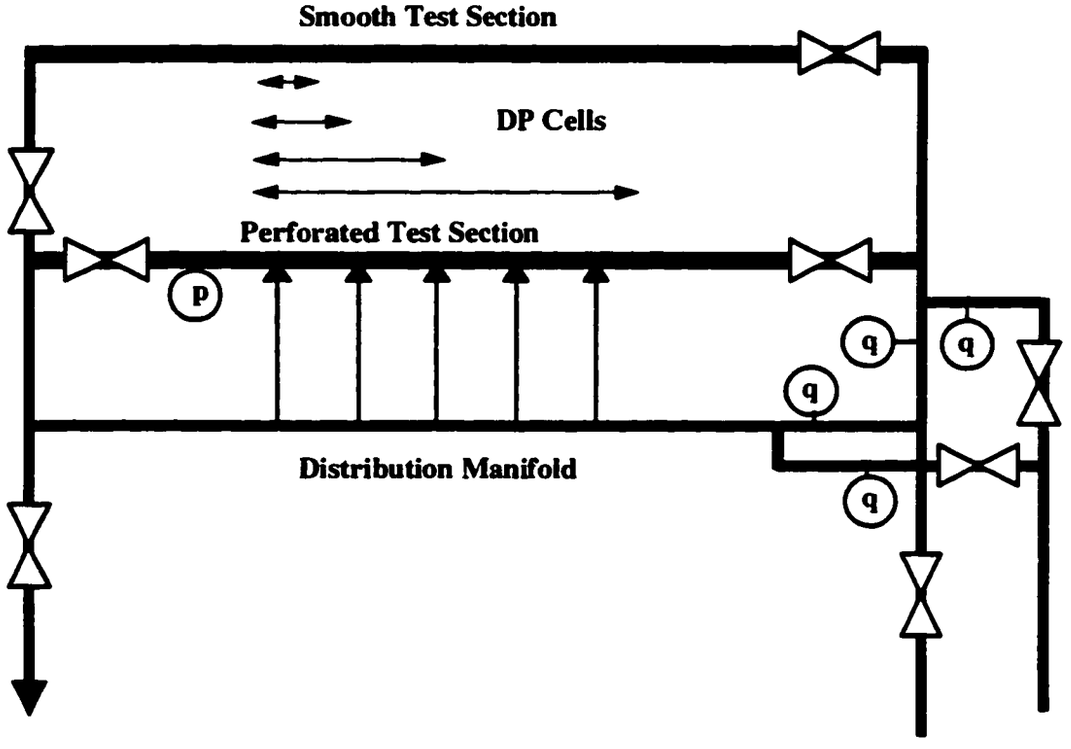

Figure 5.1: Test Loop - Schematic Layout

The two test sections were 15 m long, had internal diameters of 0.15 m and had inlet and outlet sections of 10m. One of the test sections was equipped with 30 branches for radial inflow of either water or air. The inlets were connected to the test section through

0.010 m diameter perforations in the pipe wall, all located at the top of the pipe with 0.5 m spacing.

Real production liners have radial inflow around the entire circumference of the pipe, and a variety of perforation densities. As the effects from accelleration of the radial inflow rather than the added roughness from a large number of perforations were to be investigated, the perforation density was chosen to be small. Each inlet was connected to the distribution manifold with rubber hoses, and equipped with a fixed flow characteristics pressure reduction device. The system was designed to handle single phase radial inflow only. Both the test section and the distribution manifold were connected to the supply lines for water. The water flowrates were controlled with manual regulating valves. The flowrates were logged with mass flow meters. The gauge pressure was measured at the end of the test section and pressure differences were measured from the end of the section to four separate locations.

A PC based data acquisition system was used to gather data with a frequency of two measurements per second. An average of the last fifteen seconds of data was written to disk in two second intervals over a two minute period. These data were again averaged to form the final measurement. A computer model was developed to calculate the correct material and momentum balance within the test loop and in this way account for the additional friction due to inflow. The frictional pressure losses were calculated between the perforations by applying an absolute roughness of 0. 001 mm and rough pipe friction factor. The pipe roughness was found by matching the model to the no radial inflow cases. The radial inflow velocity from each perforation was calculated rigorously by using a

calibrated relationship for the radial flow vs. the pressure difference across the radial inlets. The model included a separate term (Equation 3.19)

$$\Delta p_{acc.} = \frac{(\dot{m}_d v_d - \dot{m}_u v_u)}{A}$$

to account for the change in momentum taking place at each perforation.

Measured and calculated pressure losses for the full length of the test section vs. radial superficial velocities are given for different values of superficial axial inlet velocities in Figure 5.2. A good match was obtained between the calculated and the measured data by using the relationship for acceleration pressure loss given in Equation 3.19.

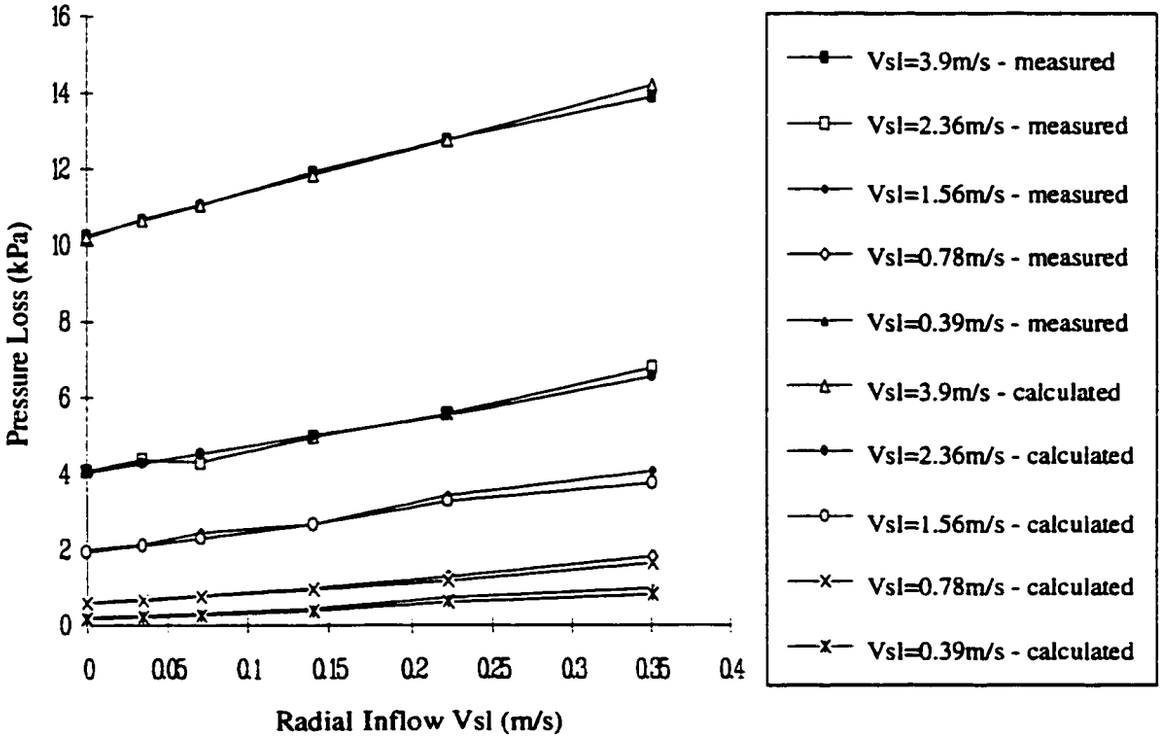

Figure 5.2: Measured pressure loss vs. radial inflow (Vsl) , axial Vsl varied.

5.2 Validation of Steady State Flow Assumption in the Near Wellbore Reservoir

To determine the validity of the steady state assumption used in the upscaling of the near wellbore zone, simulations were run for a three dimensional reservoir model with a 200 m long horizontal well. The reservoir model had a 21 x 40 x 20 (1000 m x 1210 m x 101 m) Cartesian grid where a 11 x 20 x 11 block, 1 m resolution grid extended 5.5 m away from the well. The fine grid near wellbore region was configured as six 1 m thick regions completely enclosing each other. As a measure of the development of steady-state flow, the difference in mass flow rates across the near wellbore region was calculated versus time and as a fraction of the total well mass flowrate. Cases of different permeability and number of phases were run.

As a measure of the time until steady-state flow was obtained we selected the time at which the difference in mass flowrates in and out of the near wellbore region had been reduced to 1 % of the well mass flowrate. All of the high permeability (1000 md) single and multiphase flow cases showed that steady state flow based on the above criteria occurred in a fraction of a day. However, the low permeability (10 md) cases are treated in more detail as they require considerably longer time before a steady state zone develops.

Figure 5.3 shows the results for three homogeneous cases with permeability of 10 md. The oil production rate was initially specified as a constant 1000 Sm^3/d until a minimum bottom hole pressure of 1.0 bar was reached. Thereafter, the well was produced at a constant bottom hole pressure.

The following cases were simulated:

a) Live Oil, (S_g critical = 10 %) b) Dead Oil c) Live Oil (S_g critical = 0.0 %)

Both cases b) and c) showed a 1.0 % difference in the mass flow after 1 day. However, the difference in cumulative mass flow dropped to 1.0 % at 30 days. The case with live oil and critical gas saturation for flow set to 10.0 % had a 1.0 % deviation in mass balance across the near wellbore zone at 5 days based on instantaneous rates and a 1.0 % deviation in cumulative mass balance after 100 days. The slower development of a steady state zone is due to an increase in gas saturation in the near well bore region before the critical gas saturation for flow is reached and gas starts to flow. Thus, this represents an accumulation of gas in the near wellbore region.

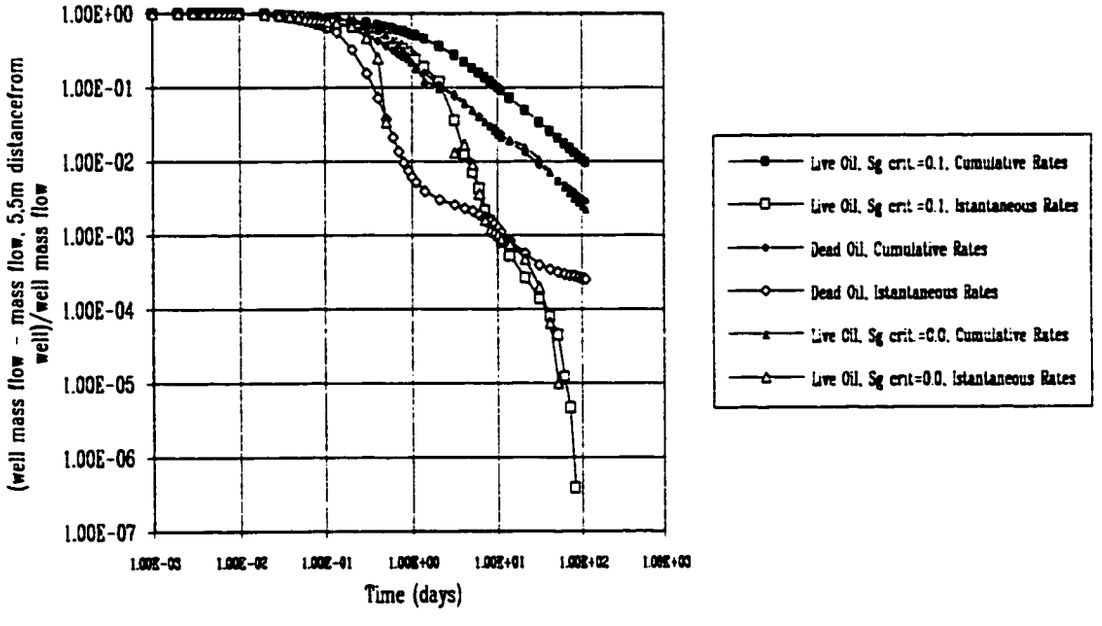

Figure 5.3: Fractional Difference in Mass Balance Across the Near Wellbore Region vs. Time.

To illustrate how the steady state approach can be applied to the near wellbore region, Figure 5.4 indicates that the relationship between total mass flowrate and pressure loss across the near wellbore region for the dead oil case. This relationship becomes approximately constant at the beginning of steady state flow. Thus, the problem is to find these constants for specific locations along the well, assuming that the steady state relationships are valid in the presence of near wellbore heterogeneity.

The application of a steady-state method is questionable for multiphase flow scenarios due to the effects of accumulation of gas in the reservoir near the wellbore, and possible grid effects combined with upscaling to large grid blocks close to the wellbore. The feasibility of upscaling near wellbore geology for multiphase coning scenarios is further investigated in Section 5.6.

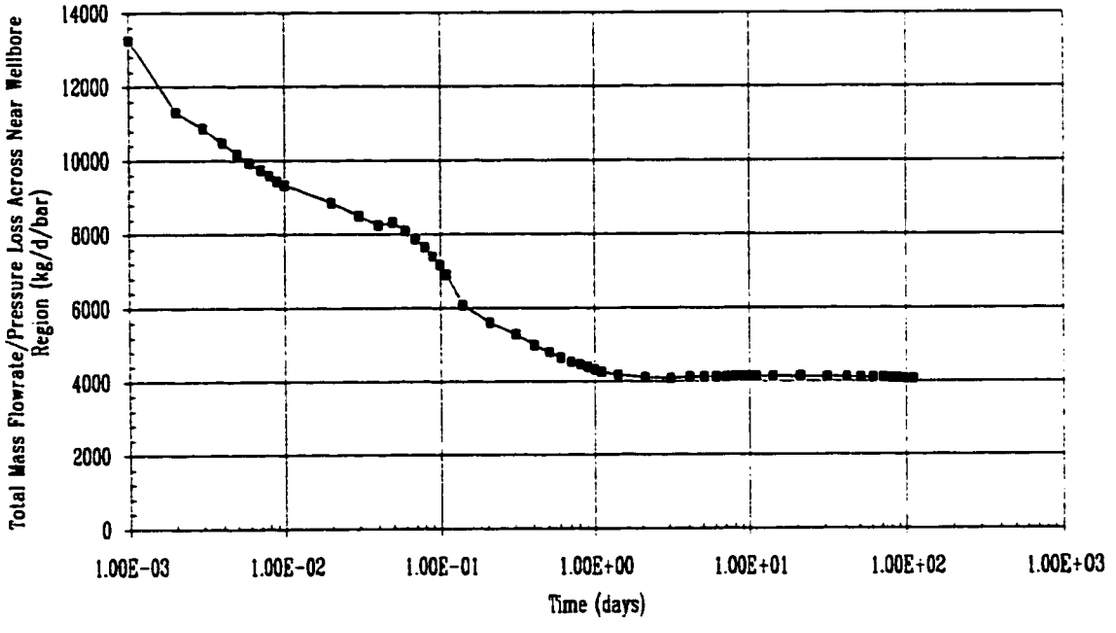

Figure 5.4: Total Mass Flowrate / Pressure Loss Across Near Wellbore Region vs. Time

5.3 Permeability Upscaling of a Heterogeneous Near Wellbore Zone

The integrity of the presented upscaling procedure was investigated by comparing the results obtained from reservoir simulations with upscaled permeabilities in the near wellbore reservoir zone and the network model simulations, to the results from reservoir simulations with fine scaled heterogeneous near wellbore zone. The two areas of particular interest in the comparison were the total well productivity and conservation of the inflow profile along the wellbore.

A simulation model in Eclipse with a fine grid and original permeability distribution in the near wellbore zone represented the "true case". Coarse and fine grid reservoir models with upscaled permeabilities around the wellbore and the network model with upscaled permeabilities were run under otherwise similar conditions.

The heterogeneous near wellbore zone had permeabilities ranging from 1.5 md to 3000 md, and a heterogeneity coefficient (standard deviation of permeability divided by mean of permeability) of 2.24 which could be expected from a well penetrating different facies. The near wellbore zone which originally consisted of 7260 grid blocks was reduced to 60 blocks. Thus, the coarsening of the grid was considerable, upscaling by a factor of 121. The fine and coarse grids had the same resolution in the direction of the well. Complete input data for the fine grid Eclipse base case are given in Appendix C. The base case used for the verification had the following characteristics.

Reservoir: Absolute permeability outside near wellbore zone - 100 md

Porosity - 20 %

Endpoint relative permeability for oil - 0.85

Fully penetrating well

Fine grid - Cartesian 15 x 60 x 15 blocks

(213 m x 300 m x 213 m)

Coarse grid - Cartesian 5 x 60 x 5 blocks

(213 m x 300 m x 213 m)

Partially penetrating well

Fine grid - Cartesian 15 x 64 x 15 blocks

(213 m x 510 m x 213 m)

Coarse grid - Cartesian 5 x 64 x 5 blocks

(213 m x 510 m x 213 m)

Near wellbore zone: Fine grid - Cartesian 11 x 62 x 11 blocks (11 m x 310 m x 11 m)

Coarse grid - Cartesian 1 x 62 x 11 blocks (11 m x 310 m x 11 m)

Average permeability - 100 md, Porosity - 20 %

Variance of permeability distribution - 50,000 md^2

Spatial distribution from simulated annealing

Spherical semi-variogram

Range: Rx = 20 m, Ry=20 m, Rz=5 m

Horizontal well: Length - 300 m

Wellbore radius - 0.1 m

Centrally located in the reservoir along the Y direction

Number of connections between well and reservoir - 60

Fluids: Oil, μ_o = 1.82 cP and ρ_o=811 kg/m^3 (at reservoir conditions)

The simulations considered were:

1) Eclipse with fine grid heterogeneous near wellbore zone - true case
2) Eclipse with fine grid, upscaled near wellbore zone permeabilities
3) Eclipse with coarse grid, upscaled near wellbore zone permeabilities
4) Network simulator with upscaled near wellbore zone permeabilities

For the cases with upscaled permeabilities (2)-(4), the following two methods for permeability upscaling were applied:

a) No-cross flow upscaling
b) Incomplete layers upscaling $(k_{nc}+k_{ve}) / 2$

The simulations were run with a uniform pressure along the well and with a constant oil production rate of 100 Sm^3/d. for a period of 11 days.

5.4.1 Fully Penetrating Horizontal Well

Prior to upscaling of permeabilities, the permeability field was converted from a Cartesian to a cylindrical reservoir grid. The method for this conversion is given in Appendix A with a FORTRAN source code in Appendix B. As the upscaling procedure was developed to perform upscaling of permeabilities in a cylindrical grid, the procedure reflects the physics of the near well reservoir flow along the middle parts of the well. However, close to the end points of the well, the flow is not strictly radial and a spherical model would be required for a more accurate upscaling. Thus, to eliminate the potential errors from end contributions, a fully penetrating horizontal well was first considered.

Figure 5.5 presents the error in drawdown resulting from replacing the original permeability field with the upscaled permeabilities while still using the original fine grid. The upscaled, fine grid case was run to eliminate potential error from a coarsening of the grid. Both the results for the incomplete layers method and the no cross flow method are shown.

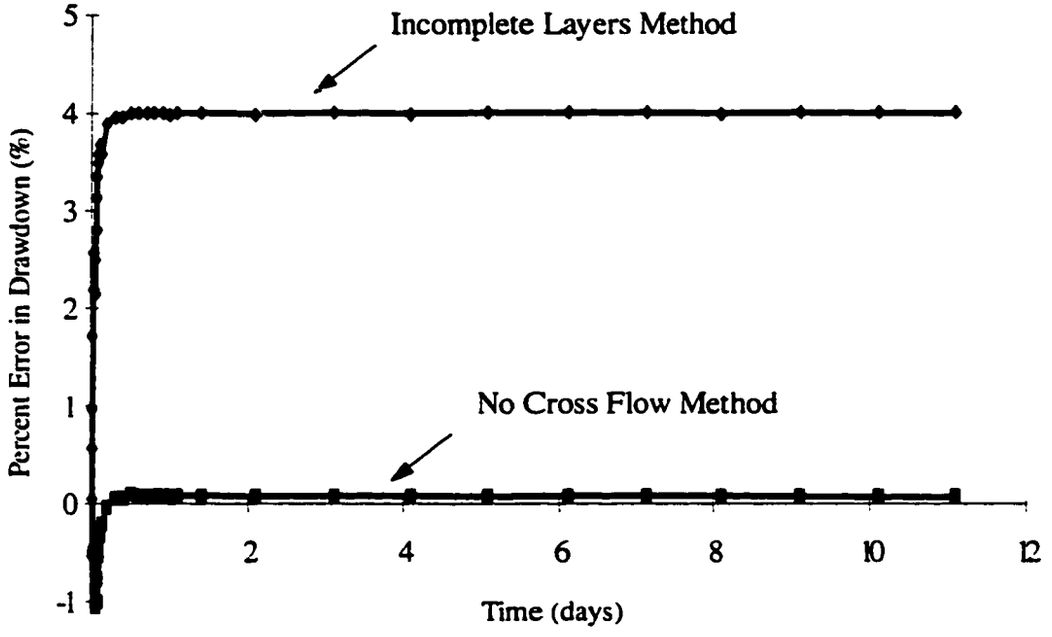

Figure 5.5: Error in Drawdown vs. Time - Fully Penetrating Horizontal Well, Fine Grid

The error in drawdown changes rapidly for both cases during the first fraction of a day, but stabilizes at 0.5 days at a value of 4 % for the vertical equilibrium case and at 0.05 % for the no cross flow case. The errors during the first 0.5 days of production are expected and are due to transient flow occurring in the near wellbore zone while steady-state flow assumptions were used in the upscaling process. In Section 5.2 it was shown that the near

wellbore zone went through an early transient period before attaining steady-state flow behaviour. The superior performance for the no cross flow method indicates that the flow around the well is close to radial.

The results of the drawdown comparison and an inspection of the stream-lines around the wellbore for the "true" case indicated that the no-cross flow method described more accurately the flow behavior in the reservoir close to the well. This could be expected as the pressure gradient in this part of the reservoir is very large in the radial direction compared to in the θ and well direction.

However, if the upscaling is to present any CPU time saving benefit, the grid has to be coarsened accordingly. Thus, rather than using the upscaled permeabilities in the original fine grid, the next step was to also replace the fine near wellbore grid of 7260 blocks with a grid consisting of 60 blocks, one block for each well connection. Each block had an upscaled, apparent permeability.

Figure 5.6 shows the error in drawdown resulting from this comparison. The same large error indicating transient behavior was seen also for this case at during the first 0.5 days of production. However, the error in drawdown increased with 1.0 % for both the incomplete layers method and the no cross flow method. Thus, a 1.0 % error can be attributed to a coarsening of the grid.

The other point of interest was to detect any distortion of the inflow profile along the well resulting from applying the upscaled permeabilities and a coarser grid close to the well. Figure 5.7 shows the oil inflow profile along the well for both the heterogeneous, fine grid case and the coarse grid case with the upscaled permeabilities.

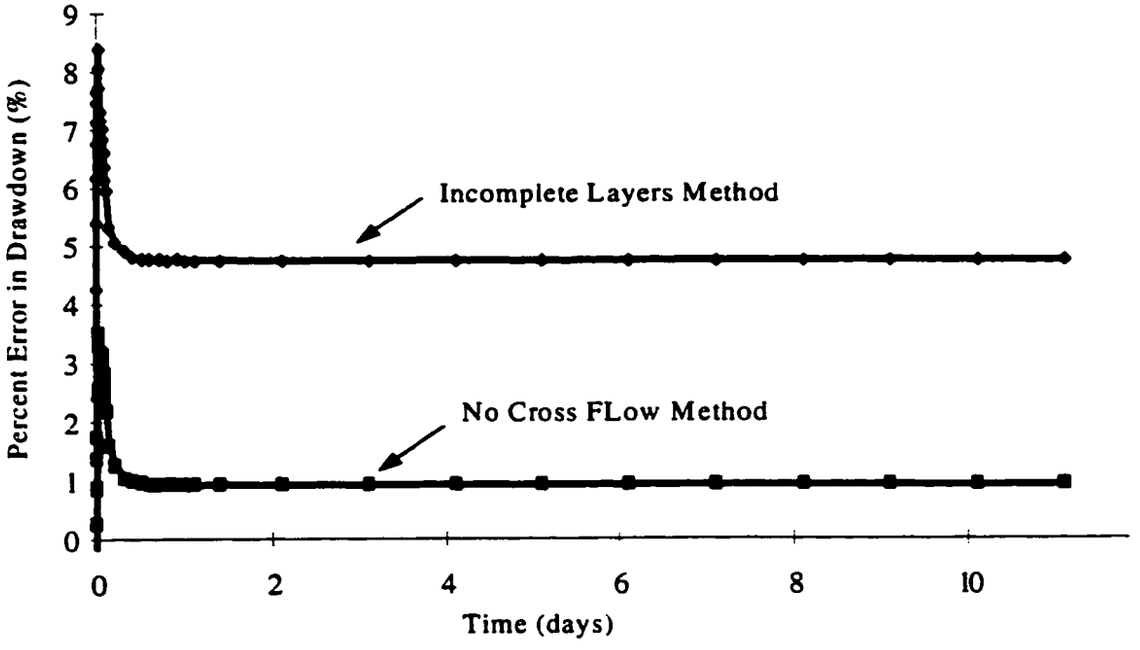

Figure 5.6: Error in Drawdown vs. Time - Fully Penetrating Horizontal Well, Coarse Grid

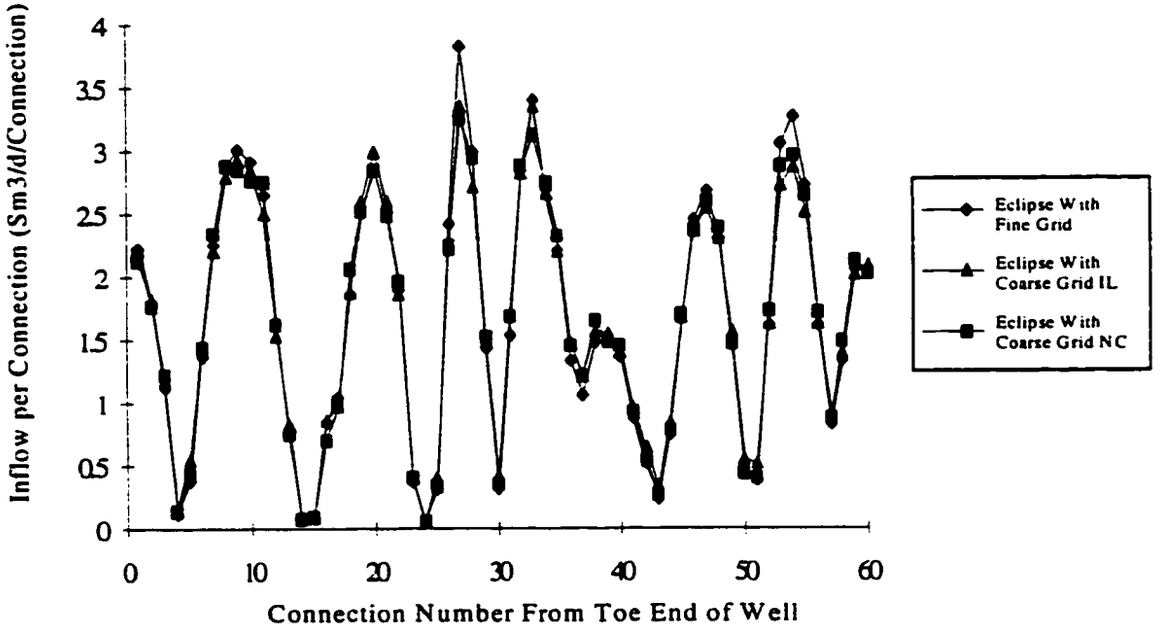

Figure 5.7: Oil Inflow Profile Comparison - Fully Penetrating Horizontal Well, Coarse Grid

Compared to the heterogeneous, fine grid case, a slight smoothening of the inflow profile is seen for the upscaled cases at connections with very high inflow. However, no appreciable difference in preserving the inflow profile along the well is observed between the two upscaled cases.

As the goal was to replace the traditional numerical reservoir simulator with a network simulator, permeabilities from the no cross flow upscaling method were applied in the network simulator and used to calculate the pressure loss through the near wellbore zone. Darcy's law for radial flow was used as the pressure loss model in the network flow connections. The main difference between using upscaled permeabilities in the network simulator and coarse grid reservoir simulations is the lack of possibility for cross flow between arcs in the network simulator. The reservoir simulator allows flow between upscaled blocks.

We wanted to eliminate any discrepancy between Eclipse and the network simulator in the handling of the outer part of the reservoir. Thus, in the reservoir model a thin (1 m) layer of grid blocks was configured right outside the near wellbore zone to allow an estimation of the pressure loss between these points and the well. The network simulator was then run with these pressures as its upstream boundary conditions rather than configuring additional flow connections describing the reservoir response. These boundary pressures were obtained from the reservoir simulations at 11 days into production. The down stream boundary condition for the network simulator was a total production rate of 100 Sm^3/d.

A 0.87 % error in drawdown was experienced for this case. Figure 5.8 shows a comparison of the inflow profile between the network simulator with upscaled apparent

permeabilities and the heterogeneous, fine grid reservoir simulation case. The oil inflow profile resulting from the network simulation is in good agreement with the inflow profile from the heterogeneous, fine grid simulations. As was observed in the previous comparison, the flowrate for high productivity connections is somewhat reduced and the flowrate for low productivity connections is increased when applying the network simulator. This is due to the network simulator's lack of cross flow ability in the direction of the well.

