THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

OPTIMAL NUMERICAL PROCEDURE TO SOLVE TWO-DIMENSIONAL THREE-PHASE PETROLEUM RESERVOIR SIMULATOR

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

By

MUHAMMAD ALI AL-MARHOUN

Norman, Oklahoma

1978

OPTIMAL NUMERICAL PROCEDURE TO SOLVE TWO-DIMENSIONAL

THREE-PHASE PETROLEUM RESERVOIR SIMULATOR

A DISSERTATION

APPROVED FOR THE DEPARTMENT OF PETROLEUM ENGINEERING

By

Henry Ridder

James A. Payne

D E Mcenzie

Adel A. Aly

A. W. McCray

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to the Chairman of his Graduate Committee, Dr. H.B. Crichlow, Director of the department of Petroleum and Geological Engineering, for his assistance in connection with this work.

Appreciation is also extended to Dr. D.E. Menzie, Dr. A.W. McCray of Petroleum Engineering Department, Dr. J.A. Payne of Computing Sciences Department, and Dr. A.A. Aly of Industrial Engineering Department for their revision of this dissertation and for serving as members of the author's Graduate Committee.

Particular thanks are due to the University of Petroleum and Minerals of Dhahran, Saudi Arabia, for their financial support.

The author also wishes to make a special recognition of his parents for their constant support and inspiration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Mathematical models for petroleum reservoirs incorporate the reservoir physical properties and the interaction of natural and artificial forces to simulate reservoir behavior. Therefore, mathematical simulation helps in understanding reservoir behavior. Such information leads to the most economically desirable form of exploitation.

Reservoir simulators are used to design the most economical secondary recovery program and to lay out the complete reservoir management from discovery to depletion.

Natural reservoirs consist of one, two or three phases (oil, water, and gas) with various geometries. Therefore, various reservoir simulators were developed. They start with one-dimensional one-phase simple simulators and end up with complex three-dimensional three-phase ones. Although three-dimensional simulators are now available, the most popular simulator is the two-dimensional one because a well-organized two-dimensional simulator can be used to approximate three-dimensional one at a lower cost.

The efficiency of a simulator as a working and economical tool hinges upon the ability of its algorithms to solve the pressure equations efficiently.[6] Most of the computer time is spent in performing pressure

solutions. For this reason, the best simulator is the one which requires the minimum work during the pressure solutions.

The pressure equations can be solved either by iterative or direct methods.

Iterative methods are by far the most common methods of solving the pressure equations used in simulation. This is because iterative methods are easy to program and require less storage and less computation compared to existing direct method algorithms. Hence during the early years of work in this discipline, the direct solution techniques were almost completely abandoned as a solution process. However, in recent years with the development of sparse matrix techniques, the direct solution process has become less costly, while producing as good a result. For this reason, direct solution techniques are once again being considered in reservoir simulation areas.

The existing numerical methods (direct or iterative) to solve the mathematical models of petroleum reservoirs are time consuming. Those methods consume different computer time when applied on reservoirs with different characteristics.

The computer time is expensive, therefore, a tool to choose the fastest, most efficient methods for a given reservoir is needed. A comparative study helps to show the features of each method. Such information leads to the choice of the most economically suitable method.

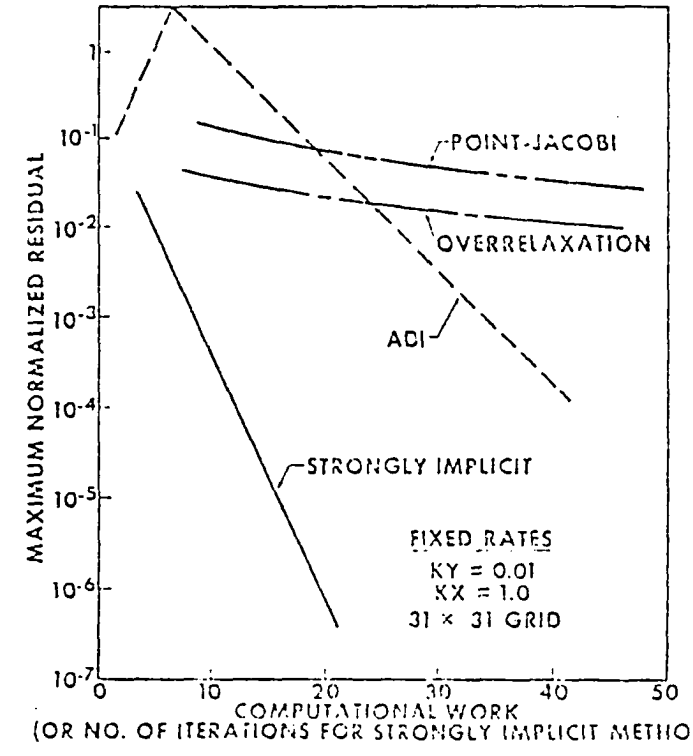
The comparative studies done in literature are limited to two or three methods and most of these comparative evaluations are done on a system of simulaneous equations not as part of a complete simulation. Also most of these comparative studies used idealized model of fixed rates and square reservoir with pre-determined flow coefficients.