The correct estimation of pressure loss in the wellbore is dependent on a correct inflow profile. It should, however, be noticed that the pressure loss in the wellbore is not directly dependent on the inflow profile, but rather upon the axial flowrate along the well. Thus, the effect on the wellbore pressure loss from a small discrepancy in the inflow is increasingly dampened towards the heel part of the well.

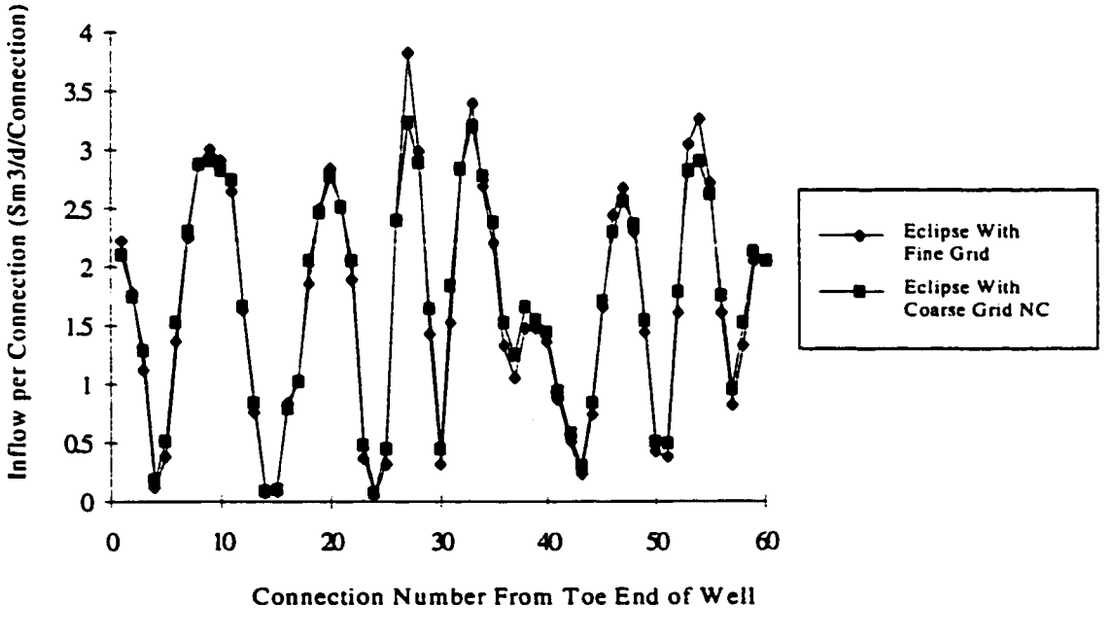

Figure 5.8: Oil Inflow Profile Comparison - Fully Penetrating Horizontal Well, Network Simulator

The discrepancy in drawdown obtained with the network simulator was 0.87 % as compared to the results from the heterogeneous, fine scaled case. Thus, some grid effects were possibly counteracted by the lack of cross flow in the network simulator.

5.3.2 Partially Penetrating Horizontal Well (End Contribution Effects)

In a field application, the well is not likely to fully penetrate the reservoir in the horizontal direction. Thus, the previous comparison only verifies the upscaling methods to be used along the middle part of the well. As the developed upscaling procedure does not fully capture the physics of the flow problem at the ends of the well, some discrepancy in the inflow profile is expected between the "true case" and the upscaled cases for partially penetrating horizontal wells. However, if the well is sufficiently long, the end effect error may not have significant impact on the total well productivity or pressure loss along the well. The reservoir was extended 100 m beyond each end of the well. Figure 5.9 illustrates the inflow profile along a partially penetrating well. A 10.0 % error in inflow rate is now present for the end connections. In this case the end connections contribute 3.0% of the total production, and will contribute even less for longer wells.

The discrepancy in drawdown between the heterogeneous and upscaled case for partially penetrating well stabilized at 0.3 % after 0.5 days. This is an improvement as compared to the results from the fully penetrating case. Thus, the error in upscaling close to the ends of the well counteracts some effects experienced from a coarsening of the grid.

It can be concluded that the limitation in modeling end contribution effects in the near wellbore zone has negligible impact on the total productivity and the inflow profile for long horizontal wells.

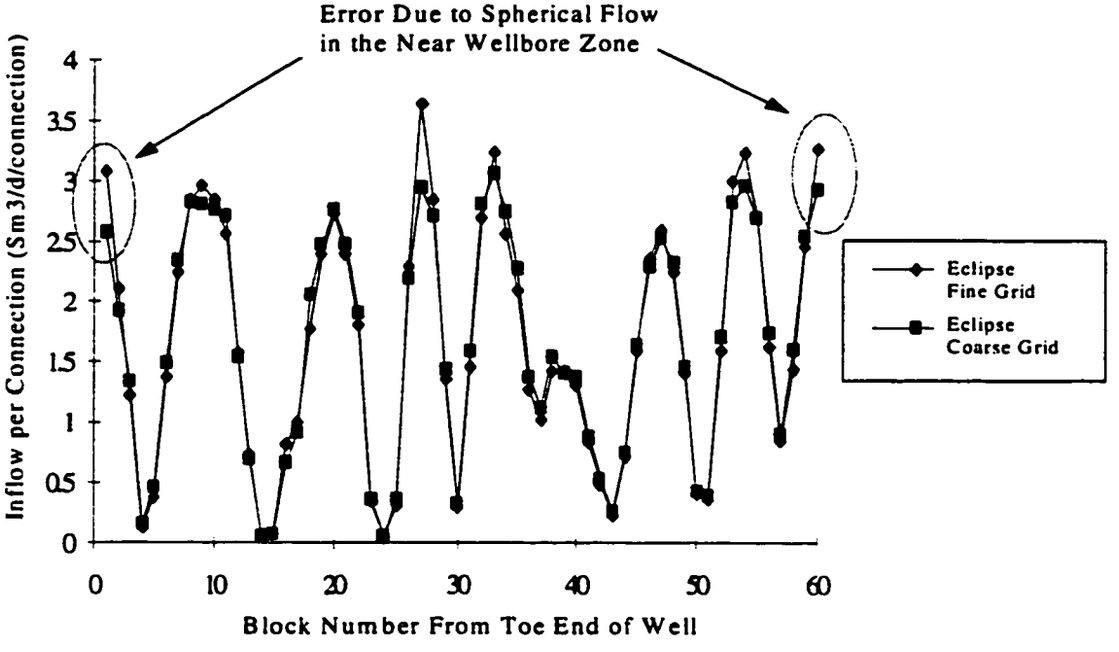

Fig. 5.9 Oil Inflow vs. Location, Eclipse with Coarse Grid and Upscaled Permeabilities versus Fine Grid and Heterogneous Permeabilities - Fully Penetrating Horizontal Well

5.3.3 Anisotropic Near Wellbore Zone

The validity of the method for implementing anisotropy in the permeability upscaling procedure was examined by comparing the results from a heterogeneous, fine grid reservoir model to the results from a coarse grid reservoir model with upscaled permeabilities for the near wellbore zone. For both cases only the near wellbore zone had anisotropic permeabilities. For the "real" fine grid case, each block in the heterogeneous

near wellbore zone had vertical permeabilities of only 20.0 % of the horizontal permeabilities.

For the upscaled, coarse grid case an apparent radial permeability was upscaled for each well block. The apparent permeability was provided for use both as x and z permeabilities in the reservoir simulator. Thus, this permeability could have been used as an apparent radial permeability in the network model for close to the same results. Figure 5.10 provides the percentage error in drawdown between the anisotropic fine grid case and the anisotropic, upscaled coarse grid case. The percentage error in drawdown (or total productivity) stabilizes at - 0.3 % at 0.5 days. Thus, the effects of anisotropy are being well represented by the proposed procedure. Again, the inflow profiles were compared and found to agree well. As can be seen from Figure 5.11, no additional errors were seen due to the presence of anisotropy.

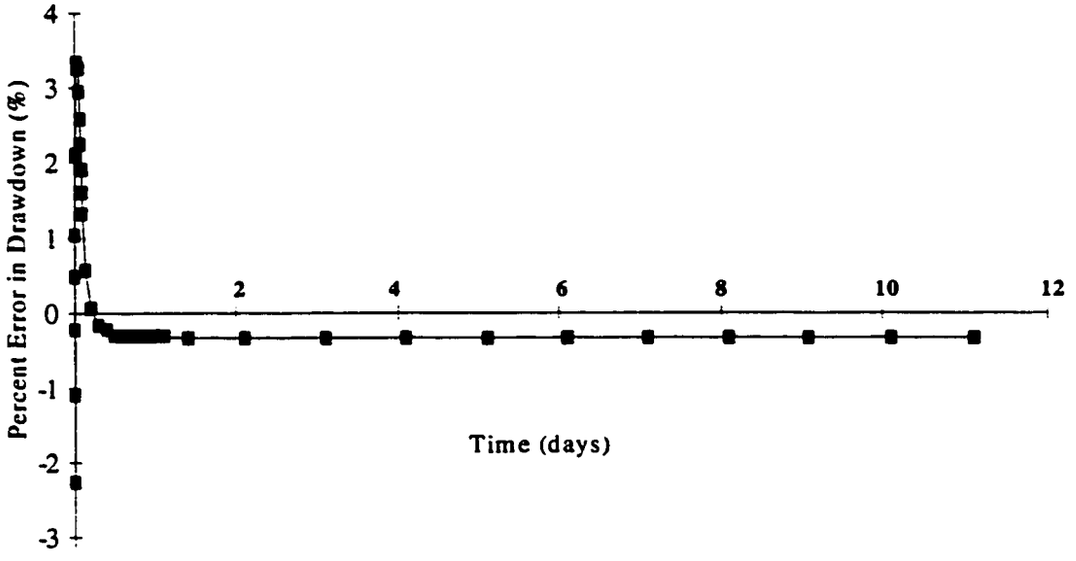

Figure 5.10: Error in Drawdown vs. Time - Fully Penetrating Horizontal Well, Coarse Grid - Anisotropic Case (kv/kh)=0.2

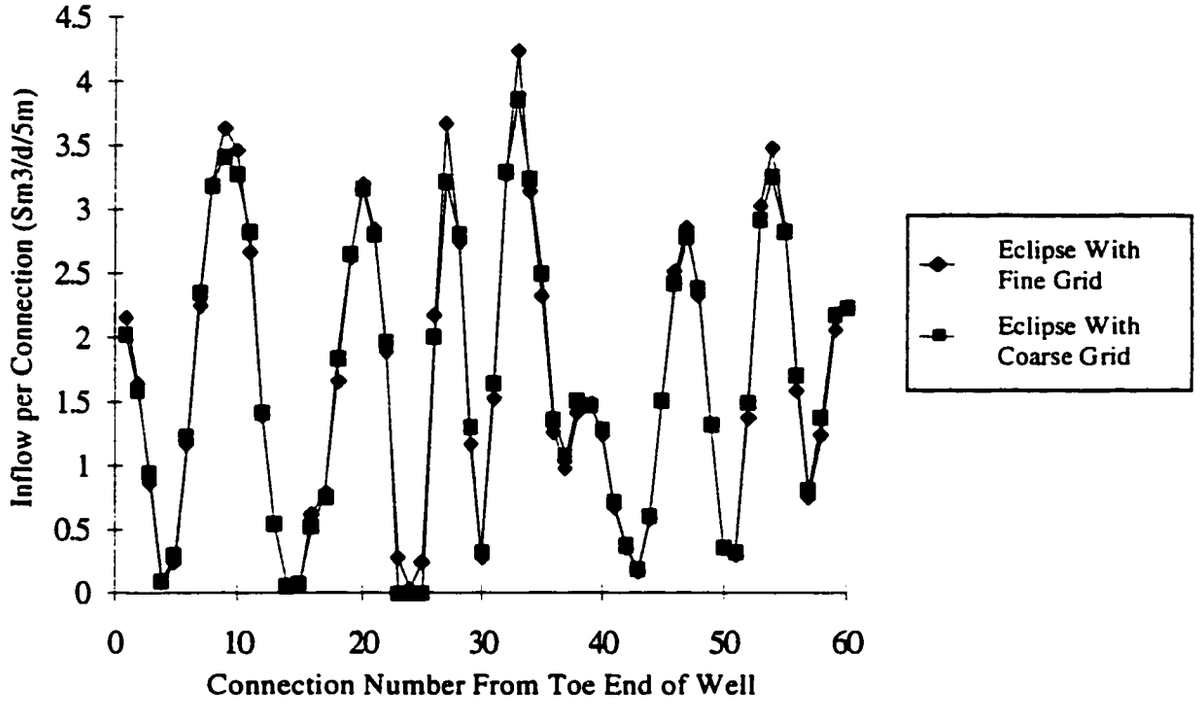

Figure 5.11: Oil Inflow Profile Comparison - Partially Penetrating Horizontal Well, Coarse Grid - Anisotropic Case, (kv/kh) = 0.2

5.4 Pseudo Steady State Reservoir Response from Superposition in 3D Space

The superposition scheme for implementation of the outer reservoir response in the network simulator was tested versus the performance of Eclipse using a model with a more refined outer reservoir grid than used in the previous cases. The 100 m extension of the reservoir in all directions from the well was, in this case, divided into 25 m blocks. Since it had already been verified that the upscaled representation of the near wellbore zone gave satisfactory results, the coarse near wellbore grid reservoir model was used as the "true case" in the comparison.

A homogeneous case with a permeability of 100 md, with uniform pressure along the wellbore and with a constant production rate of 100 Sm^3/d was simulated with Eclipse. Both the pressures and flowrates along the interface between the near wellbore zone and the outer reservoir were recorded. Using the grid block pressures along the boundary of the reservoir, an average reservoir boundary pressure was calculated from the pressures of the boundary grid blocks. From this information, the productivity of the outer reservoir could be calculated for all connections along the well.

The same case was configured in the network model with a total of 180 flow connections for the well, near wellbore zone and the outer reservoir. The average reservoir pressure and the well pressure recorded from the reservoir simulation were applied as boundary conditions. The apparent radii of the spherical wellbores used for the outer reservoir flow connections were adjusted until a total flowrate of 100 Sm^3/d was obtained. An apparent wellbore radius of 7.0 m matched these results.

Figure 5.12 shows a comparison of the outer reservoir productivity profiles at 111 days resulting from both Eclipse and the network solver. The productivity profiles correspond satisfactorily and show that the applied superposition scheme captures the physics of the outer reservoir flow for the homogeneous, uniform wellbore pressure case.

In addition to verifying the method for a simple, homogeneous case, this exercise was performed to obtain the apparent radii for the elements of the near wellbore zone for later use in more advanced cases with heterogeneous reservoir and pressure loss along the completion.

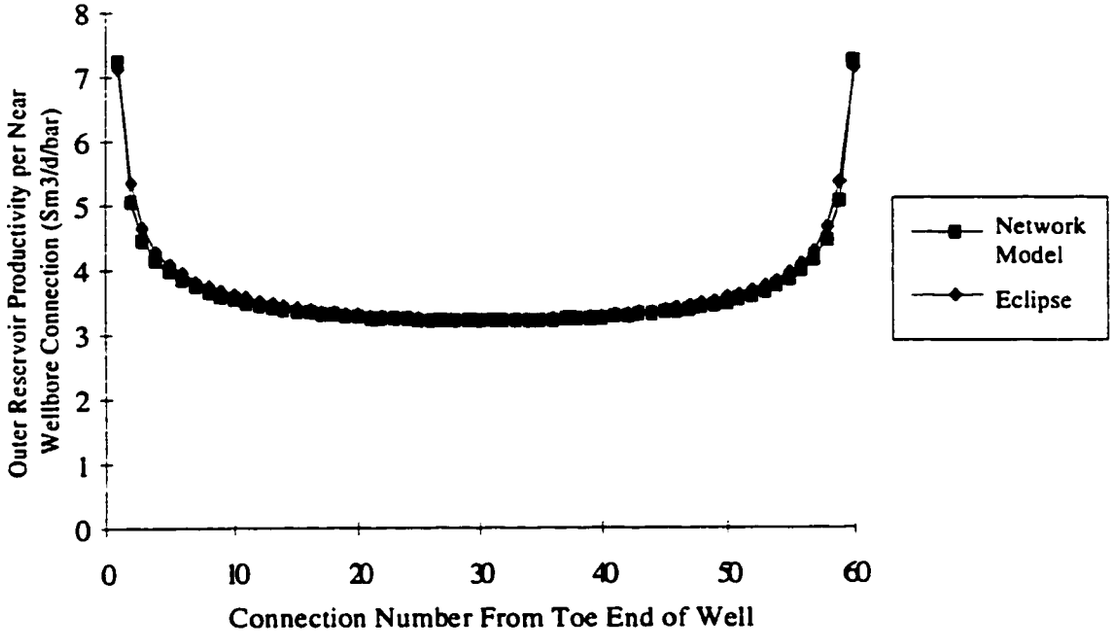

Fig. 5.12: Outer reservoir Productivity Profile Comparison - Eclipse and Network Solver, Constant Pressure Boundary

The case with a heterogeneous near wellbore zone was then tested to verify that the superposition scheme would work also for arbitrary inflow and pressure profiles along the near wellbore zone. The network model was configured with the upscaled apparent permeabilities representing the heterogeneous near wellbore formation and the apparent wellbore radii previously found from the homogeneous case.

The reservoir simulator was now run for 111 days with a constant production rate of 100 Sm^3/d. The average reservoir boundary pressure and the wellbore pressure recorded from the reservoir simulation at the end of simulation were applied as boundary conditions for the network model. With these boundary conditions, the total production rate from using the network simulation was 99.7 Sm^3/d. This was within 0.3 % of the target

flowrate used in Eclipse. Figure 5.13 shows a comparison of the inflow profiles along the well for both simulators.

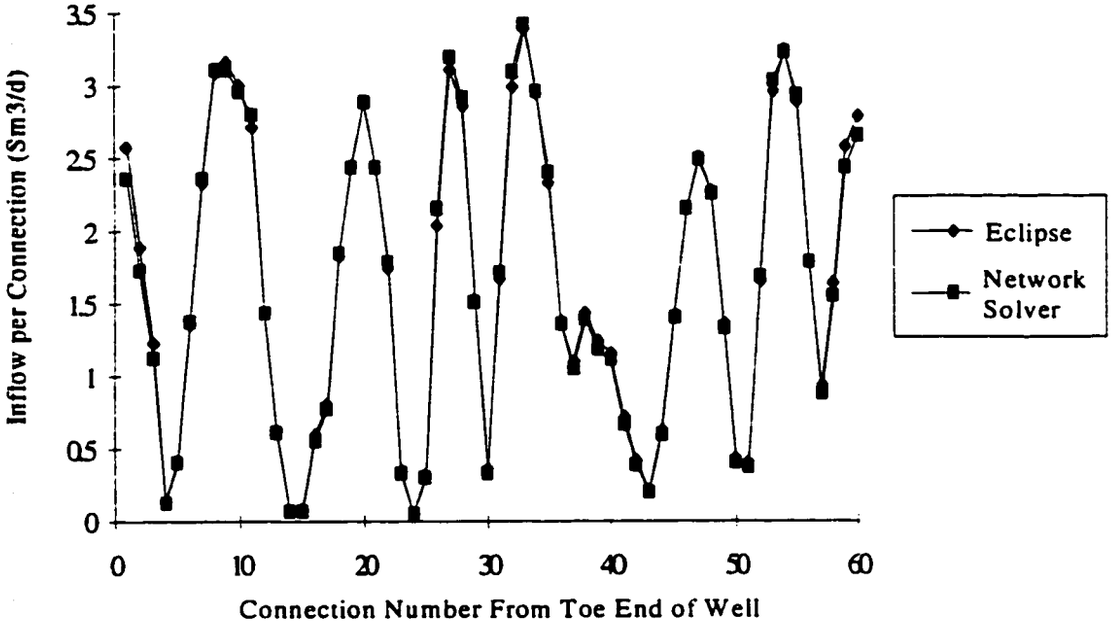

Fig. 5.13: Oil Inflow vs. Location Along Well, Comparison. Partially Penetrating Horizontal Well - Eclipse vs. Network Simulator With Superposition in Space used for Outer Reservoir

The inflow profiles are in good agreement and show that the superposition procedure can be applied to predict pseudo-steady state outer reservoir response for a non-uniform inflow and pressure profile along the near wellbore zone boundary. Thus, by applying the matched near wellbore zone radii found from the homogeneous case, various cases of different wellbore and near wellbore configurations may now be run with the network model.

5.5 Finite Conductivity Wellbore

As an illustration of the methodology's capability, a case with a 1.0in internal diameter completion was run with the network simulator and compared to the no friction case. The small internal diameter was selected to emphasize the effects of pressure loss and is not necessarily recommended practice for a field application. Figure 5.14 compares both the pressures along the wellbore and along the near wellbore zone for the two cases.

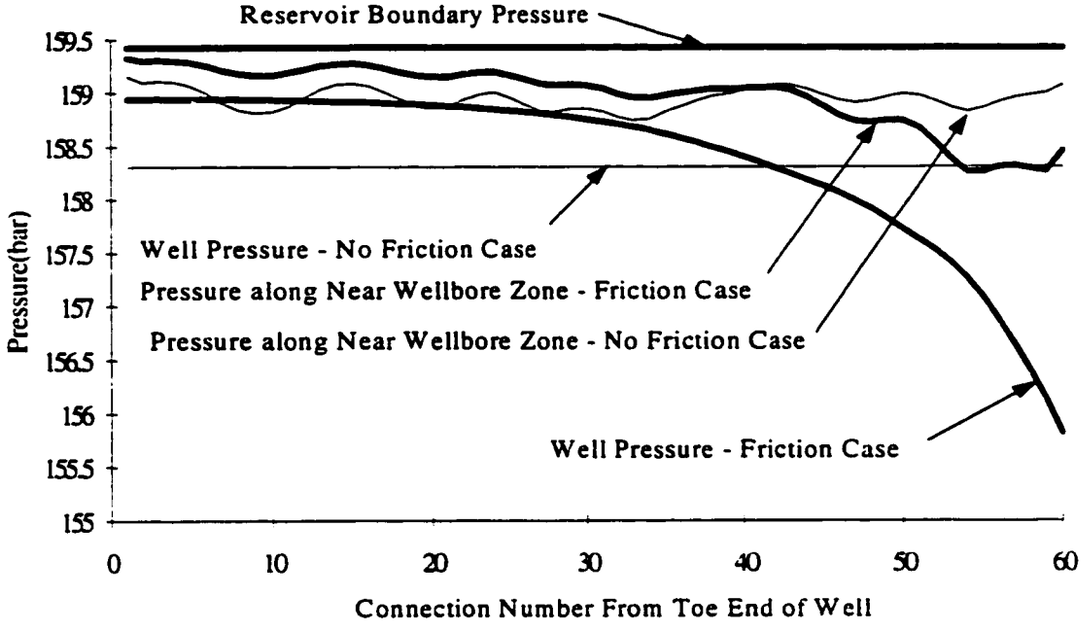

Fig. 5.14: Pressure Profile Comparison - Partially Penetrating Horizontal Well - Friction Less Wellbore vs. 1.0" ID Wellbore

For the friction case, an exponential drop in wellbore pressure occurs towards the down stream end of the well. The pressure along the near wellbore zone boundary is irregular due to the heterogeneous near wellbore zone, and declines towards the heel part

of the well. On the average, approximately 2/3 of the total pressure loss through the reservoir occurrs across the near wellbore zone. Thus, the productivity at a certain location is strongly affected by the first 5m reservoir. It can also be observed that the pressure loss along the well is much larger than the average drawdown along the well. Thus, a very unevenly distributed drawdown exists along the wellbore.

In addition to the non-uniform flowrate along the well, the acceleration of the incoming fluids represents the main difference between regular pipeflow and flow in the wellbore. Thus, it was of interest to quantify this contribution. The acceleration contribution to the pressure gradient in the wellbore is plotted versus location in Figure 5.15.

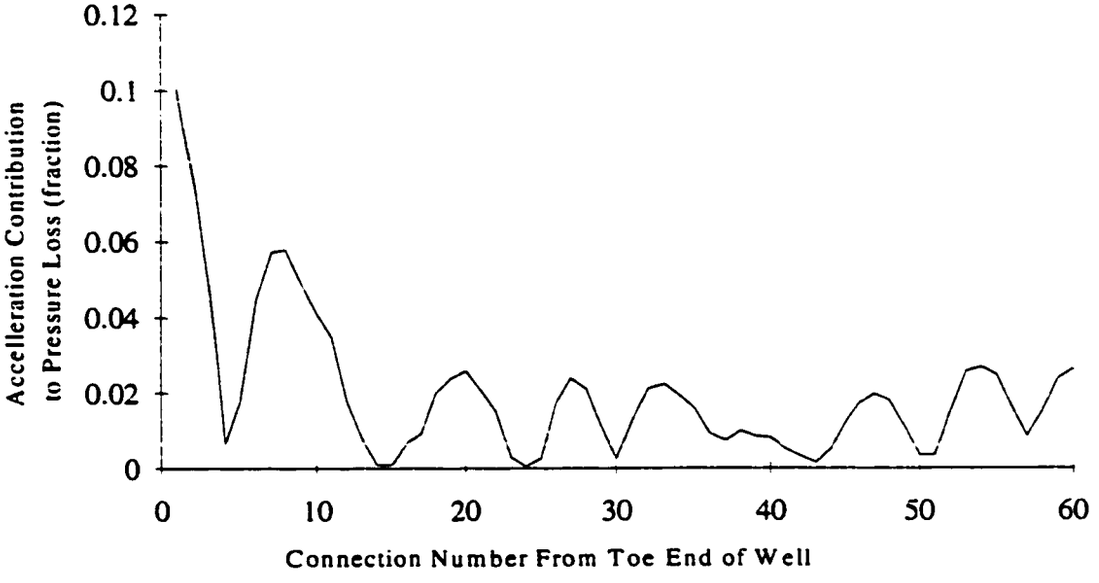

Fig. 5.15: Accelerational Losses Contribution to Wellbore Pressure Gradient vs. Location Along the Well

For most parts of the well, the contribution from acceleration losses is small, and is proportional to the amount of inflow relative to the axial flowrate. The highest percentage

contribution from acceleration losses (10 %) is seen at the upstream (connection #1) part of the well, where the frictional pressure loss is low. However, the absolute magnitude of the acceleration pressure loss increases towards the downstream (connection #60) part of the well where the total pressure gradient is large.

The influence of pressure loss along the well on the inflow profile is illustrated in Figure 5.16. The uneven distribution of drawdown for the friction case seen in Figure 5.14 had a considerable effect on the inflow profile along the wellbore.

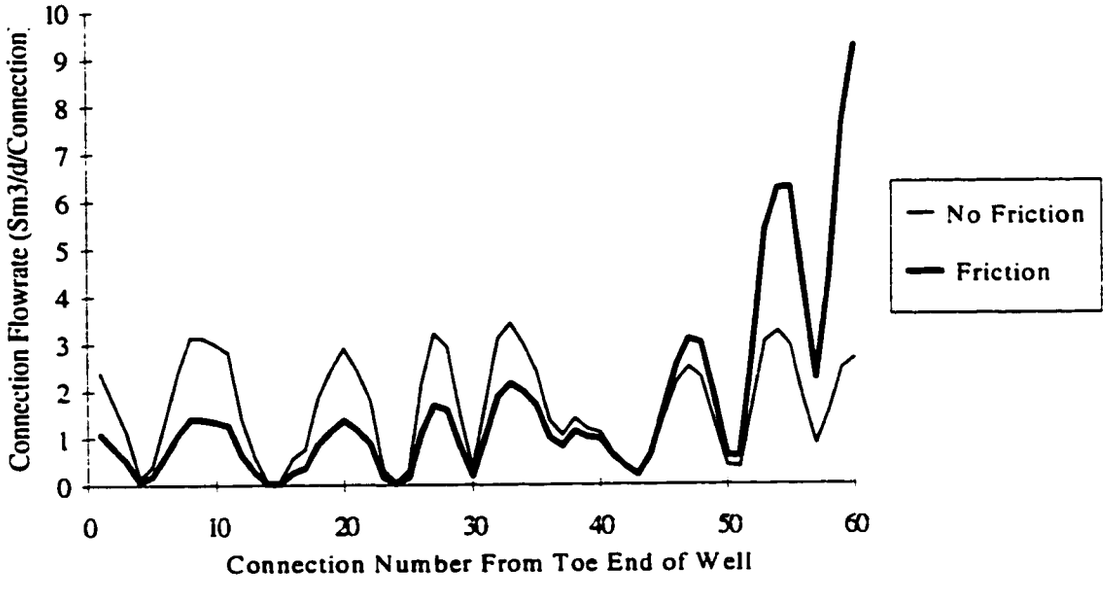

Fig. 5.16: Friction vs. no Friction Inflow Profile Comparison - Partially Penetrating Horizontal Well

Starting from the toe and moving towards the heel, the inflow per connection is reduced significantly throughout the first 70 % of the well. However, the lost production is compensated for by additional production along the last 30 % close to the heel end of

the well. For the friction case, the drawdown at the heel of the well is increased by 350 % when compared to the no-friction case. Thus, as the two cases had the same total flowrate, a 70 % reduction of total well productivity was experienced due to friction in the wellbore.

5.6 Upscaling Used for Multiphase Flow in the Near Wellbore Reservoir Zone

Multiphase flow occurring around a horizontal well can be due to saturated gas being released around the wellbore, gas and water cresting into the well from a gas cap/aquifer, or gas/water breaking through from an injection well. For saturated gas being released in the wellbore parts of the reservoir and carried with the fluids towards the wellbore, a homogeneous mixture of the phases is expected. For the coning situation, however, we will generally experience a high level of fluids segregation due to the hydrostatic forces involved and differences in mobility of the fluids.

The theory presented in Chapter IV for upscaling reservoir properties in the near wellbore zone was based on single phase flow conditions. However, it was of interest to apply the upscaling procedure to water and gas cresting cases with multiphase flow occurring in the near wellbore zone. We wanted to detect the magnitude of the error caused by the single phase flow assumption in the upscaling procedure. This exercise would indicate which physical flow phenomena were not represented correctly by the single phase upscaling. This information is critical in case of a possible extension of the upscaling procedure to also cover multiphase flow in the near wellbore zone.

A gas cap and an aquifer were configured with the gas-oil contact located 5.5 m above and the water-oil contact located 5.5 m below the well. Thus, both the gas oil contact and the water oil contact were located at the interface between the near wellbore zone and the outer reservoir.

A constant total (oil, gas and water combined) volumetric rate of 1000 m^3/d at reservoir conditions was produced throughout the simulations. The simulations were run for 111 days to allow sufficient time for gas and water to crest into the well. The same geometric layout of the reservoir and well was used for the multiphase verification runs, and the same heterogeneous near wellbore zone permeability field, as for the single phase cases, was used. The relative permeability curves and fluids' classification data were realistic data representative of a North Sea sandstone reservoir. Figure 5.17 shows the oil, gas and water relative permeabilities as functions of phase saturations.

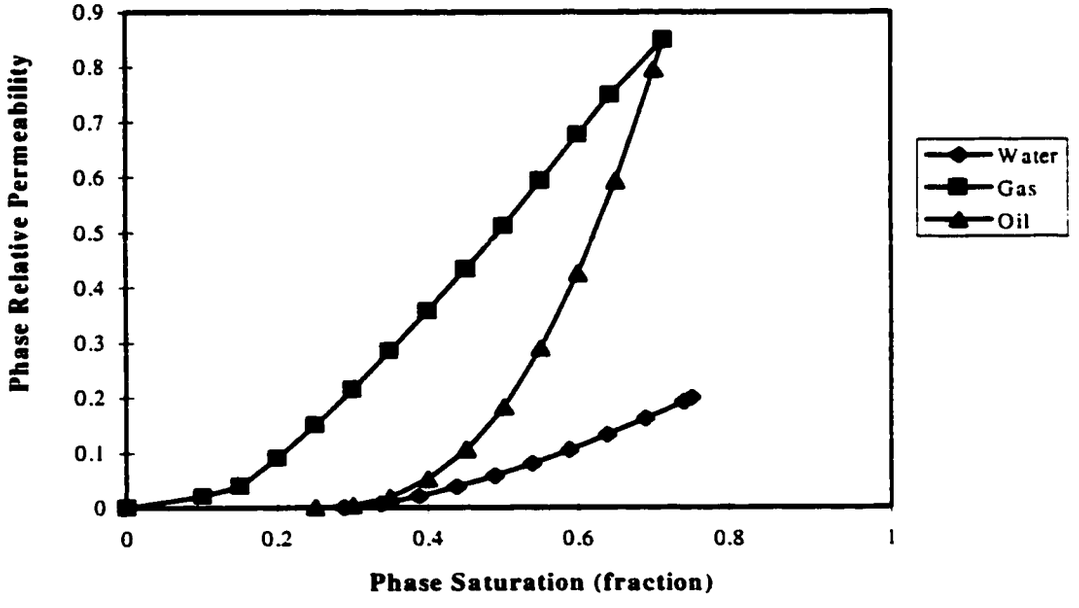

Fig. 5.17: Phase Relative Permeabilities vs. Phase Saturation.

Figures 5.18 and 5.19 illustrates the oil production rate vs. time, and the wellbore and boundary pressures vs. time for the fine grid heterogeneous case. In the early production phase, the oil production rate drops off rapidly mostly due to an immediate cresting of gas along the high permeability zones of the well. The oil rate approaches a more stable value as both the gas and water crests advance to the zones with lower permeability, and a more mature gas crest is formed. The low permeability zones above and below the well eventually becomes "gassed and watered out".

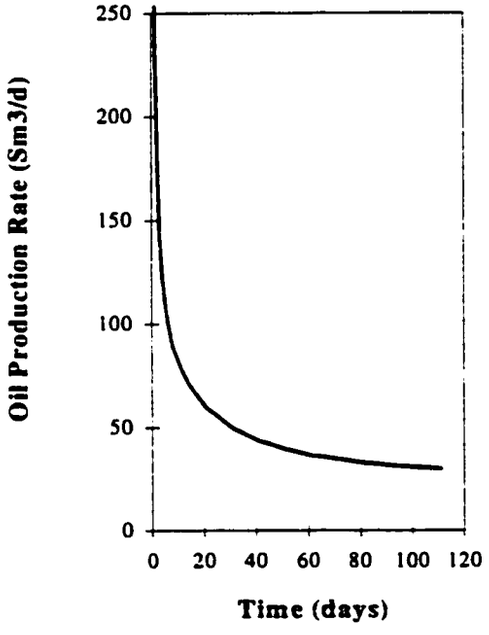

Fig. 5.18: Oil Production Rate vs. Time Heterogeneous, Multiphase Case

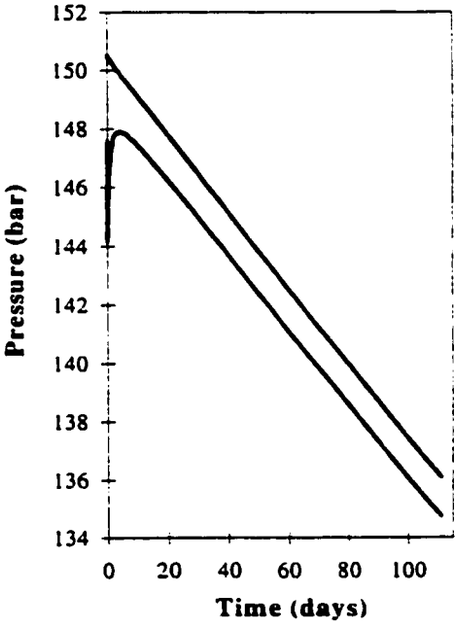

Fig. 5.19: Wellbore Pressure and Reservoir Boundary Pressure vs. Time Heterogeneous, Multiphase Case

The shapes of the gas and water crests during the late production times are determined to a large degree by the hydrostatic forces. Thus, the rapid change in phase saturations for

the near wellbore zone vanish. The pressure variations vs. time indicate a large and rapidly changing drawdown at early times and a more stable drawdown as the saturation in the near wellbore zone approaches a more stable value.

Figures 5.20 and 5.21 give watercut and GOR vs. time, respectively, for the fine grid, heterogeneous case. A delay in water cresting can be seen from the development in water cut in Figure 5.20. No water enters the well until 10 days into production.

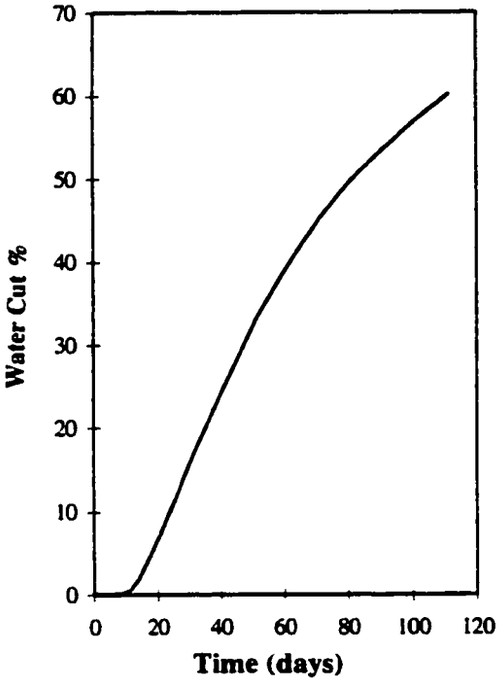

Fig. 5.20: Water Cut vs. Time
Heterogeneous, Multiphase Case

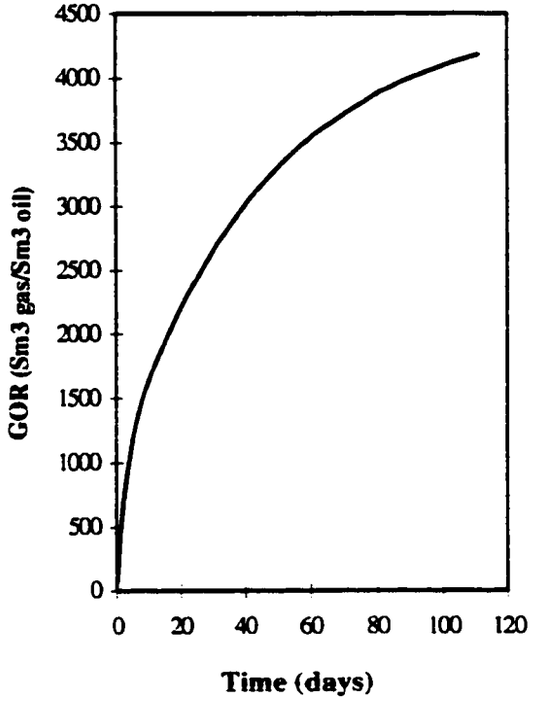

Fig. 5.21: GOR vs. Time
Heterogeneous, Multiphase Case

Thus, it takes 10 days for the water to cone into the well through the zones with the highest permeability. The water cut changes then to approximately 60 % throughout the 111 days production period. This indicates a significant change in water saturation in the

near wellbore zone throughout the entire period. The upscaled cases with large (11 m by 11 m) well blocks in the x and z directions will experience water and gas production in the well blocks immediately. However, the fine grid, heterogeneous case will more correctly experience a delay in both gas and water breakthrough as both gas and water have to flow 5 m before reaching the fine grid well blocks. Thus, this grid effect was expected to cause an error in the early production period for the upscaled cases.

Also, as the entire circumference of the near wellbore zone had been upscaled from 121 block per 5 m well section to one block per well section, no resolution exist in the θ direction for the upscaled case. The reservoir simulator will predict pressure loss vs. flowrate by assuming a homogeneous mixture of oil, gas and water within the last 5.5 m from the wellbore (within the coarse grid well block). However, the fluids are not expected to flow in a homogeneous mixture for the given coning case, but rather a high level of fluids segregation is expected from fingering and hydrostatic effects. Thus, for the same saturations, a homogeneous mixture of oil, gas and water causes a larger pressure loss than segregated oil, water and gas. A too large pressure loss through the near wellbore zone is then calculated by using the coarse, upscaled grid blocks.

We attempted to correct this error by applying straight line relative permeability curves in the upscaled near wellbore region. Straight line relative permeability relationships are representative of the pressure loss we would obtain if the phases were flowing through the block segregated from each other. Thus, we tried to correct some of the error by forcing the simulator to behave as if the flow within the wellbock was fully segregated.

Two upscaled cases were run for comparison with the original, fine grid case. The first case had the original relative permeabilities for the entire reservoir. The second case had

the original relative permeabilities replaced with straight line relative permeability curves in the near wellbore zone. Figure 5.22 illustrates the percentage error in drawdown for these two upscaled cases when compared to the "real" fine grid case.

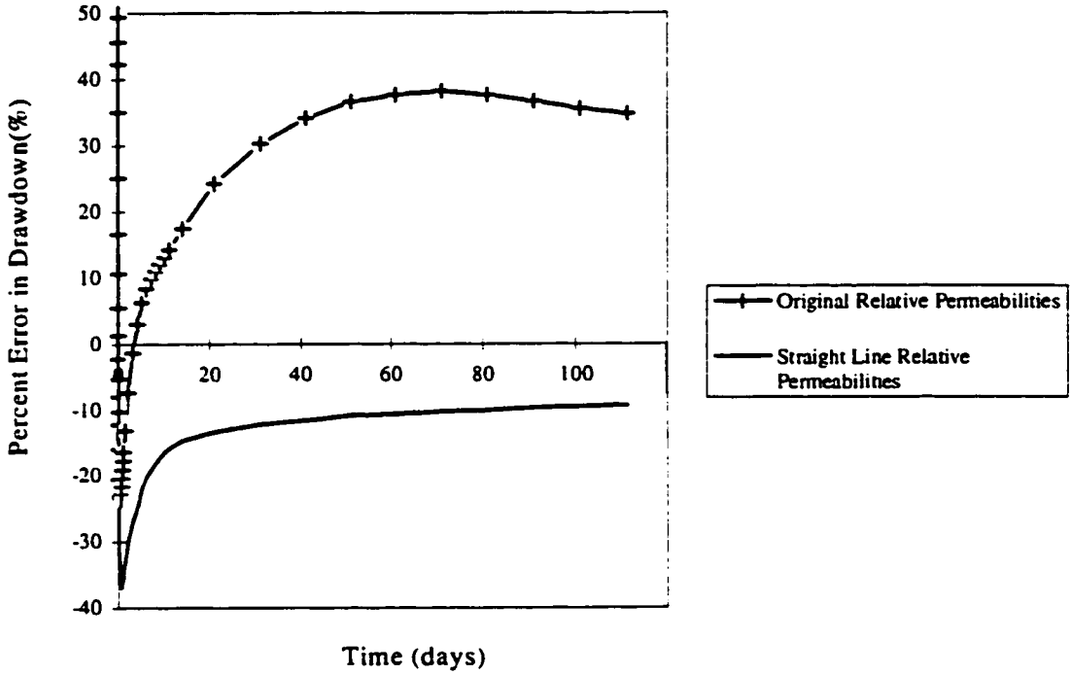

Fig. 5.22: Error in Drawdown vs. Time - Multiphase Case

Both upscaled cases start off with a significantly too small drawdown. The drawdown for the case with original relative permeabilities increased rapidly, and peaked at an approximately 40 % too large drawdown after 70 days. Thus, the pressure loss is overpredicted when using the original relative permeability curves in the near wellbore region. If we introduce straight-line relative permeability relationships in the near wellbore zone, we obtain the extreme limiting case of fully segregated flow. In this case, the drawdown is under predicted by about 15 % as shown in Figure 5.22. As expected, the

real drawdown is located between these two extreme cases. However, a 75 % smaller error was obtained by assuming fully segregated flow. An inspection of the phase saturations in the near wellbore zone confirmed a high level of fluids segregation which increased throughout the simulation.

In Figures 5.23 through 5.25, the error in water, oil and gas production rates are illustrated vs. time, respectively, for the two upscaled cases. A very large error in water production rate is seen from Figure 5.23 for both upscaled cases during the first 30 days of production.

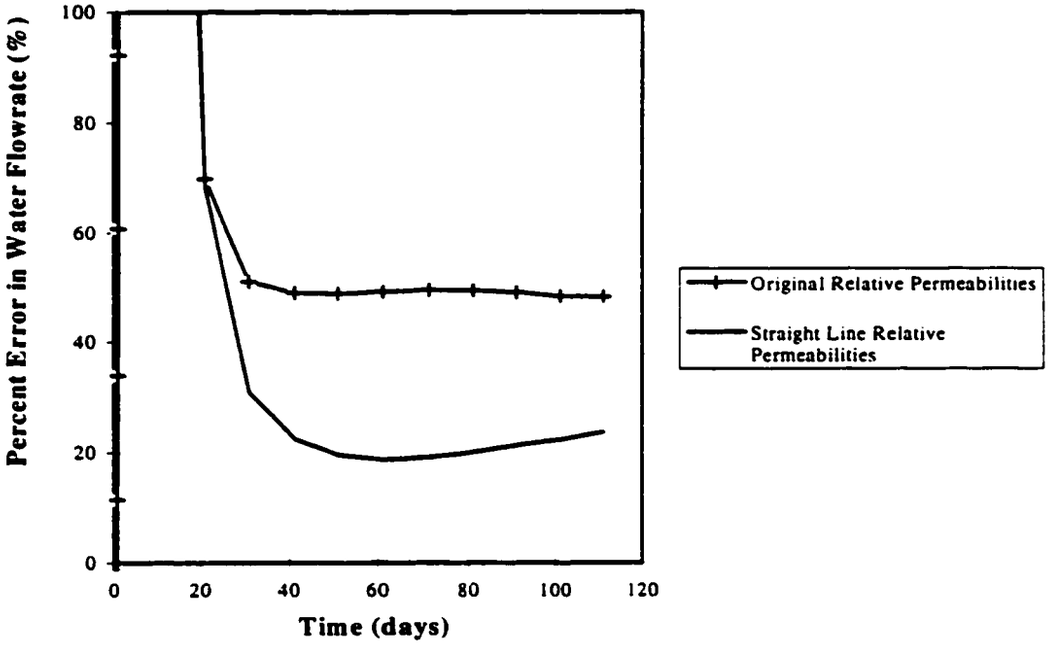

Fig. 5.23: Error in Water Production Rate vs. Time - Multiphase Case

This is due to the above explained grid effect. Water contacts the wellblocks and is produced immediately for the upscaled cases, but a delay in water breakthrough is correctly experienced for the "real" case. No cure can be suggested for this error.

However, if the oil-water and gas-oil contact had been located further away from the near wellbore zone, this effect would not have been as severe as for the given case. The error in water production rates stabilizes after 30 days at 50 % for the case with original relative permeabilities, and at 20 % for the case with straight line relative permeability curves in the near wellbore reservoir zone. Thus, although still in error, a significant improvement is experienced also with respect to the water production rates by assuming fully segregated flow.

Figure 5.24 gives the error versus time for the oil flowrate. Both upscaled cases have a too small oil production rate early in the production period. This error is mainly due to the large early error in water production rates and can, thus, be attributed to grid effects. The case with the original relative permeabilities stabilizes, however, at a 30 % too small oil production rate, whereas the case with straight line relative permeabilities again performs better with only a 10 % too large stabilized oil production rate. Thus, a 66 % improvement in error is seen by assuming fully segregated flow.

Finally, Figure 5.25 gives the error in gas production rate vs. time for the two upscaled cases. Again, a large, positive error in gas production can be seen for both cases at very early times. This is again attributed to grid effects and different breakthrough times for the original and upscaled cases. However, only a 1 % error in gas production rates is seen throughout most of the remaining time for the original relative permeability case, and virtually no error is experienced for the case with straight line relative permeabilities.

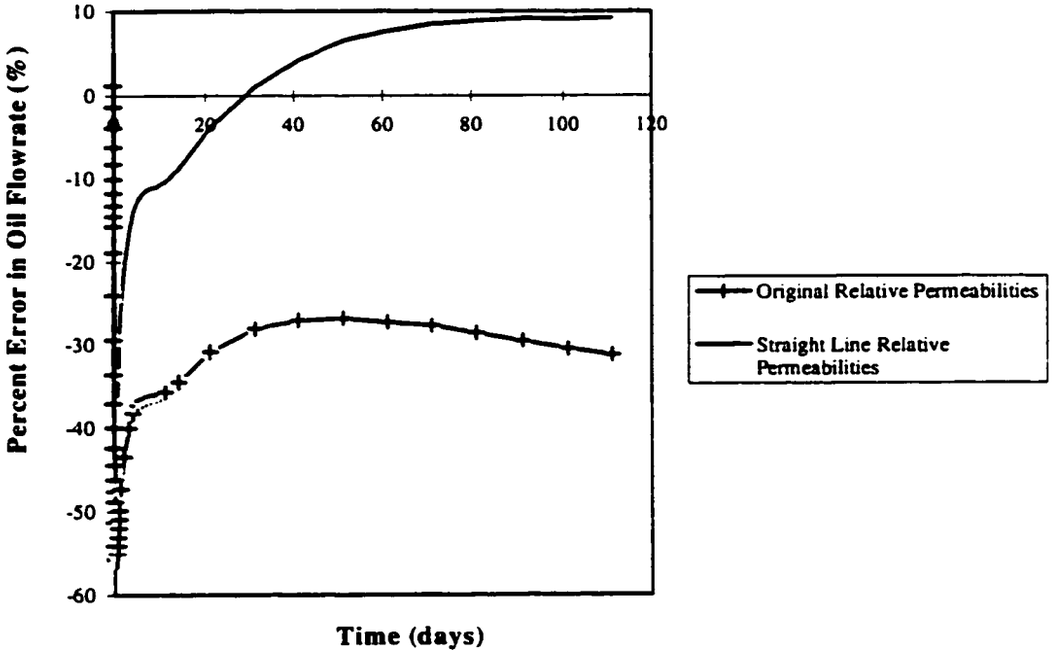

Fig. 5.24: Error in Oil Production Rate vs. Time - Multiphase Case

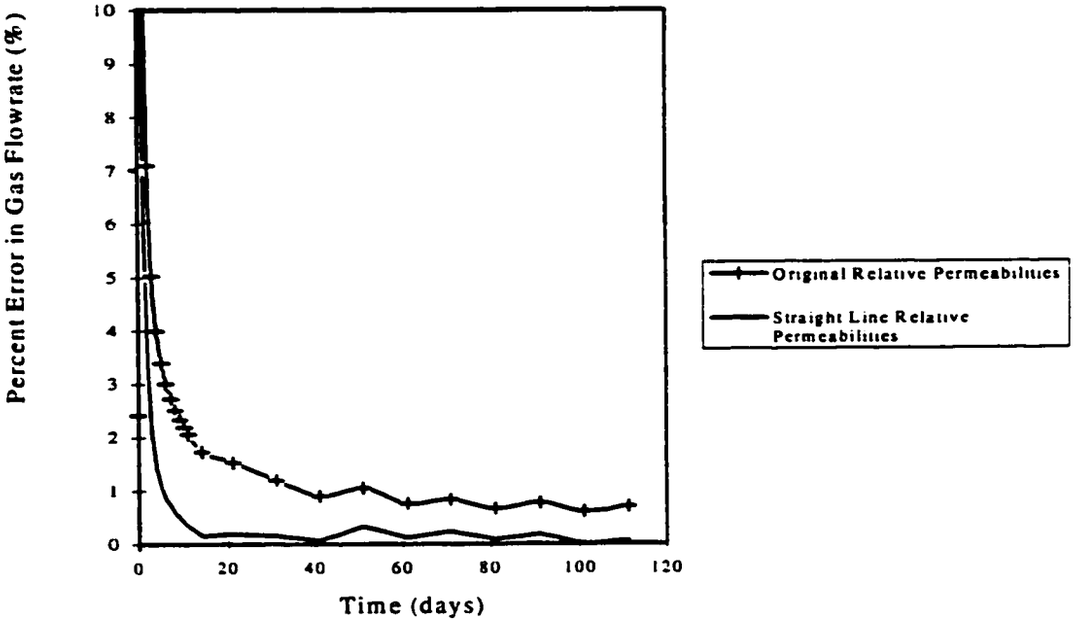

Fig. 5.25: Error in Gas Production Rate vs. Time - Multiphase Case

Several complicating multiphase aspects have been illustrated concerning upscaling of permeabilities in the near wellbore region:

a) For coning or flooding cases, the delay of water and gas breakthrough is decreased by increasing the size of the well blocks. This large error is a grid effect that can not be easily accounted for.

b) The error from the lack of ability to correctly represent segregation of fluids in a coarse grid, upscaled wellblock is significant. This effect for water and gas coning cases can, however, be reduced by correcting the relative permeability curves in order to emulate fully segregated flow in the well blocks.

c) Accumulation of gas due to a critical gas saturation for flow delays the steady-state flow behavior in the near wellbore zone necessary to honor the steady-state flow assumption in the upscaling procedure.

CHAPTER VI

PRODUCTIVITY AND RISK ASSESSMENT FOR HORIZONTAL WELLS

EXAMPLE APPLICATION

The potential advantage of the developed approach is illustrated by applying the network model in a statistical study. Probability distributions were generated for total well productivity of a horizontal well. The main goal was to calculate the uncertainty in the total well productivity caused by uncertainties in near wellbore geology. The objective was also to detect how parameters such as well length and friction through the completion would influence the probability distribution for well productivity.

Unconditional Sequential Gaussian Simulation was used to generate 300 isotropic, heterogeneous permeability fields for the reservoir zone close to the wellbore. The near wellbore realizations satisfied a log-normal permeability distribution. The geometric layout of the reservoir was kept similar to the above described verification model for a partially penetrating horizontal well. However, the horizontal and vertical correlation lengths (variogram range) were set to be 150 m and 1.5 m, respectively, which would give a base case for more field like conditions.

Each of the 300 near wellbore realizations were upscaled from 7260 permeabilities to 60 apparent permeabilities for use in the network simulator. The outer reservoir response was implemented by applying the semi-analytical method in the network simulator as for the above explained verification runs.

The network model was run for all the 300 sampled near wellbore realizations. Each simulation series (300 runs) was carried out for well lengths varied in steps between 50 m and 300 m with and without friction in the wellbore.

6.1 Infinite Conductivity Cases

A distribution consisting of 300 total well productivities was obtained for each well length. Figure 6.1 gives the average well productivity, the most likely (mode) well productivity if only one well is being considered, and the boundaries where 20 % and 80 % of the productivities fall below the indicated value. All productivities are given as productivity per unit well length.

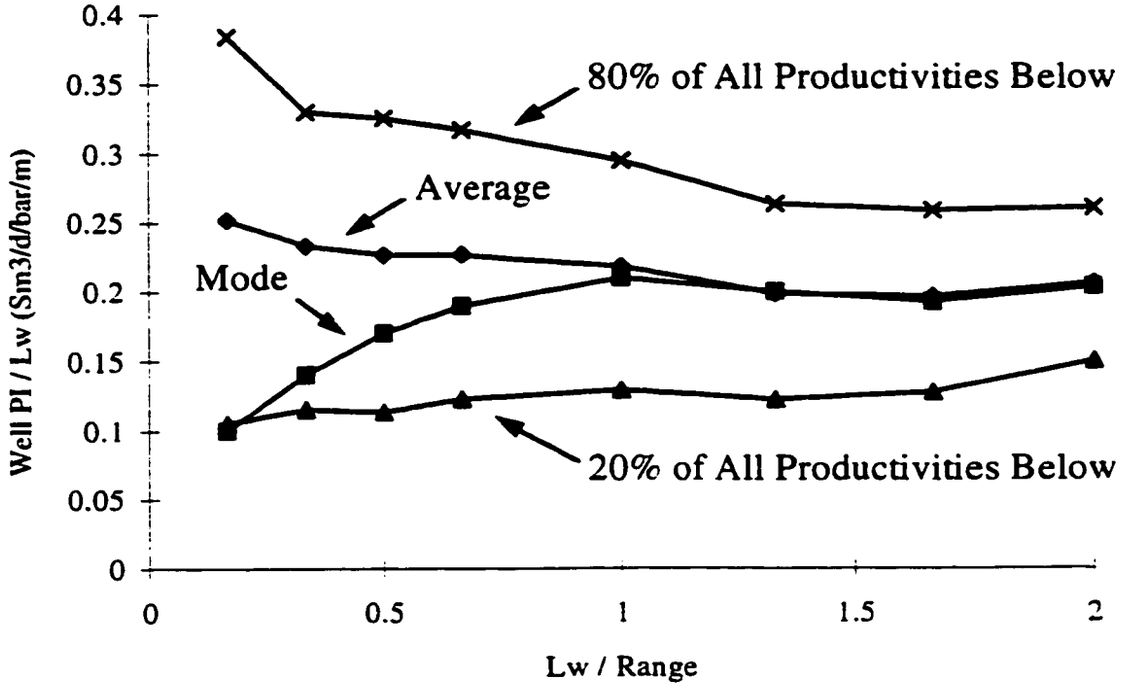

Fig. 6.1: Infinite Conductivity Wellbore - Average, Most Likely, 20 % Limit and 80 % Limit of Total Productivity / Well Length vs. Well length / Range

For very small well lengths, the average productivity per unit well length attain a high of 0.25 $Sm^3/d/bar/m$ due to an efficient spherical flow geometry in the reservoir (point drainage). However, if only one well is considered, the most likely productivity would be the mode. For a well length of 25 m, the most likely outcome of a one well operation would be 0.1 $Sm^3/d/bar/m$ which is among the 20 % lowest productivity cases. This is a 60 % reduction of productivity as compared to the average. Thus, for well lengths shorter than the range of the variogram, the mode of the distribution is considerably smaller than the average. This indicates that the distribution is skewed to the right (log-normal like shape), as seen in Figure 6.2, where the probability distribution for a 25 m long well is given.