Stone[14] has presented a comparison among the following numerical procedures: alternating direction implicit (ADI), strongly implicit (SIP), point successive overrelaxation (PSOR), and point Jacobi. His study was in terms of computational work and residual. He used an idealized square reservoir with fixed flow rates and a combination of homogeneous regions. His results are presented in Figure 1.1. Stone concluded that both ADI and SIP possess convergence rates a great deal faster than those of the other two methods. For homogeneous case, ADI is slightly faster than SIP. However, SIP is significantly faster than ADI in case of heterogeneous reservoirs. Also he concluded that PSOR is more rapid than point Jacobi method.

Breitenbach, et al.[3] gave a very brief comparison between Gauss elimination, PSOR, and ADI. For his model, he showed that PSOR and ADI are more efficient than Gauss elimination for more than 9x26 cell system. His results are shown in Figure 1.2.

Watts[19] has presented a comparison among ADI, SIP, line successive overrelaxation (LSOR), and corrected LSOR in terms of accuracy. His results are shown in Figure 1.3. He concluded that for two-dimensional strongly anisotropic problems, the corrected LSOR is faster than other available techniques. However, in homogeneous isotropic square problem, the corrected LSOR is slower than ADI, SIP, and LSOR.

Weinstein, et al.[20] have compared SIP with ADI for a two-dimensional two-phase model. Their results are shown in Figure 1.4. They found that the computational work ratio, which is the ratio of ADI computer time to SIP computer time, generally increases with increasing transmissibility (ρ_{max}). They also found that ADI fails to converge for higher transmissibility.



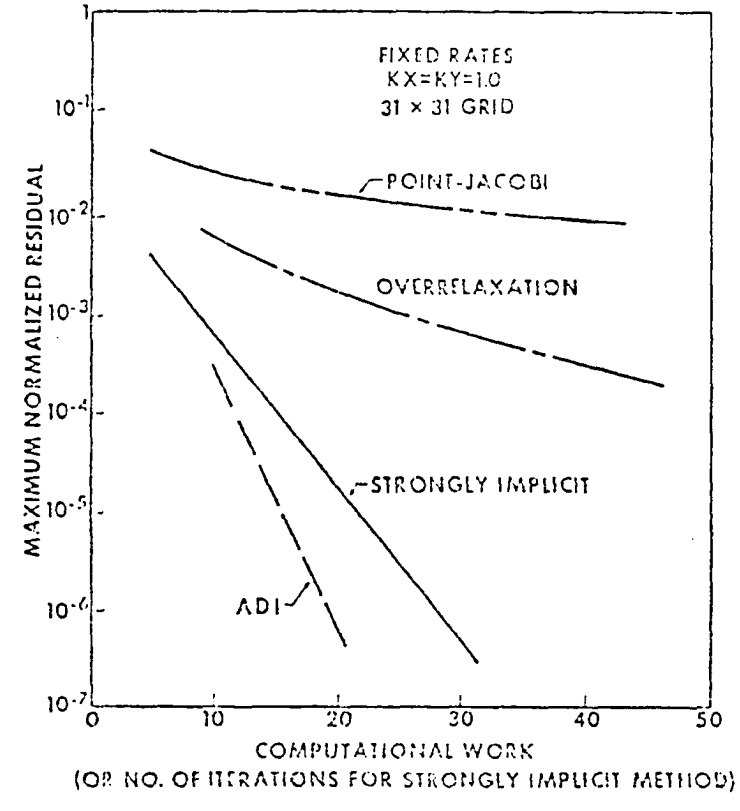


Figure 1.1 - Stone's Comparative Study

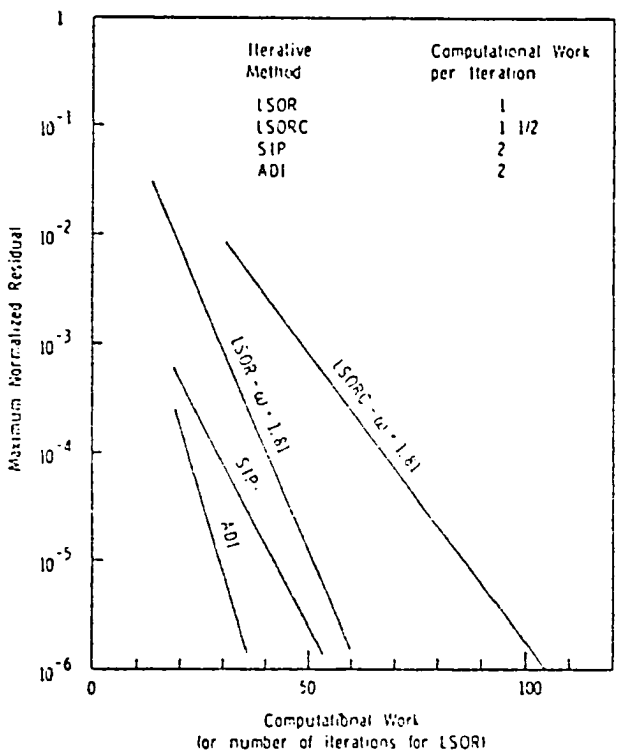
Grid	Method		
	Gauss	SOR	ADIPIT
3 x 5	2***	5	5
3 x 5 x 2	5	12 **	-
9 x 10	8	11	-
13 x 11	18	25	19
9 x 26 *	21	40 **	-
9 x 26	21	21	22
26 x 23	94	59	62
34 x 36	302	122	111
20 x 24 x 2	564	110	-