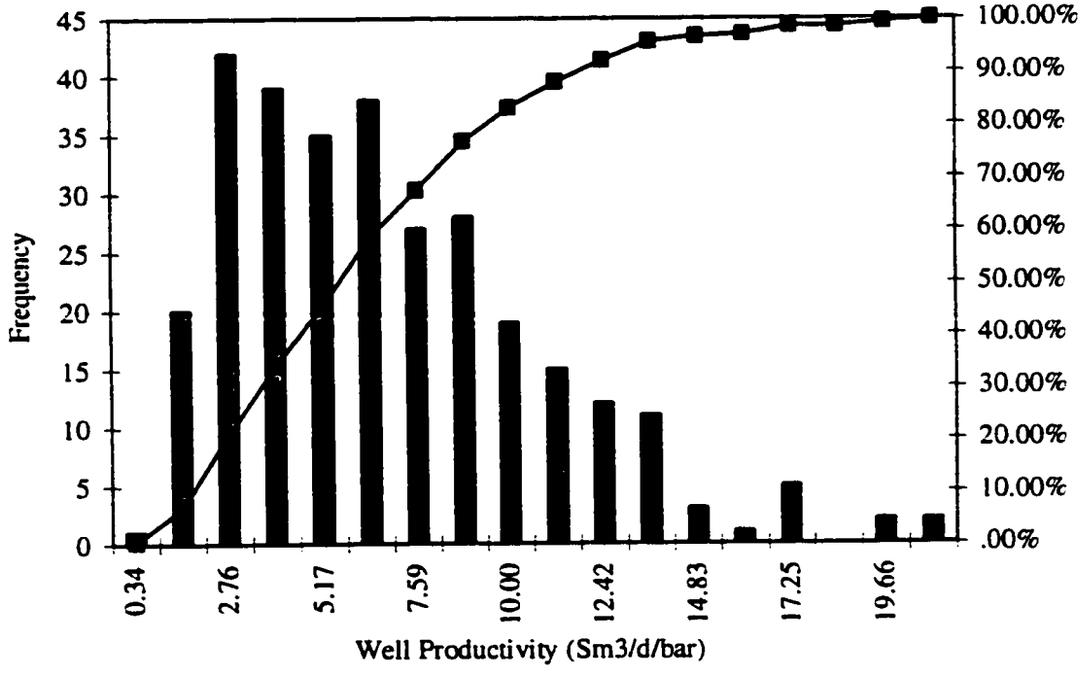

Fig. 6.2: Probability Distribution for Well Productivity at 111 days into Production, Well Length = 25 m, Infinite Conductivity Wellbore

As pointed out in the literature, a log-normal distribution of productivities may be expected due to log-normally distributed permeabilities[2]. Since productivity is strongly dependent on the near wellbore zone permeability, the productivity distribution will for short wells resemble the permeability distribution.

However, for well lengths exceeding the horizontal variogram range, a normal distribution is expected. For a long well with infinite conductivity wellbore, we assume that the well productivity is approximately a linear and proportional function of the average of the near wellbore zone permeabilities, and that a sufficiently long well contacts an adequately large sample of the permeability distribution. The relationship between well productivity and the average near wellbore zone permeability depends on the relative magnitude of the pressure loss through the near well bore zone when compared to the total reservoir pressure loss. If this relative magnitude is large, we are indirectly sampling a proportional function of the average of the permeability field.

If permeability values are independent, then the Central Limit Theorem[17] states that the average values of large numbers of permeabilities is approximately normally distributed. If productivity is proportional to average well block permeability, then the productivities for long wells would also be approximately normally distributed. Since permeability values in this case are correlated within a range of 150 m, the Central Limit Theorem does not apply directly. However, one would still assume that the averages of a large number of permeabilities would be approximately normally distributed if the well length is much longer than the variogram range.

It can be noted from Figure 6.1 that the average and the mode of well productivity take approximately the same value for well lengths greater than the horizontal variogram range.

This indicates a normal distribution. The same argument can be made by inspecting the shape of the productivity distribution for a well length of 300 m, as given in Figure 6.3.

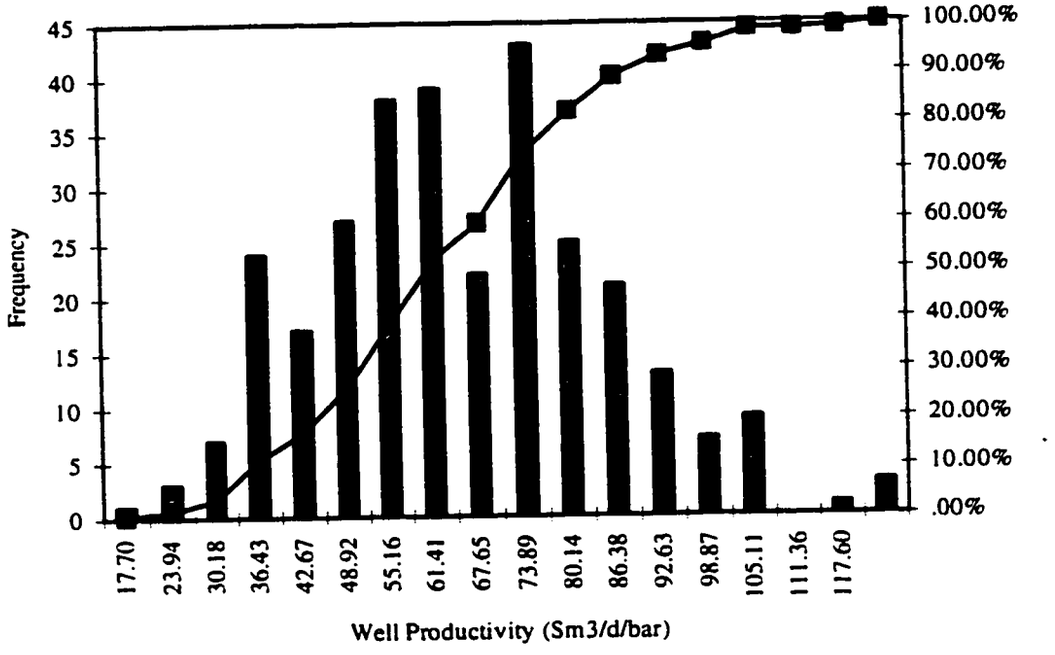

Fig. 6.3: Probability Distribution for Well Productivity at 111 days into Production, Well Length = 300 m, Infinite Conductivity Wellbore

Thus, if a well effectively contacts the formation through at least one correlation length (range of the variogram), one may expect the well productivities to be normally distributed. However, since the productivity is in a large degree dependent on the perforated well blocks, rather than well blocks in general, we assume that the entire length of the wellbore is open for inflow. This may not be the case due to formation damage, poor cleanup or too sparsely, randomly perforated cemented liners. If the location of perforations are based on some prior knowledge (i.e. logs), the given approach is not valid.

By inspecting the 20% and 80% limits for productivity in Figure 6.1, it can be observed that the uncertainty in well productivity per unit well length is decreasing with increasing well length. The Central Limit Theorem also states that the variance of the sampled averages will be approximately proportional to the inverse of the sample size (number of involved well blocks).

Thus, assuming that the well is sufficiently long and that productivity per unit well length is proportional to average well block permeability, the Central Limit Theorem suggests that the variance of productivity per unit well length for undamaged wells (with no formation damage and no friction in the wellbore) is proportional to the inverse of the well length. To support this theory, the variance of productivity per unit well length multiplied by the well length is plotted versus well length in Figure 6.4.

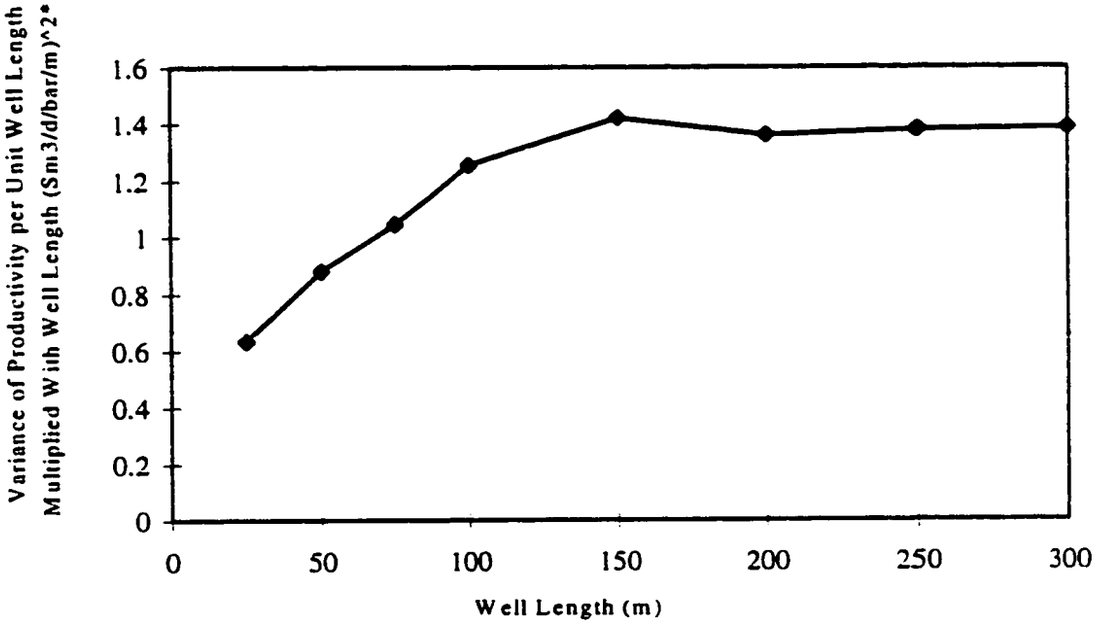

Fig. 6.4: Infinite Conductivity Wellbore - Variance of Productivity per Unit Well Length times Well Length vs. Well Length

It can be noted that the variance of productivity per unit well length multiplied with the well length becomes approximately constant for well lengths greater than the horizontal correlation length (variogram range). This supports the theory that for well lengths greater than the variogram correlation length, the variance of the probability distribution for well productivity per unit well length declines proportionally to the well length.

6.2 Finite Conductivity Cases

It has been clearly demonstrated in the literature that the well productivity for a horizontal well can be severely reduced from pressure losses occurring in the wellbore[1]. However, the effect of wellbore pressure loss on the shape of the productivity distribution has not yet been studied in detail. Thus, we know that the average productivity will be reduced, and we have been able to calculate this effect by applying average values as input to our calculations. However, here we attempt to study the effect of friction in the wellbore on the most expected productivity (mode of the distribution) and on the variance of the probability distribution.

For the finite conductivity wellbore cases, the same geometric layout of the well and the reservoir was selected. The completion was configured with an internal diameter of 1.5 in. The roughness of the completion may depend for a real case on several factors such the quality of completion manufacturing, installation and clean-up. Thus, to include this as an uncertainty, the absolute roughness was sampled randomly from a uniform distribution between 0.1 mm and 20 mm for each liner segment. Figure 6.5 gives for finite conductivity cases the same information as was shown in Figure 6.1 for the infinite conductivity cases.

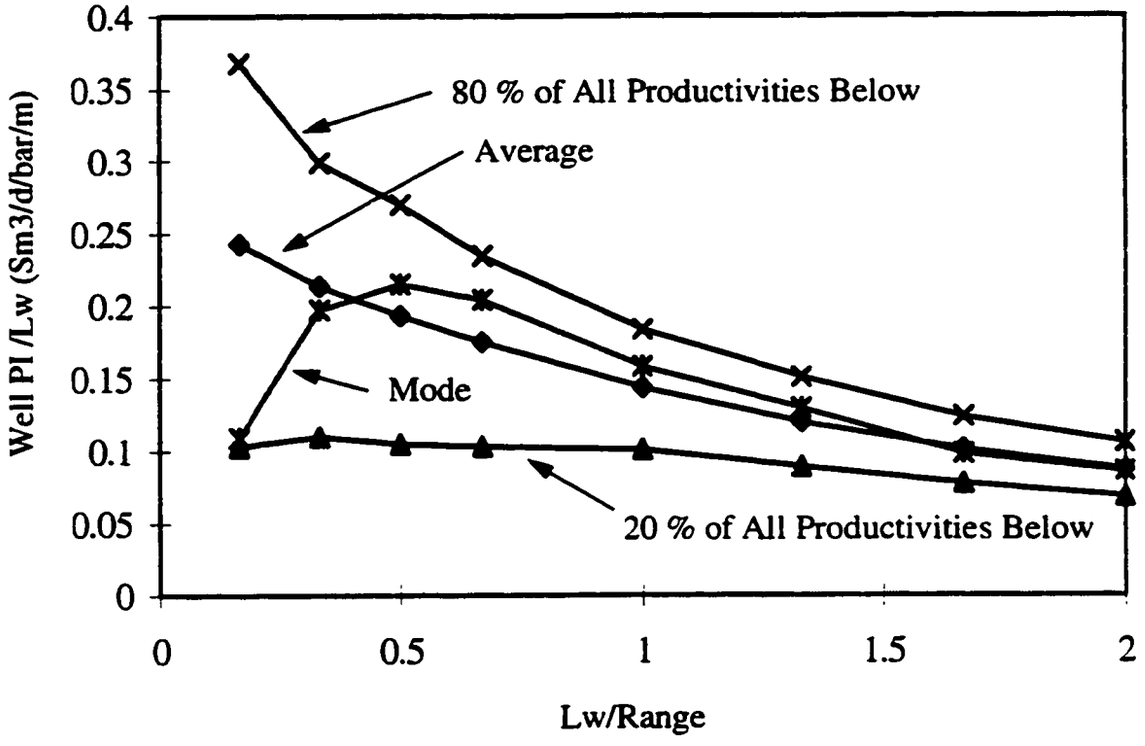

Fig. 6.5: Finite Conductivity Wellbore, Average, Most Likely, 20 % Limit and 80 % Limit of Total Productivity / Well Length vs. Well length / Range.

The average well productivity, the most likely (mode) well productivity and the 20 % and 80 % limits are given versus well length divided by correlation length.

For very short well lengths, where the friction loss is small, all values are comparable to the no-friction cases. However, as well length increases, the mean and the 80 % limit decline rapidly due to the effects from more pressure loss in the wellbore. The 20 % limit is not as strongly affected which indicates that pressure loss in the wellbore affects low productivity cases less than high productivity cases. For a more clear illustration of the wellbore pressure loss effect, the average well productivity versus well length divided by correlation length for friction and no friction cases are compared in Figure 6.6.

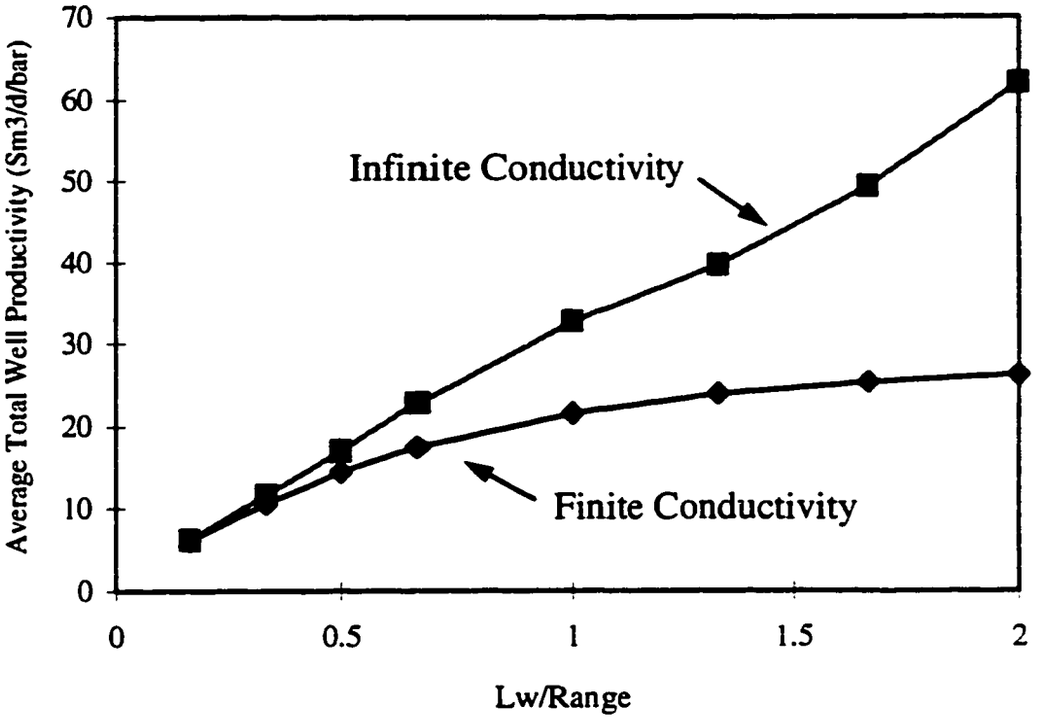

Fig. 6.6: Comparison of Average Total Well Productivity - Infinite and Finite Conductivity Wellbore

It is apparent that extending the well length beyond 300 m has little or no effect on the total productivity when friction is included in the calculations. However, for the no friction case, a proportional increase in productivity is always seen for increasing well lengths.

It should also be noticed that the uncertainty in productivity is reduced considerably by the introduction of friction in the well. Thus, in addition to shifting the probability distribution, friction also has an effect on the shape of the distribution. For well lengths up to a certain limit (in this case ≈ 200 m), friction has a considerably stronger damaging effect on the productivity for high productivity than low productivity cases. Thus, friction skews the distribution towards smaller productivities. This can be seen from the behavior of the mode, which increases rapidly, and for medium well lengths is greater than the

average productivity. In Figure 6.7 the finite conductivity productivity as a fraction of infinite conductivity productivity is plotted versus infinite conductivity productivity for all realizations of the near wellbore permeability field. All cases are for a well length of 50 m.

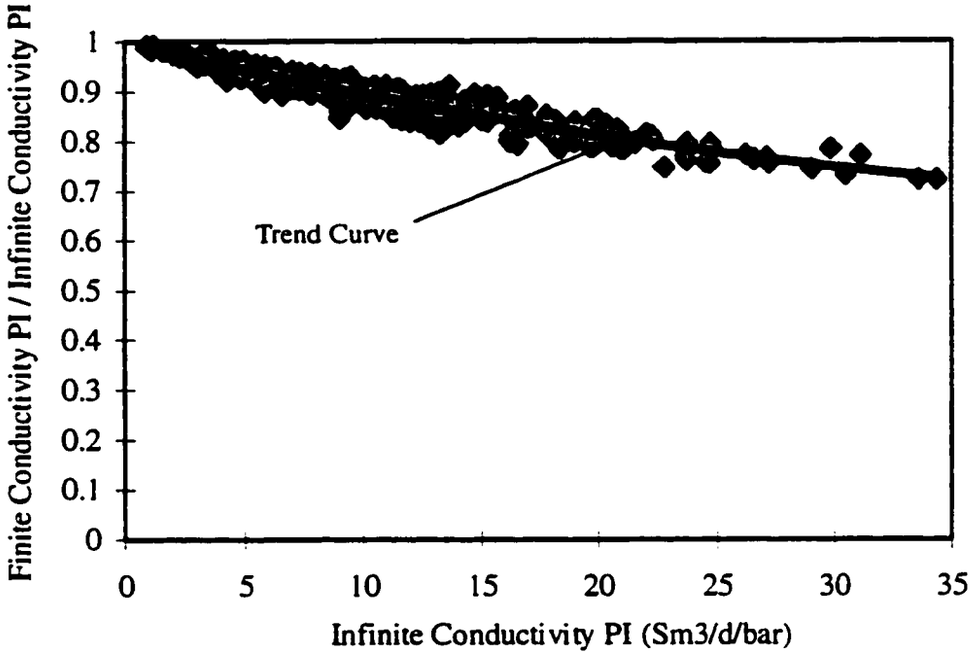

Fig. 6.7: Finite Conductivity PI as Fraction of Infinite Conductivity PI vs. Infinite Conductivity PI - Well Length = 50 m

For this well length, the damaging effect from friction is a strong and almost linear function of well productivity. However, if the well length is further increased, the damaging effect from friction tends to discriminate less among the cases. Thus, a more constant shift of the distribution is seen. For 300 m long wells, the reduction in productivity due to friction is given vs. productivity in Figure 6.8. A more constant reduction of productivity can be seen for a wide range of high productivities.

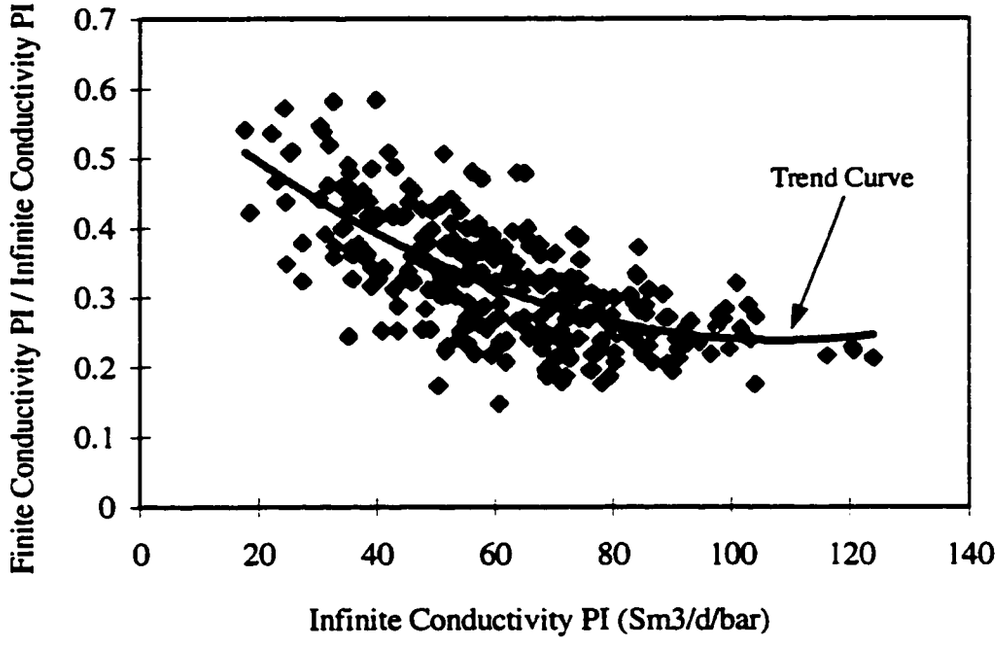

Fig. 6.8: Finite Conductivity PI as Fraction of Infinite Conductivity PI vs. Infinite Conductivity PI - Well Length = 300 m

This is also supported by Figure 6.5, where it can be noted that the mode and the average approach the same value for long wells.

At first glance, this might seem unreasonable since the pressure loss, as a rule, has a more unfavorable effect on the productivity of high productivity wells than low productivity wells. However, by further inspection, the effect seen is reasonable since the magnitude of the frictional losses depends on the average flow length for the fluid though the completion. For low productivity cases with a fairly uniform influx, this average length approaches 50 % of the total well length, and the friction pressure loss is large. High productivity cases have most of the fluids entering into the wellbore close to the heel of

the well which results in a considerably smaller average flow length and smaller frictional pressure loss. Thus, there appears to be a limit to the productivity reduction due to friction pressure loss in the wellbore. This is an increasingly dominant effect as well length is increased.

6.3 Efficiency of Network Approach

The CPU time combined with the upscaling procedure was 10 hours on a Pentium PC, but may be reduced considerably by focusing on the no-cross flow upscaling procedure alone and by taking advantage of geometric repetitiveness along the well and between cases where only block permeabilities are changed (different realizations). A total number of 4800 simulations were automatically performed with the network simulator for this study. These simulations took 1/2 hour to run on a Pentium PC, while fine grid simulations using Eclipse on a HP 700 would have taken on the order of 400 hours to run. Thus, a reduction in computer time on the order of 1/40 was obtained.

However, by taking advantage of similar grid and well layout between the realizations, the CPU time for upscaling could be reduced by a factor of 1/300. Thus, the total CPU time for the network approach could be 32 minutes instead of 10.5 hours. Also, by taking advantage of the repetition of grid and well geometry along the well there is a further potential of reducing CPU time for upscaling in this case by a factor of 1/60.

CHAPTER VII

CONCLUSIONS

This chapter summarizes the major contributions and conclusions of this dissertation.

1. A network model capable of solving nonlinear networks has been developed and successfully applied for comprehensive flow prediction of horizontal wells producing single-phase fluids. The simulator is capable of combining the effects resulting from non uniform wellbore pressure, the effects from heterogeneous and anisotropic permeabilities near the wellbore and the effects from the outer reservoir. For the simulated cases, the network model predicts well productivity within an accuracy of 1% when compared to fine grid reservoir simulations.
2. A method, based on superposition in space, has been developed and implemented in the network model to include pseudo steady-state reservoir response for an arbitrary pressure and inflow profile along the well. Thus, the long term well performance can be obtained without performing transient, numerical reservoir simulations.
3. An upscaling procedure for single-phase radial flow in heterogeneous, anisotropic near wellbore formation has been developed, tested and successfully applied to drastically reduce the grid resolution needed to accomplish accurate flow predictions for horizontal wells.
4. The following difficulties associated with upscaling for multiphase flow have been detected. The time until breakthrough for a coning case is decreased by increased size

of the upscaled well blocks. This error is a grid effect that can not be easily accounted for. The error made from calculation for homogeneous mixture flow in the large, upscaled wellblocks can, for water and gas coning cases, be decreased by correcting the relative permeabilities to reflect fully segregated flow in the near wellbore region. Accumulation of released associated gas in the near wellbore zone delays the time at which steady-state flow is obtained in the near wellbore zone. Thus, the steady state assumtion in the upscaling procedure is not valid for this transient production period.

5. Upscaling by applying the no cross flow assumption around the wellbore appears to be the superior method for the single phase cases investigated in this study.
6. A procedure for converting block permeabilities from a Cartesian to a cylindrical coordinate system was developed, implemented and applied to facilitate radial flow upscaling.
7. An existing model for accelerational pressure loss in horizontal wells producing single-phase fluids was verified through experiment and implemented in the network approach.
8. By using the proposed approach to perform an example statistical study of the total well productivity, the required CPU time was reduced by a factor of 40 (with potential for improvement) as compared to using a numerical reservoir simulator.
9. Statistical analysis of the well productivity due to variations in near wellbore geology indicated a log-normal (skewed to the right) distribution of productivities for well lengths shorter than the horizontal correlation length (range).
10. As the well length exceeds the horizontal correlation length, a normally distributed probability distribution is obtained for the horizontal well total productivity.

11. Friction in the wellbore generally skews the productivity distribution to the left.

12. For longer wells than the horizontal correlation length, the variance of productivity per unit well length is approximately proportional to the inverse of the well's length.

NOMENCLATURE

Symbol	
A	area (m^2)
e_D	relative roughness
f	friction factor (Moody)
h	length of reservoir section (m)
k	permeability (md)
l	length (m)
$\dot{m}$	mass flowrate (kg/s)
N_{Re}	Reynolds number
p	pressure (bar)
PI	productivity ($Sm^3/d/bar$)
q	flowrate (m^3/d)
r	radius
s	skin factor
v	velocity (m/s)
ρ	density (kg/m^3)
μ	viscosity (cp)
θ	angle from horizontal

Subscripts

app	apparent
e	relates to outer boundary
d	down-stream
k	iteration counter
tot	total
u	up-stream
w	wellbore
x	horizontal direction (perpendicular to the well)
y	horizontal direction (parallell with the well)
z	vertical direction

REFERENCES

1. Brekke K. and S. C. Lien: "New, Simple Completion Methods for Horizontal Wells Improve Production Performance in High Permeability Thin Oil Zones," *SPE Drilling and Completion*, September 1994, pp 205-209.
2. Beliveau D.: "Heterogeneity, Geostatistics, Horizontal Wells, and Black Jack Poker," *SPE Journal of Petroleum Technology*, December 1995, pp 1068-1074.
3. Eclipse 100, Release 94A, Reference Manual, Intera Information Technologies Limited, England
4. Deutsch, C. V. and Journel, A. G.: "GSLIB - Geostatistical Software Library and Users Guide," New York / Oxford, Oxford University Press, 1992.
5. Engler T. W., Osisanya S. and Tiab D.: "Measuring Skin While Drilling", paper SPE 29526 presented at the Production Operations Symposium, Oklahoma, 1995.
6. Dolan A. and Aldous J.: "Networks and Algorithms," John Wiley and Sons, Chichester (1981).
7. Asbjornsen O. A.: "Algoritms and Programs for the Solution of General Fluid Flow Networks," Internal Report, Norsk Hydro, 1990.
8. Brekke K., T. E. Johansen and R. Olufsen: "A New Modular Approach to Comprehensive Simulation of Horizontal Wells," paper SPE 26518 presented at the 68th. SPE Annual Technical Conference and Exhibition, Houston, 1994.

9. Brekke, K and L. G. Thompson.: "Productivity and Risk Assesment for Horizontal Wells," paper SPE 36578 presented at the 70th. SPE Annual Technical Conference and Exhibition, Denver, 1996.

10. Kasap, E., Class Notes: "Upscaling of Reservoir Heterogeneities," The University of Tulsa, OK (1995).

11. Kasap, E. *et al*: " Calculating the Effective Permeability Tensor of a Grid Block," *SPEFE*, June 1990.

12. Sagar, R.: "Reservoir Description by Integration of Well Test Data and Spatial Statistics," Ph. D. dissertation, University of Tulsa, OK (1993).

13. Aasum, Y *et al*: "Analytical Up-scaling of Small Scale Permeability Using a Full Tensor," *Petroleum Geoscience*, (Vol 1, no. 4, 1995), European Association of Petroleum Geoscientists.

14. Muskat, *Flow of Homogeneous Fluids Through Porous Media*, McGaw Hill, 1937.

15. Aasheim, H.: "A Flow Resistance Correlation for Completed Wellbore," *Journal of Petroleum Science and Engineering*, (vol. 8), 1992, pp 97-104.

16. Brekke K, A. Valle and Utvik, O. H.: "Two Phase Flow in Horizontal Wells," PROFIT, a project directed by "det Norske Petroleumsdirektorat" , 1994.

17. Robert V. Hogg and Johannes Ledolter, *Applied Statistics for Engineers and Scientists*, Macmillan Publishing Co., USA, 1992.

APPENDIX A

In this appendix, the theory is presented for the transformation of a heterogeneous permeability field from a cartesian coordinate system to a cylindrical coordinate system. As illustrated in Chapter V, the basis for this transformation is a voumetric averaging of the permeabilities for the cartesian block volumes enclosed in a cylindrical block.

To avoid an excessively complex, three dimensional transformation, the problem was reduced to two dimensions by making the assumption that the Cartesian grid block division in the well direction always corresponds to the required grid block division for the cylindrical grid. Thus, rather than calculating volumes for use in the cylindrical block permeablility averaging, it is sufficient to calculate the areas of the involved Cartesian blocks in a cross-section perpendicular to the well. If the permeability field provided in the cartesian system has a finer resolution in the well direction than the number of upscaled blocks desired for network simulation, the cylindrical grid can be coarsened by performing upscaling also in the well direction.

Fig. A.1 provides an illustration of the two grid systems arrangement with respect to the well and to each other. A definition of the naming convention for the quadrants of the cylinder surrounding the well is given (Q-I...Q-IV). Calculation of areas for all quadrants are based on geometry for quadrant Q-I.

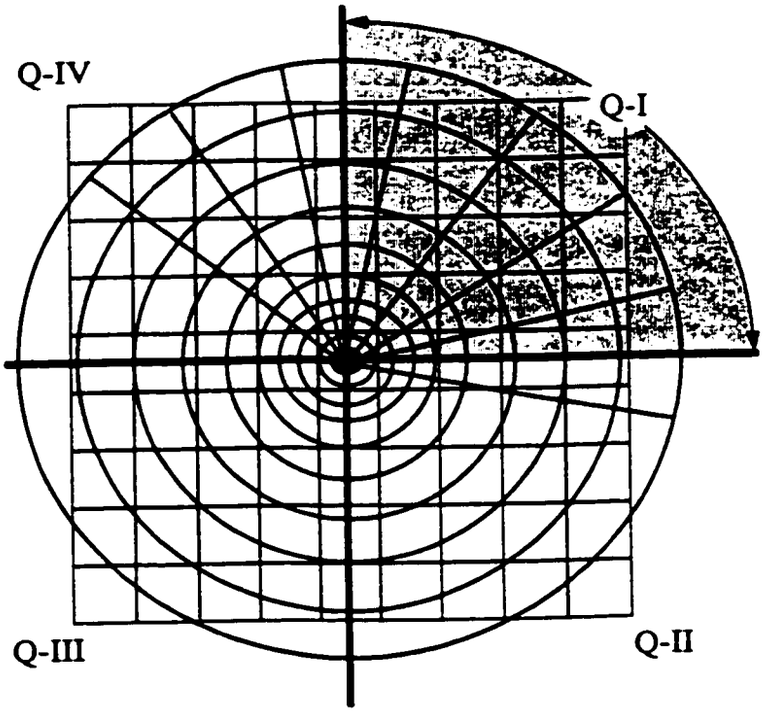

Fig. A.1: Grid System Illustration with Quadrant Naming Convention

A.1 Functional Requirements For Conversion Routine

Since a nearly unlimited variety of wellbore and near wellbore configurations may be encountered, the following functional requirements were defined to be met by the transformation routine:

a) The transformation should work for any cylindrical vs. Cartesian relative block size and relative orientation.

b) The routine should allow any parallelepiped shape of the Cartesian blocks.

c) To facilitate transformation for undulating wellbores, the routine should allow the well to have arbitrary x and z location of the wellbore within the near wellbore zone.

d) The routine should allow for a non-uniform Cartesian grid in the direction of the well.

e) The cylindrical grid division in the radial and θ direction should be entirely flexible to allow upscaling of any angular slice of the near wellbore zone.

A.2 Outline of Conversion Routine

The major functionality for the resulting transformation procedure can be outlined as follows:

1) The geometry and the block permeabilities for the Cartesian grid system are read from a file and stored in arrays.
2) The geometry of the cylindrical grid is defined together with arrays for the averaged cylindrical block permeabilities to be found.
3) For each cylindrical block, the areas of the Cartesian blocks enclosed in the cylindrical block are calculated and stored in an array with corresponding indices describing each Cartesian block location.
4) Average vertical (z) and horizontal (x) permeability for the cylindrical block are calculated based on a volumetric averaging of the involved cartesian block permeabilities.
5) An appearent radial permeability is calculated for the cylindrical block based on the x and z permeabilities and the theory presented for implementation of anisotrophy.

A.2 Methodology for Area Calculation of Cartesian Block Contributions

The most tedious part of the conversion routine is step (3), where the Cartesian grid block areas (equivalent to volumes) are being calculated. A general approach that would cover any relative block sizes and block orientations was stressed. The resulting methodology can be outlined as follows:

3.1) All geometry derived for the volume calculations were based on the blocks being located in the the first quadrant (Q-I). Thus, if the cylindrical block in question is located in quadrants Q-II to Q-IV, transformation of angles and grid block locations were performed. (The blocks were mirrored on to the first quadrant.)

3.2) A search is performed to find Cartesian grid intersection points enclosed in the cylindrical grid block. An array of intersection points is created. The number of intersection points determines further action.

3.2.1) If no intersection points are found, the cylindrical grid block may be completely enclosed in a Cartesian block or sliced by one or several Cartesian gridlines. For this case, a search is performed to find horizontal and vertical Cartesian gridlines intersecting the cylindrical block.

3.2.1.1) If no cartesian grid lines are found to intersect the cylindrical block the block is enclosed in a cartesian block and inherits the same permeability.

3.2.1.2) If one or more Cartesian grid lines are found to intersect the cylindrical block, the areas contributed by each Cartesian block are calculated. This case is illustrated in Figure A2.

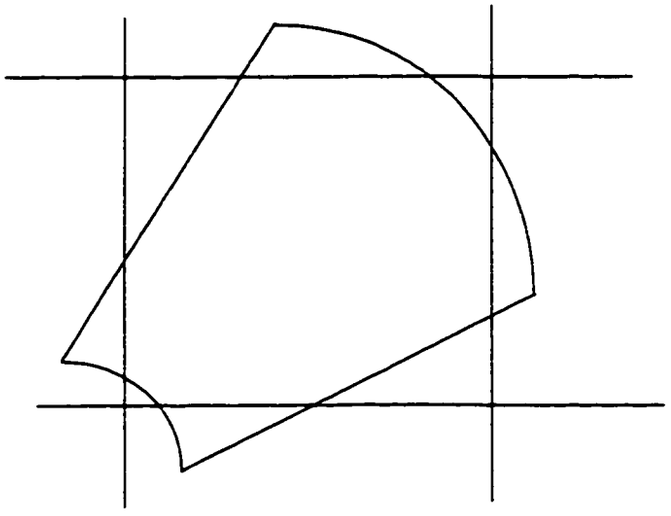

Fig. A.2: No Intersection Points Enclosed in Cylindrical Block - Block Divided by Cartesian Grid Lines

3.2.2) If only one intersection point is found inside the cylindrical block, it can be concluded that the cylindrical block is at least divided by two perpendicular Cartesian lines and, therefore, split into at least four different quadrants. An illustration with naming convention is given in Fig. A2.

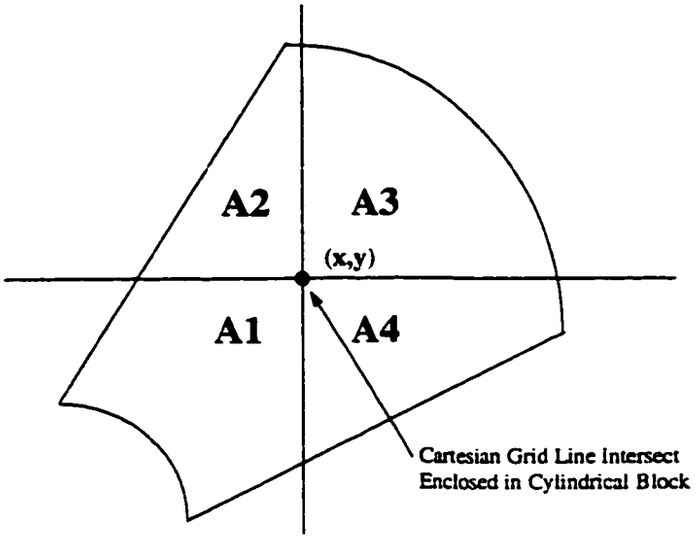

Fig. A.3: Basic Four Areas Associated With an Intersection Point

These four areas are always calculated and stored for the respective intersection point. More detailed description of this area calculation is given in the section below. However, one or several corners of the cylindrical block may be sliced of by a cartesian line that is not associated with the enclosed intersection point. Thus, also for this case, a search for vertical and horizontal intersecting grid lines not associated with enclosed intersection points is performed.

3.2.2.1) In the case where no further intersecting lines are found, the four Cartesian area contributions are stored in an array with indices pointing to the associated Cartesian grid blocks.

3.2.2.2) In the case where intersecting lines are found, which are not associated with intersection points, the associated block areas are calculated, and stored in an array together with previously calculated areas based on the detected intersection point.

3.2.3) If more than one intersection point is located inside the cylindrical block, the four areas (A1..A4) are calculated for each intersection point (see section A.4) and stored in an array with indices describing the Cartesian block associated with each area.Thus, the result from the above procedure is an (n x 4) array consisting of areas A1 to A4 corresponding each intersection point and independent Cartesian grid line cutting through the cylindrical block. Thus, n is the total number of intersects and lines found to be enclosed in the block. However, for $n > 1$, these areas are not necessarily the areas to be used in the averaging of permeability.

3.3) As most of the areas overlap, the correct areas are found by a procedure involving systematic subtraction of areas. This methodology is described in section A.3.

A.3 Methodology for Subtracting Areas to Find Cartesian Block Contributions

A general method for calculating the area contributions was developed. A variety of possibilities exist ranging from only a corner of the Cartesian block being enclosed in the cylindrical block to the entire Cartesian block being enclosed. The basic principle is to move horizontally through the intersection points starting from the bottom left corner. For each intersection point, and all four previously calculated areas, a check is performed whether the areas can be associated with another intersection point in the array. If the area can not be reached by a later point, it is being calculated.

If the area is to be calculated, the existance of surrounding enclosed intersection points is verified to determine which areas are to be subtracted from the area in question.

A simple example is used to illustrate the procedure. Fig. A.3 gives an example with 3 intersection points involved. If we start at A1,1 (for intersect I1), it is clear that none of the other involved points can be used for calculation of this area. Thus, this area must be considered. It is also seen that there are no intersection points to the left or below I1 enclosed in the cylindrical block. Thus, it can be concluded that the whole area A1,1 is to be used. If we now consider A1,2 (for point I1), we see that this area also can not be associated with any other point. Also, there are no existing neighbouring points within the cylindrical block, so again, the whole area is to be considered. When considering A1,3 and A 1,4, these poins can be reached by later appearing points, so for now these areas are

omitted from the calculations. If this procedure is followed through the cylindrical block, the following expressions for the areas result:

A1 = A1,1 A5 = A3,1-A2,1-A1,2

A2 = A2,1-A1,1 A6 = A3,4-A2,4

A3 = A2,4 A7 = A3,2

A4 = A1,2 A8 = A3,3

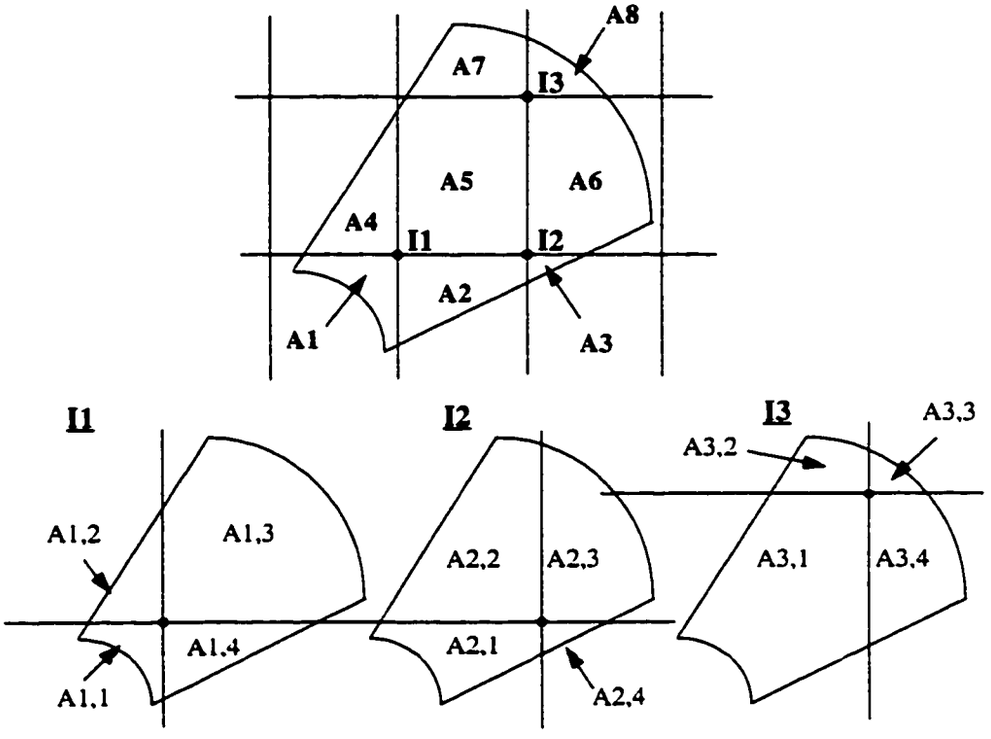

Fig. A.4: Example Involving Three Intersection Points

A.4 Calculation of the Four Quadrants of a Cylindrical Block

Figure A.5 illustrates the cylindrical block with the naming conventions for dimensions used in the calculation of the four quadrant areas divided by intersecting Cartesian grid lines. All distances are measured from the location of the well.

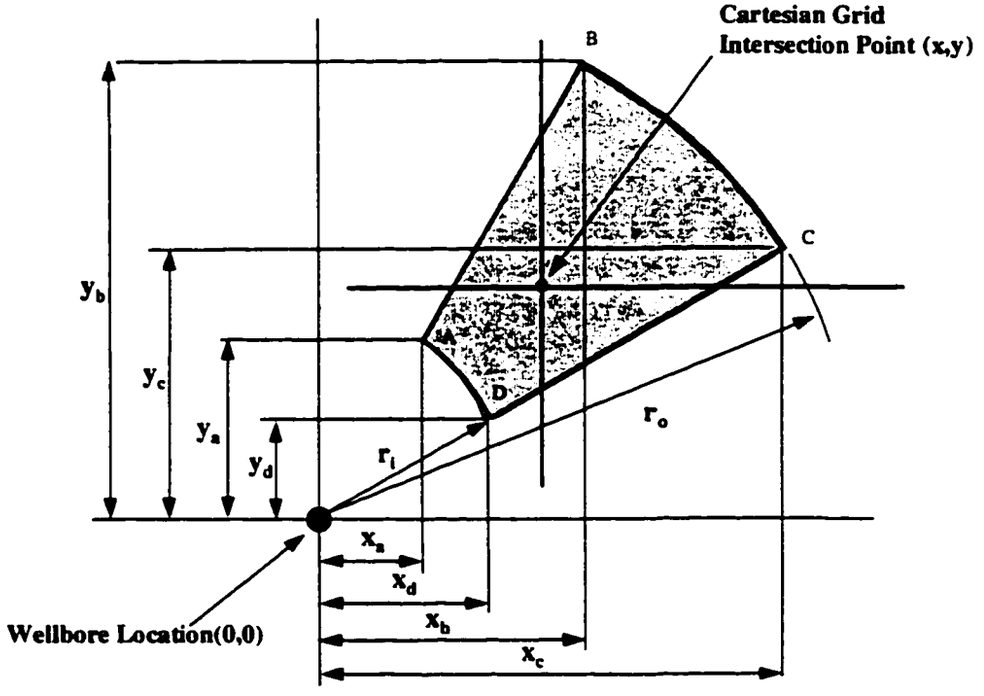

Fig. A.5: Cylindrical Block Configuration with Naming Convention for Dimensions Used in Area Calculations

For a cylindrical block divided by two perpendicular straight grid lines, there are 16 possible geometric shapes for the quadrants. The final equations for calculating these areas according to the location of the intersection point are given below:

Lower left quadrant:

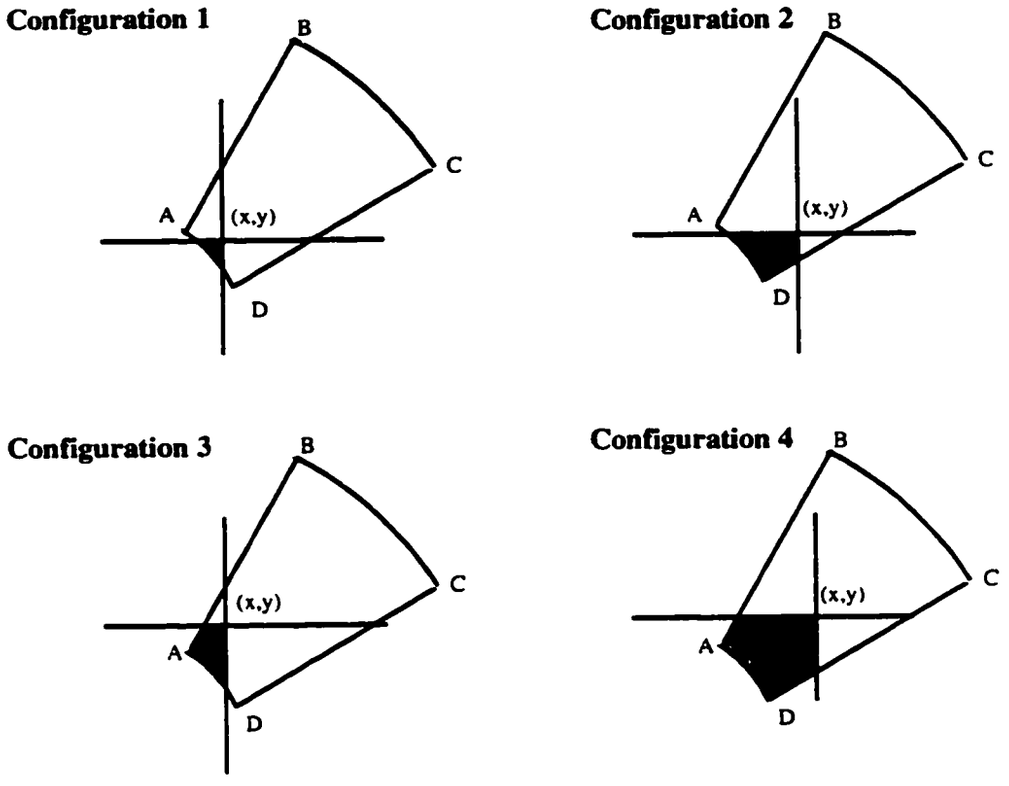

Fig. A.6: Four Possible Configurations for Calculation of Area of Lower Left Quadrant

First Configuration

$$A_{1,1} = \frac{1}{2}(x - \sqrt{r_i^2 - y^2}) \cdot (y - \sqrt{r_i^2 - x^2}) -$$

$$\frac{1}{2} r_i^2 (2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x - \sqrt{r_i^2 - y^2})^2 + (y - \sqrt{r_i^2 - x^2})^2}}{r_i}) - \qquad \text{(A1)}$$

$$\sin(2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x - \sqrt{r_i^2 - y^2})^2 + (y - \sqrt{r_i^2 - x^2})^2}}{r_i})))$$

Second Configuration

$$A_{1,2} = \frac{1}{2}(x_d - \sqrt{r_i^2 - y^2}) \cdot (y - \sqrt{r_i^2 - x_d{}^2}) -$$
$$\frac{1}{2} r_i^2 (2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x_d - \sqrt{r_i^2 - y^2})^2 + (y - \sqrt{r_i^2 - x_d{}^2})^2}}{r_i}) -$$
$$\sin(2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x_d - \sqrt{r_i^2 - y^2})^2 + (y - \sqrt{r_i^2 - x_d{}^2})^2}}{r_i}))) +$$
$$\frac{1}{2}(2y - \sqrt{r_i^2 - x_d{}^2} - \frac{x\sqrt{r_i^2 - x_d{}^2}}{x_d})(x - x_d) \qquad \text{(A2)}$$

Third Configuration

$$A_{1,3} = \frac{1}{2}(x - \sqrt{r_i^2 - y_a{}^2}) \cdot (y_a - \sqrt{r_i^2 - x^2}) -$$
$$\frac{1}{2} r_i^2 (2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x - \sqrt{r_i^2 - y_a{}^2})^2 + (y_a - \sqrt{r_i^2 - x^2})^2}}{r_i}) -$$
$$\sin(2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x - \sqrt{r_i^2 - y_a{}^2})^2 + (y_a - \sqrt{r_i^2 - x^2})^2}}{r_i}))) +$$
$$\frac{1}{2}(2x - \sqrt{r_i^2 - y_a{}^2} - \frac{x\sqrt{r_i^2 - y_a{}^2}}{y_a})(y - y_a) \qquad \text{(A3)}$$

Fourth Configuration

$$A_{1,4} = \frac{1}{2}(x_d - x_a)(y_a - y_d) -$$
$$\frac{1}{2} r_i^2 (2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x_d^2 - 2x_d x_a + x_a^2 + y_a{}^2 - 2y_a y_d + y_d{}^2}}{r_i}) -$$
$$\sin(2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(x_d^2 - 2x_d x_a + x_a^2 + y_a{}^2 - 2y_a y_d + y_d{}^2}}{r_i}))) +$$
$$\frac{1}{2}(2y_a - 2y_d - \frac{(x - x_d)y_d}{x_d})(x - x_d) + \frac{1}{2}(2x - 2x_a - \frac{(y - y_a)y_d}{y_a})(y - y_a) \qquad \text{(A4)}$$

Upper left quadrant:

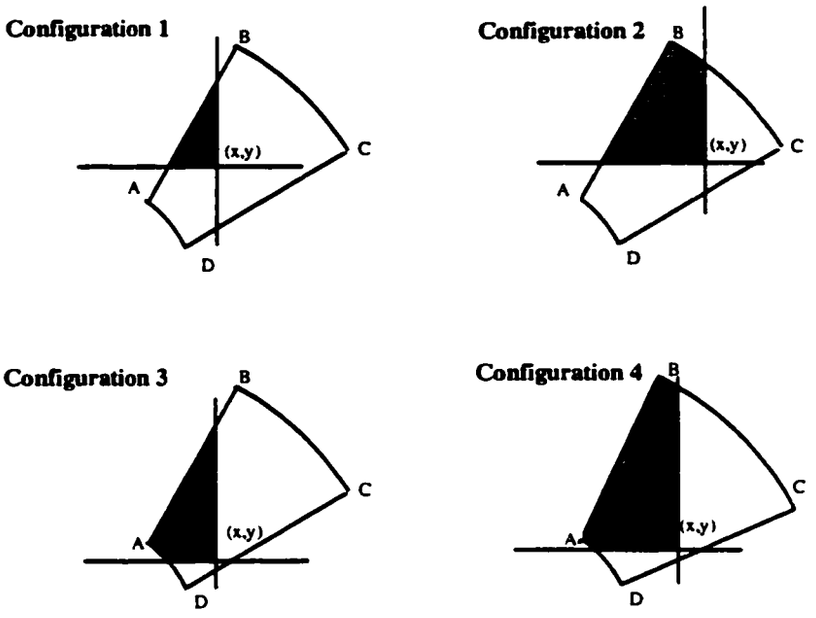

Fig. A.7: Four Possible Configurations for Calculation of Area of Upper Left Quadrant

First Configuration

$$A_{2,1} = \frac{1}{2}\frac{(\frac{x\,y_a}{x_a} - y)^2\,x_a}{y_a} \tag{A5}$$

Second Configuration

$$\begin{aligned} A_{2,2} = {} & \frac{1}{2}\frac{x_b(y_b - y)(y_b - \sqrt{r_o^2 - x^2} + \sqrt{r_o^2 + x^2} - y)}{y_b} + \\ & \frac{1}{2}(x - x_b)(y_b - \sqrt{r_o^2 - x^2}) + (x - x_b)(\sqrt{r_o^2 + x^2} - y) + \\ & \frac{1}{2}r_o^2(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x - x_b)^2 + (y_b - \sqrt{r_o^2 - x^2})^2}}{r_o}) - \\ & \sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x - x_b)^2 + (y_b - \sqrt{r_o^2 - x^2})^2}}{r_o}))) \end{aligned} \tag{A6}$$

Third Configuration

$$A_{2,3} = \frac{1}{2}(2x - \sqrt{r_i^2 - y^2} - x_a)(y_a - y) + \frac{1}{2}\frac{(x - x_a)^2 y_a}{x_a} -$$
$$\frac{1}{2} r_i^2 (2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_i^2 - y^2} - x_a)^2 + (y_a - y)^2}}{r_i}) -$$
$$\sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_i^2 - y^2} - x_a)^2 + (y_a - y)^2}}{r_i})) \qquad \text{(A7)}$$

Fourth Configuration

$$A_{2,4} = \frac{1}{2}(y_b - y_a)(x_b - x_a) + \frac{1}{2}(\sqrt{r_i^2 - y^2} - x_a)(y_a - y) +$$
$$(x - \sqrt{r_i^2 - y^2})(y_a - y) + (x - x_b)(\sqrt{r_o^2 - x^2} - y_a) +$$
$$\frac{1}{2}(x - x_a)(y_b - \sqrt{r_i^2 - y^2}) -$$
$$\frac{1}{2} r_i^2 (2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_i^2 - y^2} - x_a)^2 + (y_a - y)^2}}{r_i}) -$$
$$\sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_i^2 - y^2} - x_a)^2 + (y_a - y)^2}}{r_i})) +$$
$$\frac{1}{2} r_o^2 (2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x - x_b)^2 + (y_b - \sqrt{r_o^2 - x^2})^2}}{r_o}) -$$
$$\sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x - x_b)^2 + (y_b - \sqrt{r_o^2 - x^2})^2}}{r_o})) \qquad \text{(A8)}$$

Upper right quadrant:

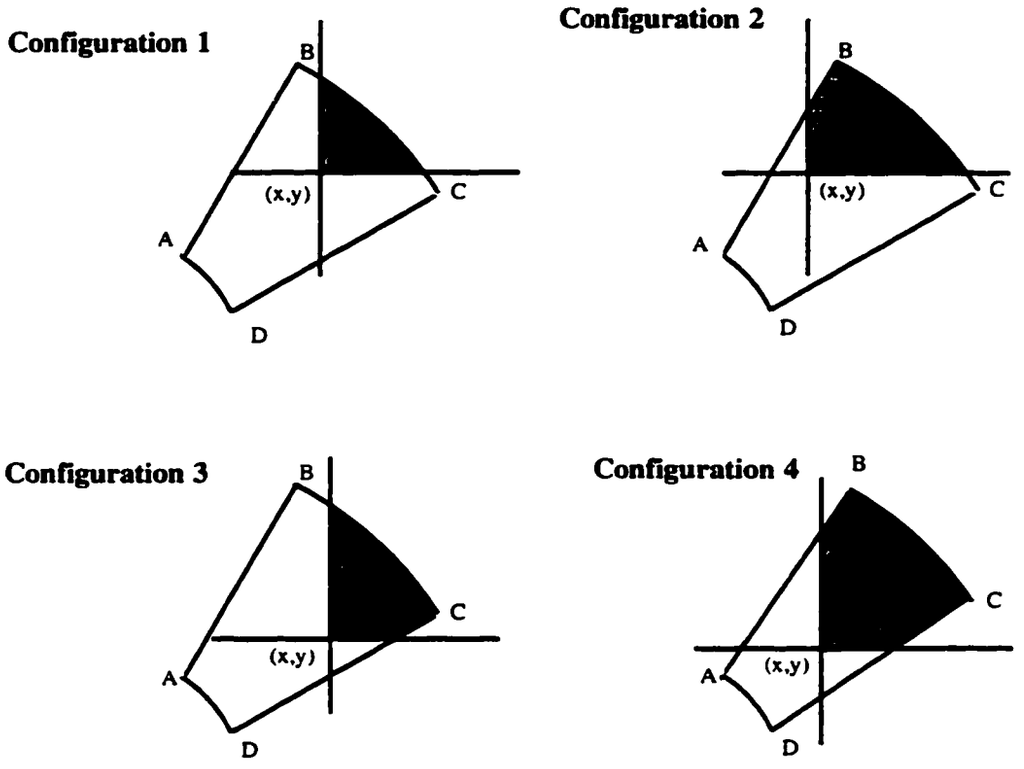

Fig. A.8: Four Possible Configurations for Calculation of Area of Upper Right Quadrant

First Configuration

$$A_{3,1} = \frac{1}{2}(\sqrt{r_o^2 - y^2} - x)(\sqrt{r_o^2 - x^2} - y) + \frac{1}{2} r_o^2 (2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(\sqrt{r_o^2 - y^2} - x)^2 + (\sqrt{r_o^2 - x^2} - y)^2}}{r_o}) - \sin(2 \cdot \sin^{-1}(\frac{1}{2} \frac{\sqrt{(\sqrt{r_o^2 - y^2} - x)^2 + (\sqrt{r_o^2 - x^2} - y)^2}}{r_o})))$$ (A9)

Second Configuration

$$A_{3,2} = \frac{1}{2}\left(\frac{xy_b}{x_b} - 2y + y_b\right)(x_b - x) + \frac{1}{2}\left(\sqrt{r_o^2 - y^2} - x_b\right)(y_b - y) + \frac{1}{2} r_o^2 \left(2 \cdot \sin^{-1}\left(\frac{1}{2}\frac{\sqrt{\left(\sqrt{r_o^2 - y^2} - x_b\right)^2 + (y_b - y)^2}}{r_o}\right) - \sin\left(2 \cdot \sin^{-1}\left(\frac{1}{2}\frac{\sqrt{\left(\sqrt{r_o^2 - y^2} - x_b\right)^2 + (y_b - y)^2}}{r_o}\right)\right)\right) \quad \text{(A10)}$$

Third Configuration

$$A_{3,3} = \frac{1}{2}\left(\frac{x_c\left(\frac{y_c(x_c - x)}{x_c} - y_c + y\right)}{y_c} + x_c - x\right)(y_c - y) + \frac{1}{2}(x_c - x)\left(\sqrt{r_o^2 - x^2} - y_c\right) + \frac{1}{2} r_o^2 \left(2 \cdot \sin^{-1}\left(\frac{1}{2}\frac{\sqrt{(x_c - x)^2 + \left(\sqrt{r_o^2 - x^2} - y_c\right)^2}}{r_o}\right) - \sin\left(2 \cdot \sin^{-1}\left(\frac{1}{2}\frac{\sqrt{(x_c - x)^2 + \left(\sqrt{r_o^2 - x^2} - y_c\right)^2}}{r_o}\right)\right)\right) \quad \text{(A11)}$$

Fourth Configuration

$$A_{3,4} = \frac{1}{2}(\sqrt{\frac{x^2 r_o^2}{x_b^2} - x^2} - 2y_c + y_b)(x_b - x) +$$

$$\frac{1}{2}(\sqrt{r_o^2 - y_c^2} - x + \frac{(y - \frac{y_c x}{\sqrt{r_o^2 - y_c^2}})\sqrt{r_o^2 - y_c^2}}{y_c})(y_c - y) +$$

$$\frac{1}{2}(\sqrt{r_o^2 - y_c^2} - x_b)(y_b - y_c) + \qquad \text{(A12)}$$

$$\frac{1}{2} r_o^2 (2 \cdot \sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_o^2 - y_c^2} - x_b)^2 + (y_b - y_c)^2}}{r_o}) -$$

$$\sin(2 \cdot \sin^{-1}(\frac{1}{2}\frac{\sqrt{(\sqrt{r_o^2 - y_c^2} - x_b)^2 + (y_b - y_c)^2}}{r_o})))$$

Lower right quadrant:

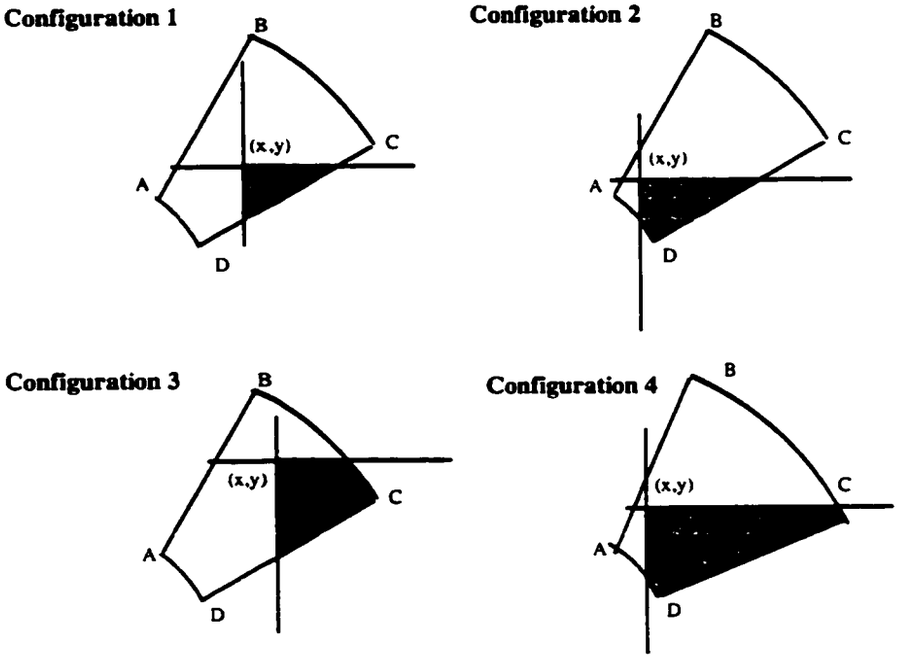

Fig. A.9: Four Possible Configurations for Calculation of Area of Lower Right Quadrant

First Configuration

$$A_{4,1} = \frac{1}{2}\left(\frac{yx_c}{y_c} - x\right)\left(y - \frac{xy_c}{x_c}\right) \tag{A13}$$

Second Configuration

$$\begin{aligned} A_{4,2} = {} & \frac{1}{2}(x_d - x)\left(\sqrt{r_i^2 - x^2} - y_d\right) + \left(y - \sqrt{r_i^2 + x^2}\right)(x_d - x) + \frac{1}{2}\frac{(y - y_d)^2\, x_d}{yd} - \\ & \frac{1}{2} r_i^2 \left(2\sin^{-1}\left(\frac{1}{2}\frac{\sqrt{(x_d - x)^2 + \left(\sqrt{r_i^2 - x^2} - y_d\right)^2}}{r_i}\right) - \right. \\ & \left. \sin\left(2\sin^{-1}\left(\frac{1}{2}\frac{\sqrt{(x_d - x)^2 + \left(\sqrt{r_i^2 - x^2} - y_d\right)^2}}{r_i}\right)\right)\right) \end{aligned} \tag{A14}$$

Third Configuration

$$\begin{aligned} A_{2,3} = {} & \frac{1}{2}\left(x_c - \sqrt{r_o^2 - y^2}\right)(y - y_c) + \frac{1}{2}\frac{(x_c - x)^2\, y_c}{x_c} + \\ & \left(\sqrt{r_o^2 - y^2} - x\right)(y - y_c) + \\ & \frac{1}{2} r_o^2 \left(2\sin^{-1}\left(\frac{1}{2}\frac{\sqrt{\left(x_c - \sqrt{r_o^2 - y^2}\right)^2 + (y - y_c)^2}}{r_o}\right) - \right. \\ & \left. \sin\left(2\sin^{-1}\left(\frac{1}{2}\frac{\sqrt{\left(x_c - \sqrt{r_o^2 - y^2}\right)^2 + (y - y_c)^2}}{r_o}\right)\right)\right) \end{aligned} \tag{A15}$$

Fourth Configuration

$$A_{2,4} = \frac{1}{2}(x_d - x)(\sqrt{r_i^2 - x^2} - y_d) + \frac{1}{2}(x_c - \sqrt{r_o^2 - y^2})(y - y_c) +$$

$$(x_d - x)(y_c - \sqrt{r_i^2 - x^2}) + \frac{1}{2}(x_c - x_d)(y_c - y_d) +$$

$$(\sqrt{r_o^2 - y^2} - x)(y - y_c) -$$

$$\frac{1}{2} r_i^2 (2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x_d - x)^2 + (\sqrt{r_i^2 - x^2} - y_d)^2}}{r_i}) -$$

$$\sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x_d - x)^2 + (\sqrt{r_i^2 - x^2} - y_d)^2}}{r_i}))) +$$

$$\frac{1}{2} r_o^2 (2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x_c - \sqrt{r_o^2 - y^2})^2 + (y - y_c)^2}}{r_o}) -$$

$$\sin(2\sin^{-1}(\frac{1}{2}\frac{\sqrt{(x_c - \sqrt{r_o^2 - y^2})^2 + (y - y_c)^2}}{r_o})))$$

APPENDIX B

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c     ROUTINE FOR AVERAGING PERMEABILITY IN A CYLINDER SEGMENT
c     PROGRAMMED BY KRISTIAN BREKKE - 1996
c     AT THE UNIVERSITY OF TULSA
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE KSEG(IZ,XWELL,YWELL,ALFA1T,ALFA2T,RI,RO,KRAD,IERR)

      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)       ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)       ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)       ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)      ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)      ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)      ! CARTESIAN Z PERMS

      INTEGER NLX   ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY   ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ   ! NUMBER OF LAYERS IN Z DIRECTION


cccccc INTERNAL VARIABLES ccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION ALFA1T    ! ANGLE OF 1ST. GRID LINE-FROM HRZ.-MODIFIED ANGLE TO Q1
      DOUBLE PRECISION ALFA2T    ! ANGLE OF 2ND. GRID LINE-FROM HRZ.-MODIFIED ANGLE TO Q1
      DOUBLE PRECISION ALFA1(2)  ! ANGLE OF 1ST. GRID LINE-FROM HRZ.-MODIFIED ANGLE TO Q1
      DOUBLE PRECISION ALFA2(2)  ! ANGLE OF 2ND. GRID LINE-FROM HRZ.-MODIFIED ANGLE TO Q1
      DOUBLE PRECISION RI        ! RADIUS OF INNER GRID LINE - FROM WELL CENTER
      DOUBLE PRECISION RO        ! RADIUS OF OUTER GRID LINE - FROM WELL CENTER
      DOUBLE PRECISION A(100)    ! AREA OF CARTESIAN BLOCK CONTRIBUTIONS
      DOUBLE PRECISION AT(100,2)! AREA OF CARTESIAN BLOCK CONTRIBUTIONS
      DOUBLE PRECISION ATOT      ! TOTAL AREA ADDED
      DOUBLE PRECISION ASEG      ! CALCULATED AREA OF SEGMENT
      DOUBLE PRECISION XWELL     ! X LOCATION OF WELL - FROM LEFT BOTTOM CORNER
      DOUBLE PRECISION YWELL     ! Y LOCATION OF WELL - FROM LEFT BOTTOM CORNER
      DOUBLE PRECISION XL(100)   ! X LOCATION INT. SECT LINES (FAKE PT.)
      DOUBLE PRECISION YL(100)   ! Y LOCATION INT. SECT LINES (FAKE PT.)
      DOUBLE PRECISION KRAD      ! RADIAL PERMEABILITY OF RADIAL BLOCK - RETURNED
      DOUBLE PRECISION KRX       ! X PERMEABILITY OF RADIAL BLOCK - RETURNED
      DOUBLE PRECISION KRY       ! Y PERMEABILITY OF RADIAL BLOCK - RETURNED
      DOUBLE PRECISION ALF       ! AVERAGE ANGLE

      INTEGER I,II
      INTEGER ISX(100),ISY(100)       ! ARRAY CONTAINING LOCATION OF INT.SECT PTS
      INTEGER IBX(100),IBY(100)       ! ARRAY CONTAINING LOCATION OF CONTR. BLOCKS
      INTEGER IBXT(100,2),IBYT(100,2)! ARRAY CONTAINING LOCATION OF CONTR. BLOCKS
      INTEGER IZ                      ! THE Z LAYER BEING AVERAGED
      INTEGER NIS                     ! NUMBER OF INTERSECT POINTS IN SEGMENT
      INTEGER NIB                     ! NUMBER OF BLOCKS IN SEGMENT
      INTEGER NIBT(2)                 ! NUMBER OF BLOCKS IN SEGMENT
      INTEGER IERR                    ! ERROR INDICATOR 1 - AREAS DO NOT MATCH
      INTEGER NSEGS                   ! NUMBER OF SEGMENTS - AFTER AXIS CHECK

cccccc DEFINITION OF COMMONS ccccccccccccccccccccccccccccccccccccccccc

      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

ccccccc  SET REMAINDING VARIABLES  - PRELIMINARY ccccccccccccccccccccc

      IERR=0
```

```
CCCCCC  CHECK TO SEE WHETHER SEGMENT SPANS ONE OF THE AXIS CCCCCCCCCCCC

      IF(ALFA1T.LT.3.14159/2.0.AND.ALFA2T.GT.3.14159/2.0)THEN
        ALFA1(1)=ALFA1T
        ALFA2(1)=3.14159/2.0
        ALFA1(2)=3.14159/2.0
        ALFA2(2)=ALFA2T
        NSEGS=2
      ELSEIF(ALFA1T.LT.3.14159.AND.ALFA2T.GT.3.14159)THEN
        ALFA1(1)=ALFA1T
        ALFA2(1)=3.14159
        ALFA1(2)=3.14159
        ALFA2(2)=ALFA2T
        NSEGS=2
      ELSEIF(ALFA1T.LT.3./2.*3.14159.AND.ALFA2T.GT.3./2.*3.14159)THEN
        ALFA1(1)=ALFA1T
        ALFA2(1)=3.0*3.14159/2.0
        ALFA1(2)=3.0*3.14159/2.0
        ALFA2(2)=ALFA2T
        NSEGS=2
      ELSEIF(ALFA1T.GT.ALFA2T)THEN
        ALFA1(1)=ALFA1T
        ALFA2(1)=3.14159*2.0
        ALFA1(2)=0.0
        ALFA2(2)=ALFA2T
        NSEGS=2
      ELSE
        ALFA1(1)=ALFA1T
        ALFA2(1)=ALFA2T
        NSEGS=1
      ENDIF

CCCCCC  FIND INTERSECTPOINTS INSIDE RADIAL SEGMENT CCCCCCCCCCCCCCCCCCCC
CCCCCC C FIND AREA CONTRIBUTIONS FOR BLOCKS AROUND INTERSECT POINTS CCCC

      DO I=1,NSEGS
        DO II=1,100                    !RESET INTERSECT VARIABLES
         ISX(II)=0
         ISY(II)=0
         IBX(II)=0
         IBY(II)=0
        ENDDO
        NIS=0
        NIB=0

        CALL INTERSECTP(IZ,RI,RO,ALFA1(I),ALFA2(I),XWELL,YWELL,ISX,ISY
     &                  ,NIS)

        CALL AREASEGP(IZ,XWELL,YWELL,ALFA1(I),ALFA2(I),RI,RO,
     &                ISX,ISY,NIS,A,IBX,IBY,NIB)

C        IF(NIB.GT.0)WRITE(*,*)'POINT CALCULATION'

CCCCCC   FIND INTERSECTING LINES THROUGH RADIAL SEGMENT CCCCCCCCCCCCCCCC
CCCCCC   FIND AREA CONTRIBUTIONS FOR BLOCKS AROUND INTERSECT LINE CCCC

        IF(NIB.EQ.0)THEN
          CALL INTERSECTL(IZ,RI,RO,ALFA1(I),ALFA2(I),XWELL,YWELL,
     &                    XL,YL,ISX,ISY,NIS)

          CALL AREASEGL(IZ,XWELL,YWELL,ALFA1(I),ALFA2(I),RI,RO,
     &                  ISX,ISY,NIS,XL,YL,A,IBX,IBY,NIB)
C          IF(NIB.GT.0)WRITE(*,*)'LINE CALCULATION'
C          DO II=1,NIS
C            WRITE(*,*)ISX(II),ISY(II)
C          ENDDO
        ENDIF

CCCCCC   IF CARTESIAN BLOCK TOTALLY SURROUNDS RADIAL BLOCK CCCC
CCCCCC   FIND WHICH BLOCK CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        IF(NIB.EQ.0)THEN
          CALL FINDBL(IZ,XWELL,YWELL,RI,RO,ALFA1(I),ALFA2(I),A,IBX,IBY
     &    ,NIB)
C          IF(NIB.GT.0)WRITE(*,*)'BLOCK ENCLOSES SEGMENT'
```

```
        ENDIF

        NIBT(I)=NIB
        DO II=1,NIB
          IBXT(II,I)=IBX(II)
          IBYT(II,I)=IBY(II)
          AT(II,I)=A(II)
        ENDDO
      ENDDO                                    ! I=1,NSEGS

      ATOT=0.0
      KRX=0.0
      KRY=0.0
      KRAD=0.0

      IF(ALFA2T.GT.ALFA1T)THEN
        ASEG=(RO**2.-RI**2.)*(ALFA2T-ALFA1T)/2.
      ELSEIF(ALFA1T.GT.ALFA2T)THEN
        ASEG=(RO**2.-RI**2.)*(2.*3.14159-ALFA1T+ALFA2T)/2.
      ENDIF

      DO I=1,NSEGS
        DO II=1,NIBT(I)
          ATOT=ATOT+AT(II,I)
          KRX=KRX+AT(II,I)*KXC(IZ,IBYT(II,I),IBXT(II,I))
          KRY=KRY+AT(II,I)*KYC(IZ,IBYT(II,I),IBXT(II,I))
        ENDDO
      ENDDO

      ALF=(ALFA1T+ALFA2T)/2.

      KRX=KRX/ATOT
      KRY=KRY/ATOT

      KRAD=KRX*KRY/(KRX-KRX*DCOS(ALF)**2.+KRY*DCOS(ALF)**2.)

      IF(ABS(ASEG-ATOT).GT.1E-4)IERR=1

      IF(IERR.EQ.1)THEN
       WRITE(*,*)'MISMATCH IN AREAS'
       WRITE(*,*)'ALFA1=',ALFA1*180/3.14159
       WRITE(*,*)'ALFA2=',ALFA2*180/3.14159
       WRITE(*,*)'RI=',RI
       WRITE(*,*)'RO=',RO
       WRITE(*,*)'TOTAL AREA CALCULATED =',ASEG
       WRITE(*,*)'TOTAL AREA ADDED      =',ATOT
      ENDIF

      END

CCCCCCCCCCCC END OF DRIVE ROUTINE CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c         ROUTINE TO FIND BLOCK AREAS IN A CYLINDRICAL SECTOR
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE AREASEGP(IZ,XWELL,YWELL,ALFA1,ALFA2,RI,RO,ISX,ISY,NIS,A
     &                    ,IBX,IBY,NIB)

      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)       ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)       ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)       ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)      ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)      ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)      ! CARTESIAN Z PERMS

      INTEGER NLX   ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY   ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ   ! NUMBER OF LAYERS IN Z DIRECTION

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION AREA         ! FUNCTION FOR BLOCK AREA
      DOUBLE PRECISION ALFA1        ! ANGLE TO FIRST (LEFT) GRID LINE
      DOUBLE PRECISION ALFA2        ! ANGLE TO SECOND GRID LINE
      DOUBLE PRECISION RI           ! INNER RADIUS OF SEGMENT
      DOUBLE PRECISION RO           ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION XCA          ! LEFT  BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCB          ! LEFT  UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCC          ! RIGHT UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCD          ! RIGHT BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION YCA          ! LEFT  BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCB          ! LEFT  UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCC          ! RIGHT UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCD          ! RIGHT BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION A(*)         ! AREA OF SEGMENTS
      DOUBLE PRECISION X            ! X-DISTANCE FROM WELL TO INT.SECT POINT
      DOUBLE PRECISION Y            ! Y-DISTANCE FROM WELL TO INT.SECT POINT
      DOUBLE PRECISION XWELL        ! X-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION YWELL        ! Y-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION AM(4,30,30)  ! AREA USED FOR INTERMEDIATE CALCULATIONS

      INTEGER NIS                  ! NUMBER OF INTERSECTING POINTS ENCLOSED IN SEG.
      INTEGER NIB                  ! NUMBER OF BLOCKS ENCLOSED IN SEG.
      INTEGER IZ                   ! Z - LAYER IN QUESTION
      INTEGER ISX(*)               ! ARRAY FOR K - POSITIONS OF INTERSECT POINTS
      INTEGER ISY(*)               ! ARRAY FOR J - POSITIONS OF INTERSECT POINTS
      INTEGER IBX(*)               ! ARRAY FOR K - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER IBY(*)               ! ARRAY FOR J - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER IADD                 ! ADDITIONAL COUNTER # OF BLOCKS>## OF INTERSECTS
      INTEGER I,J,II,IQ,IQT        ! COUNTERS
      INTEGER IDBG                 ! DEBUG INDICATOR
      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

cccccc CALCULATE CORNER POSITIONS FOR SEGMENT ccccccccccccccccccccccccc
cccccc TAKING INTO ACCOUNT THE QUADRANT IT IS LOCATED IN ccccccccccccccccc

      IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN  ! FIRST QUADRANT
        XCA=ABS(RI*DCOS(ALFA2))
        XCB=ABS(RO*DCOS(ALFA2))
        XCC=ABS(RO*DCOS(ALFA1))
        XCD=ABS(RI*DCOS(ALFA1))
        YCA=ABS(RI*DSIN(ALFA2))
        YCB=ABS(RO*DSIN(ALFA2))
        YCC=ABS(RO*DSIN(ALFA1))
        YCD=ABS(RI*DSIN(ALFA1))
      ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN  !SECOND QUADRANT
        XCD=ABS(RI*DCOS(3.14159-ALFA2))
```

```
        XCC=ABS(RO*DCOS(3.14159-ALFA2))
        XCB=ABS(RO*DCOS(3.14159-ALFA1))
        XCA=ABS(RI*DCOS(3.14159-ALFA1))
        YCD=ABS(RI*DSIN(3.14159-ALFA2))
        YCC=ABS(RO*DSIN(3.14159-ALFA2))
        YCB=ABS(RO*DSIN(3.14159-ALFA1))
        YCA=ABS(RI*DSIN(3.14159-ALFA1))
      ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN !THIRD QUADRANT
        XCA=ABS(RI*DCOS(ALFA2-3.14159))
        XCB=ABS(RO*DCOS(ALFA2-3.14159))
        XCC=ABS(RO*DCOS(ALFA1-3.14159))
        XCD=ABS(RI*DCOS(ALFA1-3.14159))
        YCA=ABS(RI*DSIN(ALFA2-3.14159))
        YCB=ABS(RO*DSIN(ALFA2-3.14159))
        YCC=ABS(RO*DSIN(ALFA1-3.14159))
        YCD=ABS(RI*DSIN(ALFA1-3.14159))
      ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN !FOURTH QUADRANT
        XCD=ABS(RI*DCOS(2.0*3.14159-ALFA2))
        XCC=ABS(RO*DCOS(2.0*3.14159-ALFA2))
        XCB=ABS(RO*DCOS(2.0*3.14159-ALFA1))
        XCA=ABS(RI*DCOS(2.0*3.14159-ALFA1))
        YCD=ABS(RI*DSIN(2.0*3.14159-ALFA2))
        YCC=ABS(RO*DSIN(2.0*3.14159-ALFA2))
        YCB=ABS(RO*DSIN(2.0*3.14159-ALFA1))
        YCA=ABS(RI*DSIN(2.0*3.14159-ALFA1))
      ENDIF

cccccc  RESET AREAS  ccccccccccccccccccccccccccccccccccccccccccccccccccc

      DO II=1,4
       DO I=1,30
        DO J=1,30
         IF(AM(II,I,J).NE.0.0)AM(II,I,J)=0.0
        ENDDO
       ENDDO
      ENDDO


      DO I=1,NIS      ! CALCULATE ALL QUADRANTS FOR ALL POINTS
        DO IQ=1,4
          X=ABS(XC(IZ,ISY(I),ISX(I))-XWELL)
          Y=ABS(YC(IZ,ISY(I),ISX(I))-YWELL)
          IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN              ! FIRST QUADRANT
            IF(IQ.EQ.1)IQT=1
            IF(IQ.EQ.2)IQT=2
            IF(IQ.EQ.3)IQT=3
            IF(IQ.EQ.4)IQT=4
          ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN    !SECOND QUADRANT
            IF(IQ.EQ.1)IQT=4
            IF(IQ.EQ.2)IQT=3
            IF(IQ.EQ.3)IQT=2
            IF(IQ.EQ.4)IQT=1
          ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN    !THIRD QUADRANT
            IF(IQ.EQ.1)IQT=3
            IF(IQ.EQ.2)IQT=4
            IF(IQ.EQ.3)IQT=1
            IF(IQ.EQ.4)IQT=2
          ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN  !FOURTH QUADRANT
            IF(IQ.EQ.1)IQT=2
            IF(IQ.EQ.2)IQT=1
            IF(IQ.EQ.3)IQT=4
            IF(IQ.EQ.4)IQT=3
           ENDIF
           AM(IQT,ISY(I),ISX(I))=AREA(IQ,RI,RO,XCA,XCB,XCC,XCD,
     &                                YCA,YCB,YCC,YCD,X,Y)
        ENDDO
       ENDDO
       IADD=0
       IDBG=1
       DO I=1,NIS    ! CONSTRUCT ARRAY WITH AREA CONTRIBUTIONS
        IF(AM(4,ISY(I),ISX(I)-1).LT.1.0D-6.AND.
     &     AM(3,ISY(I)-1,ISX(I)-1).LT.1.0D-6.AND.
     &     AM(2,ISY(I)-1,ISX(I)).LT.1.0D-6)THEN

ccccccc CALCULATE FIRST QUADRANT ccccccccccccccccccccccccccccccccccccc
```

```
      IF(AM(2,ISY(I)-1,ISX(I)).GT.1.0D-6.AND.
     &    AM(4,ISY(I),ISX(I)-1).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I),ISX(I)-1)
     &           -AM(1,ISY(I)-1,ISX(I))+AM(1,ISY(I)-1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - BOTH CORNER POINTS EXIST'
      ELSEIF(AM(1,ISY(I),ISX(I)-1).GT.1.0D-6)THEN
       IF(AM(1,ISY(I)-1,ISX(I)-1).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I),ISX(I)-1)
     &           -AM(4,ISY(I)-1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - LEFT CORNER POINT EXIST - DIAGONAL PT. EXIST'
       ELSE
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I),ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - LEFT CORNER EXIST - DIAGONAL PT. NOT EXIST'
       ENDIF
      ELSEIF(AM(1,ISY(I)-1,ISX(I)).GT.1.0D-6)THEN
       IF(AM(1,ISY(I)-1,ISX(I)-1).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I)-1,ISX(I))
     &           -AM(2,ISY(I)-1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - BOTTOM CORNER POINT EXIST - DIAG. PT. EXIST'
       ELSE
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I)-1,ISX(I))
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - BOTTOM CORNER EXIST - DIAGONAL PT. NOT EXIST'
       ENDIF
      ELSEIF(AM(1,ISY(I)-1,ISX(I)-1).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))-AM(1,ISY(I)-1,ISX(I)-1)
     &           -AM(2,ISY(I)-1,ISX(I)-1)-AM(4,ISY(I)-1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - ONLY DIAG. CORNER EXIST'
      ELSE
        IADD=IADD+1
        A(IADD)=AM(1,ISY(I),ISX(I))
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)
C        WRITE(*,*)'Q1 - NONE OF CORNER POINTS EXIST'
       ENDIF
      ENDIF

      IF(AM(3,ISY(I),ISX(I)-1).LT.1.0D-6)THEN

CCCCCCC  CALCULATE SECOND QUADRANT CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       IF(AM(1,ISY(I),ISX(I)-1).GT.1.0D-6.AND.
     &    AM(1,ISY(I)+1,ISX(I)).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(2,ISY(I),ISX(I))-AM(2,ISY(I)+1,ISX(I))
     &           -AM(2,ISY(I),ISX(I)-1)+AM(2,ISY(I)+1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)+1
C        WRITE(*,*)'Q2 - BOTH CORNER POINTS EXIST'
       ELSEIF(AM(1,ISY(I)+1,ISX(I)).GT.1.0D-6)THEN
        IF(AM(1,ISY(I)+1,ISX(I)-1).GT.1.0D-6)THEN
        IADD=IADD+1
        A(IADD)=AM(2,ISY(I),ISX(I))-AM(2,ISY(I)+1,ISX(I))
     &           -AM(1,ISY(I)+1,ISX(I)-1)
        IBX(IADD)=ISX(I)
        IBY(IADD)=ISY(I)+1
C        WRITE(*,*)'Q2 - TOP CORNER POINT EXIST - DIAGONAL PT. EXIST'
```

```
          ELSE
           IADD=IADD+1
           A(IADD)=AM(2,ISY(I),ISX(I))-AM(2,ISY(I)+1,ISX(I))
           IBX(IADD)=ISX(I)
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q2 - TOP CORNER EXIST - DIAGONAL PT. NOT EXIST'
          ENDIF
         ELSEIF(AM(1,ISY(I),ISX(I)-1).GT.1.0D-6)THEN
          IF(AM(1,ISY(I)+1,ISX(I)-1).GT.1.0D-6)THEN
           IADD=IADD+1
           A(IADD)=AM(2,ISY(I),ISX(I))-AM(2,ISY(I),ISX(I)-1)
     &              -AM(3,ISY(I)+1,ISX(I)-1)
           IBX(IADD)=ISX(I)
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q2 - LEFT CORNER POINT EXIST - DIAGONAL PT. EXIST'
          ELSE
           IADD=IADD+1
           A(IADD)=AM(2,ISY(I),ISX(I))-AM(2,ISY(I),ISX(I)-1)
           IBX(IADD)=ISX(I)
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q2 - LEFT CORNER EXIST - DIAGONAL PT. NOT EXIST'
          ENDIF
         ELSEIF(AM(1,ISY(I)+1,ISX(I)-1).GT.1.0D-6)THEN
           IADD=IADD+1
           A(IADD)=AM(2,ISY(I),ISX(I))-AM(1,ISY(I)+1,ISX(I)-1)
     &              -AM(2,ISY(I)+1,ISX(I)-1)-AM(3,ISY(I)+1,ISX(I)-1)
           IBX(IADD)=ISX(I)
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q2 - ONLY DIAG. CORNER EXIST'
         ELSE
           IADD=IADD+1
           A(IADD)=AM(2,ISY(I),ISX(I))
           IBX(IADD)=ISX(I)
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q2 - NONE OF CORNER POINTS EXIST'
         ENDIF
        ENDIF

CCCCCCC  CALCULATE THIRD QUADRANT CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

         IF(AM(1,ISY(I),ISX(I)+1).GT.1.0D-6.AND.
     &      AM(1,ISY(I)+1,ISX(I)).GT.1.0D-6)THEN
           IADD=IADD+1
           A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I),ISX(I)+1)
     &             -AM(3,ISY(I)+1,ISX(I))+AM(3,ISY(I)+1,ISX(I)+1)
           IBX(IADD)=ISX(I)+1
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q3 - BOTH CORNER POINTS EXIST'
         ELSEIF(AM(1,ISY(I),ISX(I)+1).GT.1.0D-6)THEN
          IF(AM(1,ISY(I)+1,ISX(I)+1).GT.1.0D-6)THEN
           IADD=IADD+1
           A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I),ISX(I)+1)
     &             -AM(2,ISY(I)+1,ISX(I)+1)
           IBX(IADD)=ISX(I)+1
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q3 - RIGHT CORNER POINT EXIST - DIAGONAL PT. EXIST'
          ELSE
           IADD=IADD+1
           A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I),ISX(I)+1)
           IBX(IADD)=ISX(I)+1
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q3 - RIGHT CORNER EXIST - DIAGONAL PT. NOT EXIST'
          ENDIF
         ELSEIF(AM(1,ISY(I)+1,ISX(I)).GT.1.0D-6)THEN
          IF(AM(1,ISY(I)+1,ISX(I)+1).GT.1.0D-6)THEN
           IADD=IADD+1
           A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I)+1,ISX(I))
     &             -AM(4,ISY(I)+1,ISX(I)+1)
           IBX(IADD)=ISX(I)+1
           IBY(IADD)=ISY(I)+1
C           WRITE(*,*)'Q3 - TOP CORNER POINT EXIST - DIAGONAL PT. EXIST'
          ELSE
           IADD=IADD+1
           A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I)+1,ISX(I))
           IBX(IADD)=ISX(I)+1
```

```
          IBY(IADD)=ISY(I)+1
C          WRITE(*,*)'Q3 - TOP CORNER EXIST - DIAGONAL PT. NOT EXIST'
         ENDIF
        ELSEIF(AM(1,ISY(I)+1,ISX(I)+1).GT.1.0D-6)THEN
          IADD=IADD+1
          A(IADD)=AM(3,ISY(I),ISX(I))-AM(3,ISY(I)+1,ISX(I)+1)
     &             -AM(2,ISY(I)+1,ISX(I)+1)-AM(4,ISY(I)+1,ISX(I)+1)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)+1
C          WRITE(*,*)'Q3 - ONLY DIAG. CORNER EXIST'
        ELSE
          IADD=IADD+1
          A(IADD)=AM(3,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)+1
C          WRITE(*,*)'Q3 - NONE OF CORNER POINTS EXIST'
        ENDIF

       IF(AM(3,ISY(I)-1,ISX(I)).LT.1.0D-6.AND.
     &    AM(2,ISY(I)-1,ISX(I)+1).LT.1.0D-6)THEN

CCCCCCC  CALCULATE FOURTH QUADRANT CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        IF(AM(1,ISY(I)-1,ISX(I)).GT.1.0D-6.AND.
     &     AM(1,ISY(I),ISX(I)+1).GT.1.0D-6)THEN
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(4,ISY(I)-1,ISX(I))
     &            -AM(4,ISY(I),ISX(I)+1)+AM(4,ISY(I)-1,ISX(I)+1)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - BOTH CORNER POINTS EXIST'
        ELSEIF(AM(1,ISY(I)-1,ISX(I)).GT.1.0D-6)THEN
         IF(AM(1,ISY(I)-1,ISX(I)+1).GT.1.0D-6)THEN
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(4,ISY(I)-1,ISX(I))
     &             -AM(3,ISY(I)-1,ISX(I)+1)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - BOTTOM CORNER POINT EXIST - DIAG. PT. EXIST'
         ELSE
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(4,ISY(I)-1,ISX(I))
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - BOTTOM CORNER EXIST - DIAGONAL PT. NOT EXIST'
         ENDIF
        ELSEIF(AM(1,ISY(I),ISX(I)+1).GT.1.0D-6)THEN
         IF(AM(1,ISY(I)-1,ISX(I)+1).GT.1.0D-6)THEN
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(4,ISY(I),ISX(I)+1)
     &             -AM(1,ISY(I)-1,ISX(I)+1)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - RIGHT CORNER POINT EXIST - DIAGONAL PT. EXIST'
         ELSE
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(4,ISY(I),ISX(I)+1)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - RIGHT CORNER EXIST - DIAGONAL PT. NOT EXIST'
         ENDIF
        ELSEIF(AM(1,ISY(I)-1,ISX(I)+1).GT.1.0D-6)THEN
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))-AM(3,ISY(I)-1,ISX(I)+1)
     &             -AM(4,ISY(I)-1,ISX(I)+1)-AM(1,ISY(I)-1,ISX(I)+4)
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - ONLY DIAG. CORNER EXIST'
        ELSE
          IADD=IADD+1
          A(IADD)=AM(4,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
C          WRITE(*,*)'Q4 - NONE OF CORNER POINTS EXIST'
        ENDIF
```

```
 ENDIF
ENDDO                          ! DO 1,NIS
NIB=IADD
END
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c    ROUTINE TO FIND AREA CONTRIBUTION OF A CARTESIAN BLOCK
c    IN A RADIAL SECTOR
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION FUNCTION AREA(IQ,RI,RO,XA,XB,XC,XD,YA,YB,YC,YD,
     &                  X,Y)
      IMPLICIT NONE

ccccccc DEFINITION OF VARIABLES - INPUT ccccccccccccccccccccccccccccccc

      DOUBLE PRECISION RI   ! INNER RADIUS OF SEGMENT
      DOUBLE PRECISION RO   ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION XA   ! LEFT  BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XB   ! LEFT  UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XC   ! RIGHT UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XD   ! RIGHT BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION YA   ! LEFT  BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YB   ! LEFT  UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YC   ! RIGHT UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YD   ! RIGHT BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION X    ! X - DISTANCE FROM WELL TO INTERSECT POINT
      DOUBLE PRECISION Y    ! Y - DISTANCE FROM WELL TO INTERSECT POINT

      INTEGER  IQ           ! INDICATOR OF WHICH QUADRANT TO CALCULATE
                            ! 1=LEFT RIGHT

ccccccc DEFINITION OF VARIABLES - INTERNAL ccccccccccccccccccccccccccccc

      DOUBLE PRECISION A,B,C,D,E,F,G,A2,B2,C2
      DOUBLE PRECISION RIX,RIY,RIYA,RIXD,ROX,ROY,ROYC
      DOUBLE PRECISION ALPHA,ALPHA2
      DOUBLE PRECISION ASEG,ASEG2

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      IF(IQ.EQ.1)THEN       ! FIRST QUADRANT CONTRIBUTION TO SEGMENT
        IF(X.GE.XA.AND.X.LT.XD.AND.Y.GE.YD.AND.Y.LT.YA)THEN
                                                ! FIRST CONFIGURATION
c         WRITE(*,*)'QUAD 1, CONF 1'
          RIY=(RI*RI-Y*Y)**0.5
          RIX=(RI*RI-X*X)**0.5
          A=X-RIY
          B=Y-RIX
          C=(A*A+B*B)**0.5
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          AREA=A*B/2.-ASEG

        ELSEIF(X.GE.XD.AND.X.LT.XC.AND.Y.GE.YD.AND.Y.LT.YA)THEN
                                                ! SECOND CONFIGURATION
c         WRITE(*,*)'QUAD 1, CONF 2'
          RIY=(RI*RI-Y*Y)**0.5
          RIXD=(RI*RI-XD*XD)**0.5
          A=XD-RIY
          B=Y-RIXD
          C=(A*A+B*B)**0.5
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          D=X/XD*RIXD
          E=Y-D
          AREA=A*B/2.+(B+E)/2.*(X-XD)-ASEG

        ELSEIF(X.GE.XA.AND.X.LT.XD.AND.Y.GE.YA.AND.Y.LT.YB)THEN
                                                ! THIRD CONFIGURATION
c         WRITE(*,*)'QUAD 1, CONF 3'
          RIYA=(RI*RI-YA*YA)**0.5
          RIX=(RI*RI-X*X)**0.5
          A=X-RIYA
          D=X-Y/YA*RIYA
```

```
          B=YA-RIX
          C=(A*A+B*B)**0.5
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          AREA=A*B/2.+(A+D)/2.*(Y-YA)-ASEG

        ELSEIF(X.GE.XD.AND.X.LT.XC.AND.Y.GE.YA.AND.Y.LT.YB)THEN
                                                  ! FOURTH CONFIGURATION
C         WRITE(*,*)'QUAD 1, CONF 4'
          A=XD-XA
          B=YA-YD
          C=(A*A+B*B)**0.5
          D=(X-XD)*YD/XD
          E=(Y-YA)*XA/YA
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          AREA=A*B/2.+(B+YA-YD-D)/2.*(X-XD)+(X-XA+X-XA-E)/2*(Y-YA)-ASEG

        ENDIF
      ELSEIF(IQ.EQ.2)THEN    ! SECOND QUADRANT CONTRIBUTION TO SEGMENT
        IF(X.GE.XA.AND.X.LT.XB.AND.Y.GE.YA.AND.Y.LT.YB)THEN
                                                  ! FIRST CONFIGURATION
C         WRITE(*,*)'QUAD 2, CONF 1'
          A=X*YA/XA-Y
          B=A*XA/YA
          AREA=A*B/2.

        ELSEIF(X.GE.XB.AND.X.LT.XC.AND.Y.GE.YA.AND.Y.LT.YB)THEN
                                                  ! SECOND CONFIGURATION
C         WRITE(*,*)'QUAD 2, CONF 2'
          ROX=(RO*RO-X*X)**0.5
          A=X-XB
          B=YB-ROX
          C=(A*A+B*B)**0.5
          D=ROX-Y
          E=XB/YB*(YB-Y)
          ALPHA=2.0*DASIN(C/2./RO)
          ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
          AREA=E*(B+D)/2.+A*B/2.+A*D+ASEG

        ELSEIF(X.GE.XA.AND.X.LT.XB.AND.Y.GE.YD.AND.Y.LT.YA)THEN
                                                  ! THIRD CONFIGURATION
C         WRITE(*,*)'QUAD 2, CONF 3'
          RIY=(RI*RI-Y*Y)**0.5
          A=RIY-XA
          B=YA-Y
          C=(A*A+B*B)**0.5
          D=(X-XA)/XA*RI
          E=YA*D/RI
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          AREA=(X-RIY+X-XA)/2.*(YA-Y)+(X-XA)*E/2.-ASEG

        ELSEIF(X.GE.XB.AND.X.LT.XC.AND.Y.GE.YD.AND.Y.LT.YA)THEN
                                                  ! FOURTH CONFIGURATION
C         WRITE(*,*)'QUAD 2, CONF 4'
          RIY=(RI*RI-Y*Y)**0.5
          ROX=(RO*RO-X*X)**0.5
          A=RIY-XA
          B=YA-Y
          C=(A*A+B*B)**0.5
          ALPHA=2.0*DASIN(C/2./RI)
          ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
          A2=X-XB
          B2=YB-ROX
          C2=(A2*A2+B2*B2)**0.5
          ALPHA2=2.0*DASIN(C2/2./RO)
          ASEG2=0.5*RO*RO*(ALPHA2-DSIN(ALPHA2))
          AREA=(YB-YA)*(XB-XA)/2.+A*B/2.+(X-RIY)*B+A2*(ROX-YA)+A2*B2/2.
     &    +ASEG2-ASEG

        ENDIF
      ELSEIF(IQ.EQ.3)THEN    ! THIRD QUADRANT CONTRIBUTION TO SEGMENT
        IF(X.GE.XB.AND.X.LT.XC.AND.Y.GE.YC.AND.Y.LT.YB)THEN
                                                  ! FIRST CONFIGURATION
```

```
C          WRITE(*,*)'QUAD 3, CONF 1'
           ROX=(RO*RO-X*X)**0.5
           ROY=(RO*RO-Y*Y)**0.5
           A=ROY-X
           B=ROX-Y
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RO)
           ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
           AREA=A*B/2.+ASEG

         ELSEIF(X.GE.XA.AND.X.LT.XB.AND.Y.GE.YC.AND.Y.LT.YB)THEN
                                                  ! SECOND CONFIGURATION
C          WRITE(*,*)'QUAD 3, CONF 2'
           ROY=(RO*RO-Y*Y)**0.5
           A=ROY-XB
           B=YB-Y
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RO)
           ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
           D=ROY-X-A
           E=X/XB*YB-Y
           AREA=(E+B)/2*D+A*B/2.+ASEG

         ELSEIF(X.GE.XB.AND.X.LT.XC.AND.Y.GE.YD.AND.Y.LT.YC)THEN
                                                  ! THIRD CONFIGURATION
C          WRITE(*,*)'QUAD 3, CONF 3'
           ROX=(RO*RO-X*X)**0.5
           A=XC-X
           B=ROX-YC
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RO)
           ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
           D=YC-Y
           E=YC/XC*A
           F=E-D
           G=XC/YC*F
           AREA=(G+A)/2.*D+A*B/2.+ASEG

         ELSEIF(X.GE.XA.AND.X.LT.XB.AND.Y.GE.YD.AND.Y.LT.YC)THEN
                                                  ! FOURTH CONFIGURATION
C          WRITE(*,*)'QUAD 3, CONF 4'
           ROYC=(RO*RO-YC*YC)**0.5
           A=ROYC-XB
           B=YB-YC
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RO)
           ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
           D=YC-Y
           E=XB-X
           F=((X*RO/XB)**2.-X**2)**0.5-D-Y
           G=X*(Y-YC*X/ROYC)/(YC*X/ROYC)
           AREA=(F+B)/2.*E+(A+E+G)/2.*D+A*B/2+ASEG
         ENDIF

       ELSEIF(IQ.EQ.4)THEN    ! FOURTH QUADRANT CONTRIBUTION TO SEGMENT
         IF(X.GE.XD.AND.X.LT.XC.AND.Y.GE.YD.AND.Y.LT.YC)THEN
                                                  ! FIRST CONFIGURATION
C          WRITE(*,*)'QUAD 4, CONF 1'
           A=Y*XC/YC-X
           B=Y-X*YC/XC
           AREA=A*B/2.

         ELSEIF(X.GE.XA.AND.X.LT.XD.AND.Y.GE.YD.AND.Y.LT.YC)THEN
                                                  ! SECOND CONFIGURATION
C          WRITE(*,*)'QUAD 4, CONF 2'
           RIX=(RI*RI-X*X)**0.5
           A=XD-X
           B=RIX-YD
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RI)
           ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
           D=(Y-YD)*XD/YD
           AREA=A*B/2.+(Y-YD-B)*A+(Y-YD)*D/2.-ASEG

         ELSEIF(X.GE.XD.AND.X.LT.XC.AND.Y.GE.YC.AND.Y.LT.YB)THEN
```

```
                                                    ! THIRD CONFIGURATION
C          WRITE(*,*)'QUAD 4, CONF 3'
           ROY=(RO*RO-Y*Y)**0.5
           A=XC-ROY
           B=Y-YC
           C=(A*A+B*B)**0.5
           ALPHA=2.0*DASIN(C/2./RO)
           ASEG=0.5*RO*RO*(ALPHA-DSIN(ALPHA))
           D=ROY-X
           E=(XC-X)*YC/XC
           AREA=A*B/2.+E*(XC-X)/2.+D*B+ASEG

         ELSEIF(X.GE.XA.AND.X.LT.XD.AND.Y.GE.YC.AND.Y.LT.YB)THEN
                                                    ! FOURTH CONFIGURATION
C          WRITE(*,*)'QUAD 4, CONF 4'
           RIX=(RI*RI-X*X)**0.5
           ROY=(RO*RO-Y*Y)**0.5
           A=XD-X
           B=RIX-YD
           C=(A*A+B*B)**0.5
           A2=XC-ROY
           B2=Y-YC
           C2=(A2*A2+B2*B2)**0.5
           ALPHA=2.0*DASIN(C/2./RI)
           ASEG=0.5*RI*RI*(ALPHA-DSIN(ALPHA))
           ALPHA2=2.0*DASIN(C2/2./RO)
           ASEG2=0.5*RO*RO*(ALPHA2-DSIN(ALPHA2))
           D=YC-RIX
           E=ROY-X
           AREA=A*B/2.+A2*B2/2.+A*D+(XC-XD)*(YC-YD)/2.+E*B2+ASEG2-ASEG

         ENDIF
       ENDIF
       RETURN
       END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C       ROUTINE TO FIND INTERSECT POINTS ENCLOSED IN SEGMENT
C
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE INTERSECTP(IZ,RI,RO,ALFA1,ALFA2,XWELL,YWELL,ISX,
     &                      ISY,NIS)
      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)       ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)       ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)       ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)      ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)      ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)      ! CARTESIAN Z PERMS

      INTEGER NLX   ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY   ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ   ! NUMBER OF LAYERS IN Z DIRECTION

cccccc INTERNAL VAR. ccccccccccccccccccccccccccccccccccccccccccc

      INTEGER NIS                 ! NUMBER OF INTERSECT POINTS IN SEGMENT
      INTEGER IZ                  ! CURRENT Z-LAYER
      INTEGER ISX(*),ISY(*)       ! ARRAY WITH LOCATION OF INTERSECT PTS.
      DOUBLE PRECISION XWELL      ! X LOCATION OF WELL
      DOUBLE PRECISION YWELL      ! Y LOCATION OF WELL
      DOUBLE PRECISION RI         ! INNER RADIUS OF SEG
      DOUBLE PRECISION RO         ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION ALFA1      ! FIRST ANGLE (LEFT) - GRID LINE
      DOUBLE PRECISION ALFA2      ! SECOND ANGLE - GRID LINE
      DOUBLE PRECISION RINTC      ! RADIUS OF INT.SECT
      DOUBLE PRECISION ALFINTC    ! ANGLE OF INT.SECT

      INTEGER I,J,K,II

      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

      II=0
      DO I=IZ,IZ
        DO J=1,NLY
          DO K=1,NLX
            RINTC=((XC(I,J,K)-XWELL)**2.+
     &       (YC(I,J,K)-YWELL)**2.)**0.5
            ALFINTC=DATAN(ABS(YC(I,J,K)-YWELL)/ABS(XC(I,J,K)-XWELL))
            IF(YC(I,J,K)-YWELL.GT.0.0.AND.        !SECOND QUADRANT
     &         XC(I,J,K)-XWELL.LT.0.0)THEN
              ALFINTC=3.14159-ALFINTC
            ELSEIF(YC(I,J,K)-YWELL.LT.0.0.AND.    !THIRD QUADRANT
     &             XC(I,J,K)-XWELL.LT.0.0)THEN
              ALFINTC=ALFINTC+3.14159
            ELSEIF(YC(I,J,K)-YWELL.LT.0.0.AND.    !FOURTH QUADRANT
     &             XC(I,J,K)-XWELL.GT.0.0)THEN
              ALFINTC=2*3.14159-ALFINTC
            ENDIF
            IF(RINTC.GT.RI.AND.RINTC.LT.RO.AND.
     &        ALFINTC.GT.ALFA1.AND.ALFINTC.LT.ALFA2)THEN
              II=II+1
              ISX(II)=K
              ISY(II)=J
            ENDIF
          ENDDO
        ENDDO
      ENDDO
      NIS=II
      END

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C       ROUTINE TO FIND INTERSECTING LINES ENCLOSED IN SEGMENT
C
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE INTERSECTL(IZ,RI,RO,ALFA1,ALFA2,XWELL,YWELL,XL,YL,
     &                      ISX,ISY,NIS)
      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)     ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)     ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)     ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)    ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)    ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)    ! CARTESIAN Z PERMS

      INTEGER NLX   ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY   ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ   ! NUMBER OF LAYERS IN Z DIRECTION

cccccc INTERNAL VAR. ccccccccccccccccccccccccccccccccccccccccc

      INTEGER NIS               ! NUMBER OF INTERSECT POINTS IN SEGMENT
      INTEGER IZ                ! CURRENT Z-LAYER
      INTEGER ISX(*),ISY(*)     ! ARRAY WITH LOCATION OF INTERSECT PTS.

      DOUBLE PRECISION RI       ! INNER RADIUS OF SEG
      DOUBLE PRECISION RO       ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION ALFA1    ! FIRST ANGLE (LEFT) - GRID LINE
      DOUBLE PRECISION ALFA2    ! SECOND ANGLE - GRID LINE
      DOUBLE PRECISION ALFA1N   ! FIRST ANGLE (LEFT) - GRID LINE - NORMALISED TO 1ST Q
      DOUBLE PRECISION ALFA2N   ! SECOND ANGLE - GRID LINE - NORMALISED TO 1ST Q
      DOUBLE PRECISION XWELL    ! X LOCATION OF WELL - FROM LEFT BOTTOM CORNER
      DOUBLE PRECISION YWELL    ! Y LOCATION OF WELL - FROM LEFT BOTTOM CORNER
      DOUBLE PRECISION XCA      ! LEFT  BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCB      ! LEFT  UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCC      ! RIGHT UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCD      ! RIGHT BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION YCA      ! LEFT  BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCB      ! LEFT  UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCC      ! RIGHT UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCD      ! RIGHT BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YL(*)    ! ARRAY WITH Y-LOCATION OF INTERSECT LINES
      DOUBLE PRECISION XL(*)    ! ARRAY WITH X-LOCATION OF INTERSECT LINES

      INTEGER I,J,K,II

      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

cccccc CALCULATE CORNER POINTS OF RADIAL SEGMENT ccccccccccccccccccccc

      IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN      !FIRST QUADRANT
        ALFA1N=ALFA1
        ALFA2N=ALFA2
        XCA=XWELL+ABS(RI*DCOS(ALFA2N))
        XCB=XWELL+ABS(RO*DCOS(ALFA2N))
        XCC=XWELL+ABS(RO*DCOS(ALFA1N))
        XCD=XWELL+ABS(RI*DCOS(ALFA1N))
        YCA=XWELL+ABS(RI*DSIN(ALFA2N))
        YCB=XWELL+ABS(RO*DSIN(ALFA2N))
        YCC=XWELL+ABS(RO*DSIN(ALFA1N))
        YCD=XWELL+ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN ! SECOND QUADRANT
        ALFA1N=3.14159-ALFA1
        ALFA2N=3.14159-ALFA2
        XCD=XWELL-ABS(RI*DCOS(ALFA2N))
        XCA=XWELL-ABS(RO*DCOS(ALFA2N))
        XCB=XWELL-ABS(RO*DCOS(ALFA1N))
        XCC=XWELL-ABS(RI*DCOS(ALFA1N))
        YCD=YWELL+ABS(RI*DSIN(ALFA2N))
        YCA=YWELL+ABS(RO*DSIN(ALFA2N))
        YCB=YWELL+ABS(RO*DSIN(ALFA1N))
```

```
        YCC=YWELL+ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN ! THIRD QUADRANT
        ALFA1N=ALFA1-3.14159
        ALFA2N=ALFA2-3.14159
        XCC=XWELL-ABS(RI*DCOS(ALFA2N))
        XCD=XWELL-ABS(RO*DCOS(ALFA2N))
        XCA=XWELL-ABS(RO*DCOS(ALFA1N))
        XCB=XWELL-ABS(RI*DCOS(ALFA1N))
        YCC=YWELL-ABS(RI*DSIN(ALFA2N))
        YCD=YWELL-ABS(RO*DSIN(ALFA2N))
        YCA=YWELL-ABS(RO*DSIN(ALFA1N))
        YCB=YWELL-ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN ! FOURTH QUADRANT
        ALFA1N=2.*3.14159-ALFA1
        ALFA2N=2.*3.14159-ALFA2
        XCB=XWELL+ABS(RI*DCOS(ALFA2N))
        XCC=XWELL+ABS(RO*DCOS(ALFA2N))
        XCD=XWELL+ABS(RO*DCOS(ALFA1N))
        XCA=XWELL+ABS(RI*DCOS(ALFA1N))
        YCB=YWELL-ABS(RI*DSIN(ALFA2N))
        YCC=YWELL-ABS(RO*DSIN(ALFA2N))
        YCD=YWELL-ABS(RO*DSIN(ALFA1N))
        YCA=YWELL-ABS(RI*DSIN(ALFA1N))
      ENDIF

      DO II=1,100      !RESET X AND Y LOCATION OF FAKE PTS.
       XL(II)=0.0
       YL(II)=0.0
      ENDDO

      II=0

      DO I=IZ,IZ
        DO J=1,NLY
          DO K=1,NLX
            IF(YC(I,J,K).LT.YCB.AND.YC(I,J,K).GT.YCD)THEN
              IF(XC(I,J,K).GT.XCB-(XCB-XCD)*(YCB-YC(I,J,K))
     &          /(YCB-YCD))THEN
                II=II+1
                ISY(II)=J
                ISX(II)=K
C                WRITE(*,*)'HORIZ',II,ISX(II),ISY(II)
                XL(II)=XCB-(XCB-XCD)*(YCB-YC(I,J,K))/(YCB-YCD)
                YL(II)=0.0
                EXIT
              ENDIF
            ENDIF
          ENDDO
        ENDDO

        DO K=1,NLX
          DO J=1,NLY
            IF(XC(I,J,K).LT.XCC.AND.XC(I,J,K).GT.XCA)THEN
              IF(YC(I,J,K).GT.YCA-(YCA-YCC)*(XCA-XC(I,J,K))
     &          /(XCA-XCC))THEN
                II=II+1
                ISY(II)=J
                ISX(II)=K
C                WRITE(*,*)'VERTIC',II,ISX(II),ISY(II)
                XL(II)=0.0
                YL(II)=YCA-(YCA-YCC)*(XCA-XC(I,J,K))/(XCA-XCC)
                EXIT
              ENDIF
            ENDIF
          ENDDO
        ENDDO
      ENDDO        !IZ

      NIS=II
      END
```

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c         ROUTINE TO FIND BLOCK AREAS IN A RADIAL SECTOR
c         DIVIDED ONLY BY A LINE  (HORIZONTAL OR VERTICAL)
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE AREASEGL(IZ,XWELL,YWELL,ALFA1,ALFA2,RI,RO,ISX,ISY,NIS,
     &                    XL,YL,A,IBX,IBY,NIB)

      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)      ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)      ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)      ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)     ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)     ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)     ! CARTESIAN Z PERMS

      INTEGER NLX  ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY  ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ  ! NUMBER OF LAYERS IN Z DIRECTION

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION AREA           ! FUNCTION FOR BLOCK AREA
      DOUBLE PRECISION ALFA1          ! ANGLE TO FIRST (LEFT) GRID LINE
      DOUBLE PRECISION ALFA2          ! ANGLE TO SECOND GRID LINE
      DOUBLE PRECISION RI             ! INNER RADIUS OF SEGMENT
      DOUBLE PRECISION RO             ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION XCA            ! LEFT  BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCB            ! LEFT  UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCC            ! RIGHT UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCD            ! RIGHT BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION YCA            ! LEFT  BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCB            ! LEFT  UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCC            ! RIGHT UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCD            ! RIGHT BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION A(*)           ! AREA OF SEGMENTS
      DOUBLE PRECISION X              ! X-DISTANCE FROM WELL TO INT.SECT POINT
      DOUBLE PRECISION Y              ! Y-DISTANCE FROM WELL TO INT.SECT POINT
      DOUBLE PRECISION XWELL          ! X-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION YWELL          ! Y-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION XL(*)          ! X-DISTANCE TO FAKE PT.
      DOUBLE PRECISION YL(*)          ! Y-DISTANCE TO FAKE PT.
      DOUBLE PRECISION AM(2,4,30,30) ! AREA USED FOR INTERMEDIATE CALCULATIONS
                                      ! FIRST COUNTER 1 HORIZONTAL DIVISION, 2 VERTICAL

      INTEGER NIS                 ! NUMBER OF INTERSECTING POINTS ENCLOSED IN SEG.
      INTEGER NIB                 ! NUMBER OF BLOCKS ENCLOSED IN SEG.
      INTEGER IZ                  ! Z - LAYER IN QUESTION
      INTEGER ISX(*)              ! ARRAY FOR K - POSITIONS OF INTERSECT POINTS
      INTEGER ISY(*)              ! ARRAY FOR J - POSITIONS OF INTERSECT POINTS
      INTEGER IBX(*)              ! ARRAY FOR K - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER IBY(*)              ! ARRAY FOR J - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER IADD                ! ADDITIONAL COUNTER # OF BLOCKS>=# OF INTERSECTS
      INTEGER I,J,II,IQ,IQT       ! COUNTERS
      INTEGER ISUB,ISTAT          ! DUMMIES
      INTEGER IHV(100)            ! HORIZONTAL OR VERTICAL INDICATOR
      INTEGER INDHV               ! HV INDICATOR

      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

cccccc CALCULATE CORNER POSITIONS FOR SEGMENT ccccccccccccccccccccccccc
cccccc TAKING INTO ACCOUNT THE QUADRANT IT IS LOCATED IN ccccccccccccccccc

      IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN  ! FIRST QUADRANT
        XCA=ABS(RI*DCOS(ALFA2))
        XCB=ABS(RO*DCOS(ALFA2))
        XCC=ABS(RO*DCOS(ALFA1))
```

```
          XCD=ABS(RI*DCOS(ALFA1))
          YCA=ABS(RI*DSIN(ALFA2))
          YCB=ABS(RO*DSIN(ALFA2))
          YCC=ABS(RO*DSIN(ALFA1))
          YCD=ABS(RI*DSIN(ALFA1))
        ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN  !SECOND QUADRANT
          XCD=ABS(RI*DCOS(3.14159-ALFA2))
          XCC=ABS(RO*DCOS(3.14159-ALFA2))
          XCB=ABS(RO*DCOS(3.14159-ALFA1))
          XCA=ABS(RI*DCOS(3.14159-ALFA1))
          YCD=ABS(RI*DSIN(3.14159-ALFA2))
          YCC=ABS(RO*DSIN(3.14159-ALFA2))
          YCB=ABS(RO*DSIN(3.14159-ALFA1))
          YCA=ABS(RI*DSIN(3.14159-ALFA1))
        ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN !THIRD QUADRANT
          XCA=ABS(RI*DCOS(ALFA2-3.14159))
          XCB=ABS(RO*DCOS(ALFA2-3.14159))
          XCC=ABS(RO*DCOS(ALFA1-3.14159))
          XCD=ABS(RI*DCOS(ALFA1-3.14159))
          YCA=ABS(RI*DSIN(ALFA2-3.14159))
          YCB=ABS(RO*DSIN(ALFA2-3.14159))
          YCC=ABS(RO*DSIN(ALFA1-3.14159))
          YCD=ABS(RI*DSIN(ALFA1-3.14159))
        ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN !FOURTH QUADRANT
          XCD=ABS(RI*DCOS(2.0*3.14159-ALFA2))
          XCC=ABS(RO*DCOS(2.0*3.14159-ALFA2))
          XCB=ABS(RO*DCOS(2.0*3.14159-ALFA1))
          XCA=ABS(RI*DCOS(2.0*3.14159-ALFA1))
          YCD=ABS(RI*DSIN(2.0*3.14159-ALFA2))
          YCC=ABS(RO*DSIN(2.0*3.14159-ALFA2))
          YCB=ABS(RO*DSIN(2.0*3.14159-ALFA1))
          YCA=ABS(RI*DSIN(2.0*3.14159-ALFA1))
        ENDIF

cccccc  RESET AREAS  ccccccccccccccccccccccccccccccccccccccccccccccccccc
        DO INDHV=1,2
         DO II=1,4
          DO I=1,30
           DO J=1,30
            IF(AM(INDHV,II,I,J).NE.0.0)AM(INDHV,II,I,J)=0.0
           ENDDO
          ENDDO
         ENDDO
        ENDDO

        DO I=1,NIS      ! CALCULATE ALL QUADRANTS FOR ALL POINTS
          DO IQ=1,4
            IF(XL(I).LT.0.000001)THEN            !VERTICAL LINE CROSSING
             X=ABS(XC(IZ,ISY(I),ISX(I))-XWELL)
             Y=ABS(YL(I)-YWELL)
            ELSEIF(YL(I).LT.0.000001)THEN        !HORIZONTAL LINE CROSSING
             X=ABS(XL(I)-XWELL)
             Y=ABS(YC(IZ,ISY(I),ISX(I))-YWELL)
            ENDIF

            IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN            ! FIRST QUADRANT
              IF(IQ.EQ.1)IQT=1
              IF(IQ.EQ.2)IQT=2
              IF(IQ.EQ.3)IQT=3
              IF(IQ.EQ.4)IQT=4
            ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN   !SECOND QUADRANT
              IF(IQ.EQ.1)IQT=4
              IF(IQ.EQ.2)IQT=3
              IF(IQ.EQ.3)IQT=2
              IF(IQ.EQ.4)IQT=1
            ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN   !THIRD QUADRANT
              IF(IQ.EQ.1)IQT=3
              IF(IQ.EQ.2)IQT=4
              IF(IQ.EQ.3)IQT=1
              IF(IQ.EQ.4)IQT=2
            ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN  !FOURTH QUADRANT
              IF(IQ.EQ.1)IQT=2
              IF(IQ.EQ.2)IQT=1
              IF(IQ.EQ.3)IQT=4
              IF(IQ.EQ.4)IQT=3
```

```
          ENDIF
          IF(ABS(YL(I)).LT.0.000001)THEN    ! HORIZONTAL LINE CROSSING
            INDHV=1
          ELSEIF(ABS(XL(I)).LT.0.000001)THEN    ! VERTICAL LINE CROSSING
            INDHV=2
          ELSE
            WRITE(*,*)'ERROR 1'
            STOP
          ENDIF

          AM(INDHV,IQT,ISY(I),ISX(I))=AREA(IQ,RI,RO,XCA,XCB,XCC,XCD,
     &                                     YCA,YCB,YCC,YCD,X,Y)
C          WRITE(*,*)I,IQT,AM(INDHV,IQT,ISY(I),ISX(I))
        ENDDO
      ENDDO

      IADD=0

      DO I=1,NIS    ! CONSTRUCT ARRAY WITH AREA CONTRIBUTIONS
C        WRITE(*,*)'XL(I),YL(I)',XL(I),YL(I)
        IF(ABS(YL(I)).LT.0.000001)THEN    ! HORIZONTAL LINE CROSSING
          IADD=IADD+1
          A(IADD)=AM(1,1,ISY(I),ISX(I))+AM(1,4,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)
          IBY(IADD)=ISY(I)
          IHV(IADD)=1
C          WRITE(*,*)'HORISONTAL ADDING - 1, I=',I,ISX(I),ISY(I),
C     &     AM(1,1,ISY(I),ISX(I)),AM(1,4,ISY(I),ISX(I))

          IADD=IADD+1
          A(IADD)=AM(1,2,ISY(I),ISX(I))+AM(1,3,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)
          IBY(IADD)=ISY(I)+1
          IHV(IADD)=1               !HORIZONTAL DIVISION
C          WRITE(*,*)'HORISONTAL ADDING - 2, I=',I,ISX(I),ISY(I),
C     &     AM(1,2,ISY(I),ISX(I)),AM(1,3,ISY(I),ISX(I))

        ENDIF
      ENDDO

      DO I=1,NIS    ! CONSTRUCT ARRAY WITH AREA CONTRIBUTIONS
C        WRITE(*,*)'XL(I),YL(I)',XL(I),YL(I)
        IF(ABS(XL(I)).LT.0.000001)THEN    ! VERTICAL LINE CROSSING
          IADD=IADD+1
          A(IADD)=AM(2,1,ISY(I),ISX(I))+AM(2,2,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)
          IBY(IADD)=ISY(I)
          IHV(IADD)=2
C          WRITE(*,*)'VERTICAL ADDING - 1'

          IADD=IADD+1
          A(IADD)=AM(2,3,ISY(I),ISX(I))+AM(2,4,ISY(I),ISX(I))
          IBX(IADD)=ISX(I)+1
          IBY(IADD)=ISY(I)
          IHV(IADD)=2               !VERTICAL DIVISION
C          WRITE(*,*)'VERTICAL ADDING - 2'

        ENDIF
      ENDDO

      NIB=IADD

CCCCCCC  CHECK IF A BLOCK HAS BEEN MULTIPLIED CCCCCCCCCCCCCCCCCCCCCCCC
      ISTAT=0
      DO I=1,NIB-1
C        IF(ISTAT.EQ.1)EXIT
        DO II=I+1,NIB
          IF(IBX(I).EQ.IBX(II).AND.IBY(I).EQ.IBY(II))THEN    !SAME POINTS
            ISTAT=1
            IF(IHV(I).EQ.1)THEN                               !HORIZONTAL LINE CROSSING
              IF(IHV(II).EQ.2)THEN                            !VERTICAL LINE
                IF(IBY(II+1).EQ.IBY(II).AND.(IBX(II+1).EQ.IBX(II)+1
     &            .OR.IBX(II+1).EQ.IBX(II)-1))THEN
                A(I)=A(I)-A(II+1)
                A(II)=0.0
```

```
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 1'
              ELSEIF(IBY(II-1).EQ.IBY(II).AND.(IBX(II-1).EQ.IBX(II)+1
     &          .OR.IBX(II-1).EQ.IBX(II)-1))THEN
                A(I)=A(I)-A(II-1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 2'
              ENDIF
            ELSEIF(IHV(II).EQ.1)THEN                          !HORIZONTAL LINE CROSSING
              IF(IBX(II+1).EQ.IBX(II).AND.(IBY(II+1).EQ.IBY(II)+1
     &          .OR.IBY(II+1).EQ.IBY(II)-1))THEN
                A(I)=A(I)-A(II+1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 3'
              ELSEIF(IBX(II-1).EQ.IBX(II).AND.(IBY(II-1).EQ.IBY(II)+1
     &          .OR.IBY(II-1).EQ.IBY(II)-1))THEN
                A(I)=A(I)-A(II-1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 4'
              ENDIF
            ENDIF
          ELSEIF(IHV(I).EQ.2)THEN                             !VERTICAL LINE CROSSING
            IF(IHV(II).EQ.1)THEN                              !HORIZONTAL LINE
              IF(IBX(II+1).EQ.IBX(II).AND.(IBY(II+1).EQ.IBY(II)+1
     &          .OR.IBY(II+1).EQ.IBY(II)-1))THEN
                A(I)=A(I)-A(II+1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 5'
              ELSEIF(IBX(II-1).EQ.IBX(II).AND.(IBY(II-1).EQ.IBY(II)+1
     &          .OR.IBY(II-1).EQ.IBY(II)-1))THEN
                A(I)=A(I)-A(II-1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 6'
              ENDIF
            ELSEIF(IHV(II).EQ.2)THEN                                !VERTICAL LINE
              IF(IBY(II+1).EQ.IBY(II).AND.(IBX(II+1).EQ.IBX(II)+1
     &          .OR.IBX(II+1).EQ.IBX(II)-1))THEN
                A(I)=A(I)-A(II+1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 7'
              ELSEIF(IBY(II-1).EQ.IBY(II).AND.(IBX(II-1).EQ.IBX(II)+1
     &          .OR.IBX(II-1).EQ.IBX(II)-1))THEN
                A(I)=A(I)-A(II-1)
                A(II)=0.0
                IBX(II)=0
                IBY(II)=0
C                WRITE(*,*)'ALT 8'
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDDO
    ENDDO

CCCCCCC  REORGANIZE THE ARRAYS CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    ISUB=0
    DO I=1,NIB
      IF(IBX(I).EQ.0.AND.IBY(I).EQ.0)ISUB=ISUB+1
    ENDDO

    DO J=1,ISUB
```

```
  ISTAT=0
  DO I=1,NIB-J+1
    IF(I.EQ.NIB-J+1)THEN
      ISTAT=1
      A(I)=0.
      IBX(I)=0
      IBY(I)=0
    ELSEIF((IBX(I).EQ.0.AND.IBY(I).EQ.0).OR.ISTAT.EQ.1)THEN
      ISTAT=1
      A(I)=A(I+1)
      IBX(I)=IBX(I+1)
      IBY(I)=IBY(I+1)
    ENDIF
  ENDDO
ENDDO
NIB=NIB-ISUB
END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c          ROUTINE TO FIND BLOCK THAT TOTALLY SURROUNDS A RADIAL SECTOR
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE FINDBL(IZ,XWELL,YWELL,RI,RO,ALFA1,ALFA2,A,IBX,IBY,NIB)

      IMPLICIT NONE

cccccc COMMON VARIABLES ccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XC(70,20,20)       ! CARTESIAN X COORDINATES
      DOUBLE PRECISION YC(70,20,20)       ! CARTESIAN Y COORDINATES
      DOUBLE PRECISION ZC(70,20,20)       ! CARTESIAN Z COORDINATES
      DOUBLE PRECISION KXC(70,20,20)      ! CARTESIAN X PERMS
      DOUBLE PRECISION KYC(70,20,20)      ! CARTESIAN Y PERMS
      DOUBLE PRECISION KZC(70,20,20)      ! CARTESIAN Z PERMS

      INTEGER NLX  ! NUMBER OF LAYERS IN X DIRECTION
      INTEGER NLY  ! NUMBER OF LAYERS IN Y DIRECTION
      INTEGER NLZ  ! NUMBER OF LAYERS IN Z DIRECTION

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DOUBLE PRECISION XWELL        ! X-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION YWELL        ! Y-DISTANCE FROM WELL TO ORIGO
      DOUBLE PRECISION ALFA1        ! ANGLE TO FIRST (LEFT) GRID LINE
      DOUBLE PRECISION ALFA2        ! ANGLE TO SECOND GRID LINE
      DOUBLE PRECISION ALFA1N       ! ANGLE TO FIRST (LEFT) GRID LINE - NORM.
      DOUBLE PRECISION ALFA2N       ! ANGLE TO SECOND GRID LINE       - NORM.
      DOUBLE PRECISION RI           ! INNER RADIUS OF SEGMENT
      DOUBLE PRECISION RO           ! OUTER RADIUS OF SEGMENT
      DOUBLE PRECISION XCA          ! LEFT  BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCB          ! LEFT  UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCC          ! RIGHT UPPER  CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION XCD          ! RIGHT BOTTOM CORNER OF SEG. X-DIST. FROM WELL
      DOUBLE PRECISION YCA          ! LEFT  BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCB          ! LEFT  UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCC          ! RIGHT UPPER  CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION YCD          ! RIGHT BOTTOM CORNER OF SEG. Y-DIST. FROM WELL
      DOUBLE PRECISION A(*)         ! AREA OF SEGMENTS

      INTEGER NIB                  ! NUMBER OF BLOCKS ENCLOSED IN SEG.
      INTEGER IZ                   ! Z - LAYER IN QUESTION
      INTEGER IBX(*)               ! ARRAY FOR K - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER IBY(*)               ! ARRAY FOR J - POSITIONS OF CONTRIBUTING BLOCKS
      INTEGER I,J,K                ! COUNTERS

      COMMON /CARTDAT/ XC,YC,ZC,KXC,KYC,KZC,NLX,NLY,NLZ

cccccc CALCULATE CORNER POSITIONS FOR SEGMENT ccccccccccccccccccccccccc
cccccc TAKING INTO ACCOUNT THE QUADRANT IT IS LOCATED IN ccccccccccccccccc

      IF(ALFA1.GE.0.0.AND.ALFA2.LE.3.14159/2.)THEN      !FIRST QUADRANT
        ALFA1N=ALFA1
        ALFA2N=ALFA2
        XCA=XWELL+ABS(RI*DCOS(ALFA2N))
        XCB=XWELL+ABS(RO*DCOS(ALFA2N))
        XCC=XWELL+ABS(RO*DCOS(ALFA1N))
        XCD=XWELL+ABS(RI*DCOS(ALFA1N))
        YCA=YWELL+ABS(RI*DSIN(ALFA2N))
        YCB=YWELL+ABS(RO*DSIN(ALFA2N))
        YCC=YWELL+ABS(RO*DSIN(ALFA1N))
        YCD=YWELL+ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3.14159/2.0.AND.ALFA2.LE.3.14159)THEN ! SECOND QUADRANT
        ALFA1N=3.14159-ALFA1
        ALFA2N=3.14159-ALFA2
        XCD=XWELL-ABS(RI*DCOS(ALFA2N))
        XCA=XWELL-ABS(RO*DCOS(ALFA2N))
        XCB=XWELL-ABS(RO*DCOS(ALFA1N))
        XCC=XWELL-ABS(RI*DCOS(ALFA1N))
        YCD=YWELL+ABS(RI*DSIN(ALFA2N))
```

```
        YCA=YWELL+ABS(RO*DSIN(ALFA2N))
        YCB=YWELL+ABS(RO*DSIN(ALFA1N))
        YCC=YWELL+ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3.14159.AND.ALFA2.LE.3./2.*3.14159)THEN ! THIRD QUADRANT
        ALFA1N=ALFA1-3.14159
        ALFA2N=ALFA2-3.14159
        XCC=XWELL-ABS(RI*DCOS(ALFA2N))
        XCD=XWELL-ABS(RO*DCOS(ALFA2N))
        XCA=XWELL-ABS(RO*DCOS(ALFA1N))
        XCB=XWELL-ABS(RI*DCOS(ALFA1N))
        YCC=YWELL-ABS(RI*DSIN(ALFA2N))
        YCD=YWELL-ABS(RO*DSIN(ALFA2N))
        YCA=YWELL-ABS(RO*DSIN(ALFA1N))
        YCB=YWELL-ABS(RI*DSIN(ALFA1N))
      ELSEIF(ALFA1.GE.3./2.*3.14159.AND.ALFA2.LE.2.*3.14159)THEN ! FOURTH QUADRANT
        ALFA1N=2.*3.14159-ALFA1
        ALFA2N=2.*3.14159-ALFA2
        XCB=XWELL+ABS(RI*DCOS(ALFA2N))
        XCC=XWELL+ABS(RO*DCOS(ALFA2N))
        XCD=XWELL+ABS(RO*DCOS(ALFA1N))
        XCA=XWELL+ABS(RI*DCOS(ALFA1N))
        YCB=YWELL-ABS(RI*DSIN(ALFA2N))
        YCC=YWELL-ABS(RO*DSIN(ALFA2N))
        YCD=YWELL-ABS(RO*DSIN(ALFA1N))
        YCA=YWELL-ABS(RI*DSIN(ALFA1N))
      ENDIF

CCCCCC  FIND BLOCK CCCCCCCCCCCCCCCCCCCCCCCCCC

      DO I=1,IZ
        DO J=1,NLY
          DO K=1,NLX
            IF(XC(I,J,K).GT.XCC.AND.YC(I,J,K).GT.YCB)THEN
              IBX(1)=K
              IBY(1)=J
              GOTO 222
            ENDIF
          ENDDO
        ENDDO
      ENDDO

 222  A(1)=(RO**2.-RI**2.)*(ALFA2-ALFA1)/2.
      NIB=1
      RETURN
      END
```

APPENDIX C

```
-- ECLIPSE INPUT FILE FOR HETEROGENEOUS CASE - BASE CASE - FINE GRID
--
--
RUNSPEC     ============================================================
             TEST - FINE GRID
= NDIVIX NDIVIY NDIVIZ QRDIAL NUMRES QNNCON MXNAQN MXNAQC QDPORO QDPERM
    15     60     15      F      1      T      0      0      F      F /
=   OIL   WATER   GAS  DISGAS VAPOIL
     T      T      T      T      F              /
= UNIT CONVENTION
     'METRIC'                                  /
=  NRPVT  NPPVT  NTPVT
     8     13      1                           /
= NSSFUN NTSFUN QDIRKR QREVKR
    30      1      F      T                    /
= NDRXVD NTEQUL NDPRVD
     2      1    100                           /
=  NTFIP
     7                                         /
= NWMAXZ NCWMAX NGMAXZ NWGMAX MAXLGR MAXCLS
     1    60      1     60     0    13500     /
= QIMCOL NWFRIC NUPCOL
     F      0      4                           /
= MXMFLO MXMTHP MXMWFR MXMGFR MXMALQ NMMVFT
    0      0      0      0      0      0      /
= MXSFLO MXSTHP NMSVFT
     0      0      0                           /
= NAQFET NCAMAX
     0      0                                  /
=   DAY   MONTH  YEAR
     1    'JAN'  1996                         /
= QSOLVE NSTACK
     T     25   /

GRID        ============================================================
INCLUDE
 'BASE.FILLED'
/
INCLUDE
'px1'
/
INCLUDE
'py1'
/
INCLUDE
'pz1'
/
EQUALS
 'PORO'   0.2 /
/
PROPS       ============================================================
INCLUDE
 'test.pvt'
/
--/PERM
SOLUTION    ============================================================
EQUIL
 1501.  150.00   1700.  0   1501.   0   0   0   0  /
RPTSOL
1 1 1 0 0 0 2 2 1 0
/
SUMMARY     ============================================================
BPR
1 30 8 /
/
CPR
'PROD1' 8 1 8/
```

```
/
FGOR
/
FWCT
/
COFR
'PROD1' 8 1 8 /
'PROD1' 8 2 8 /
'PROD1' 8 3 8 /
'PROD1' 8 4 8 /
'PROD1' 8 5 8 /
'PROD1' 8 6 8 /
'PROD1' 8 7 8 /
'PROD1' 8 8 8 /
'PROD1' 8 9 8 /
'PROD1' 8 10 8 /
'PROD1' 8 11 8 /
'PROD1' 8 12 8 /
'PROD1' 8 13 8 /
'PROD1' 8 14 8 /
'PROD1' 8 15 8 /
'PROD1' 8 16 8 /
'PROD1' 8 17 8 /
'PROD1' 8 18 8 /
'PROD1' 8 19 8 /
'PROD1' 8 20 8 /
'PROD1' 8 21 8 /
'PROD1' 8 22 8 /
'PROD1' 8 23 8 /
'PROD1' 8 24 8 /
'PROD1' 8 25 8 /
'PROD1' 8 26 8 /
'PROD1' 8 27 8 /
'PROD1' 8 28 8 /
'PROD1' 8 29 8 /
'PROD1' 8 30 8 /
'PROD1' 8 31 8 /
'PROD1' 8 32 8 /
'PROD1' 8 33 8 /
'PROD1' 8 34 8 /
'PROD1' 8 35 8 /
'PROD1' 8 36 8 /
'PROD1' 8 37 8 /
'PROD1' 8 38 8 /
'PROD1' 8 39 8 /
'PROD1' 8 40 8 /
'PROD1' 8 41 8 /
'PROD1' 8 42 8 /
'PROD1' 8 43 8 /
'PROD1' 8 44 8 /
'PROD1' 8 45 8 /
'PROD1' 8 46 8 /
'PROD1' 8 47 8 /
'PROD1' 8 48 8 /
'PROD1' 8 49 8 /
'PROD1' 8 50 8 /
'PROD1' 8 51 8 /
'PROD1' 8 52 8 /
'PROD1' 8 53 8 /
'PROD1' 8 54 8 /
'PROD1' 8 55 8 /
'PROD1' 8 56 8 /
'PROD1' 8 57 8 /
'PROD1' 8 58 8 /
'PROD1' 8 59 8 /
'PROD1' 8 60 8 /
/
BPR
2   1  8 /
2   2  8 /
2   3  8 /
2   4  8 /
2   5  8 /
2   6  8 /
2   7  8 /
2   8  8 /
2   9  8 /
```

```
2  10  8 /
2  11  8 /
2  12  8 /
2  13  8 /
2  14  8 /
2  15  8 /
2  16  8 /
2  17  8 /
2  18  8 /
2  19  8 /
2  20  8 /
2  21  8 /
2  22  8 /
2  23  8 /
2  24  8 /
2  25  8 /
2  26  8 /
2  27  8 /
2  28  8 /
2  29  8 /
2  30  8 /
2  31  8 /
2  32  8 /
2  33  8 /
2  34  8 /
2  35  8 /
2  36  8 /
2  37  8 /
2  38  8 /
2  39  8 /
2  40  8 /
2  41  8 /
2  42  8 /
2  43  8 /
2  44  8 /
2  45  8 /
2  46  8 /
2  47  8 /
2  48  8 /
2  49  8 /
2  50  8 /
2  51  8 /
2  52  8 /
2  53  8 /
2  54  8 /
2  55  8 /
2  56  8 /
2  57  8 /
2  58  8 /
2  59  8 /
2  60  8 /
/
RUNSUM
SEPARATE
SCHEDULE   ============================================================
RPTSCHED
 0  0  0  0  0  0  2  2  2  0
 1  2  0  0  2
/
WELSPECS
'PROD1'    'PROD'  8    1   1601.5  'OIL'  0.0 'STD' 'SHUT' 'YES' 0 'SEG' 1/
/
COMPDAT
'PROD1'    8   1   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   2   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   3   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   4   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   5   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   6   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   7   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   8   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   9   8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   10  8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   11  8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   12  8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   13  8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
'PROD1'    8   14  8  8  'OPEN'   0  .000000  0.20  3*0.0 'Y'/
```

```
'PROD1'   8   15   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   16   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   17   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   18   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   19   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   20   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   21   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   22   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   23   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   24   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   25   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   26   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   27   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   28   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   29   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   30   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   31   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   32   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   33   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   34   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   35   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   36   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   37   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   38   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   39   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   40   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   41   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   42   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   43   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   44   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   45   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   46   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   47   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   48   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   49   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   50   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   51   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   52   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   53   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   54   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   55   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   56   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   57   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   58   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   59   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
'PROD1'   8   60   8   8   'OPEN'   0   .000000   0.20   3*0.0 'Y'/
/
WCONPROD
  'PROD1'   'OPEN'   'ORAT' 100.   /
/
TUNING
-- TSINIT   TSMAXZ   TSMINZ   TSMCHP   TSFMAX   TSFMIN   TSFCNV   TFDIFF
     0.001     25.   0.0001     0.15      3.0      0.3      0.1     1.25          /
-- TRGTTE   TRGCNV   TRGMBE   TRGLCV   XXXTTE   XXXCNV   XXXMBE   XXXLCV   XXXWFL
      0.1    0.001   1.0E-7   0.0001     10.0     0.01   1.0E-6    0.001    0.001   /
-- NEWTMX   NEWTMN   LITMAX   LITMIN   MXWSIT   MXWPIT   DDPLIM   DDSLIM
       12        1       25        1        8        8    1.0E6    1.0E6          /
TSTEP
 10*.001
 10*.01
 10*.1
 10*1
/
END
```

GRID SPECIFICATION FILE

```
SPECGRID
 15 60 15 1 F /

DXV
 1*100 13*1 1*100/

DYV
  60*5 /

TOPS
 900*1500 12600*/

DZ
 900*100 11700*1 900*100/

FILL
```

PVT PROPERTIES FILE

```
--   Rs        Pb      Bo    VISCo
PVTO
  13.62     40.00   1.060   3.311   /
  20.44     60.00   1.076   2.848   /
  27.41     80.00   1.091   2.504   /
  34.51    100.00   1.107   2.242   /
  41.77    120.00   1.123   2.037   /
  49.17    140.00   1.139   1.873   /
  56.36    159.06   1.154   1.747
           215.57   1.146   1.883   /
/

--  Pg      Bg  VISCg

PVDG
  40.0   0.026 0.0128
  60.0   0.018 0.0136
  80.0   0.013 0.0143
 100.0   0.010 0.0151
 120.0   0.008 0.0160
 140.0   0.007 0.0170
 160.0   0.006 0.0179
 200.0   0.005 0.0199
/

--  Pref       Bw            Cw          VISCw
PVTW
 158.20000   1.017       4.30E-5       0.45       0.00E+00 /


--  RHOo      RHOw      Rhog
DENSITY
    890.0  1045.0      0.75
/

ROCK
 158.2         2.85E-4 /

SWFN
   .288   .000000   .00000000
   .338   .007120   .00000000
   .388   .020140   .00000000
   .438   .037000   .00000000
   .488   .056965   .00000000
   .538   .079611   .00000000
   .588   .104652   .00000000
   .638   .131877   .00000000
   .688   .161122   .00000000
   .738   .192258   .00000000
   .750   .200000   .00000000
  1.000   .300000   .00000000
/
SGFN
   .000   .000000   .00000000
   .100   .020000   .00000000
   .150   .038130   .00000000
   .200   .090690   .00000000
   .250   .150548   .00000000
   .300   .215699   .00000000
   .350   .285093   .00000000
   .400   .358066   .00000000
   .450   .434157   .00000000
   .500   .513023   .00000000
   .550   .594398   .00000000
   .600   .678070   .00000000
   .642   .750000   .00000000
   .712   .850000   .00000000
/
SOF3
   .000   .000000   .00000000
   .070   .000000   .00000000
   .120   .000000   .00003127
   .170   .000000   .00050036
```

.220	.000000	.00253305
.250	.000000	.00525253
.300	.003275	.01400200
.350	.018527	.03075466
.400	.051055	.05933825
.450	.104806	.10433101
.500	.183089	.17106174
.550	.288813	.26560974
.600	.424604	.39480489
.650	.592876	.56622767
.700	.795876	.78820878
.712	.850000	.85000002

/