* Vertical cross section

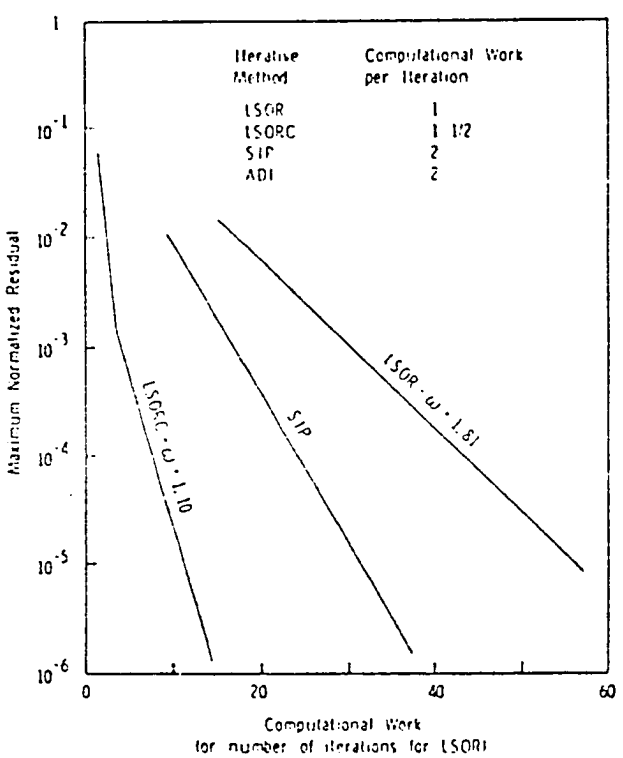
** Line SOR

*** Time in second

Figure 1.2 - Breitenbach, et al.'s Comparative Study



A — CONVERGENCE COMPARISONS, HOMOGENEOUS SQUARE 31 × 31 GRID, $k_x = k_y$.



B — CONVERGENCE COMPARISONS, HETEROGENEOUS SQUARE 31 × 31 GRID, $k_x = 100\, k_y$.

Figure 1.3 - Watts' Comparative Study

Problem	ρ_{max}	Iteration Procedure	Parameter (Max SIP / Min ADIP)	Number of Time Steps	Number of Iterations	Average Iterations/ Time Step	Iteration Time (millisec/ Iterations/ Block)	Total Time (minutes)	Work Ratio
Dissolved Gas Drive Case I*	66	SIP	0.999942, 1	8	189	23.6	2.41	3.83	
		ADIP	0.569×10^{-4}, 0	8	247	30.9	2.26	4.69	1.22
Dissolved Gas Drive Case II*	660	SIP	0.9999942, 1	8	193	24.2	2.40	3.89	
		ADIP	0.231×10^{-4}, 0		∞				∞
Gas-Oil Incompressible	3,600	SIP	0.999998	25	77	3.1	1.97	0.993	
		ADIP	0.203×10^{-4}	25	389	15.6	1.48	3.75	3.78
Water-Oil Radial Coning	7.05×10^{5}	SIP	0.9995	9**	86	9.6	1.90	0.65	
		ADIP	0.001	9**	∞				∞

*All K_y of Case II are 10 times those of Case I.

**Simulation from 30.4 to 365 days. (The 0–30.4 day time step was not included, since both SIP and ADIP failed to converge.)

Figure 1.4-Weinstein, et al.'s Comparative Study

Price and Coats[12] have presented a comparison among four different ordering schemes outlined in chapter V. A direct solution method is used with these ordering schemes to solve idealized reservoir problems. The Price and Coats study produces impressive results in terms of computational work reduction. However, these results in computational work reduction are limited to the computer time involved in elimination portion of non-zero elements of the equations system and do not include the additional work required to reorder the cell system from standard ordering to the appropriate ordering.

From the afore-mentioned survey, a development of a better direct method approach is needed. A more comprehensive comparative evaluation using a realistic petroleum reservoir is also needed. This study will address these two areas.

CHAPTER II

RESERVOIR MODEL

The introduction of Darcy's law into the two dimensional continuity equation for each of the three immiscible fluid phases in petroleum reservoirs leads to the following system of partial differential equations :

$$\frac{\partial}{\partial x}\left(h\rho_o\lambda_o\frac{\partial\Phi_o}{\partial x}\right)+\frac{\partial}{\partial y}\left(h\rho_o\lambda_o\frac{\partial\Phi_o}{\partial y}\right)+\frac{q_o\rho_o^*}{A}=h\frac{\partial}{\partial t}\left(\phi\rho_o S_o\right) \qquad (2.1)^1$$

for the oil phase,

$$\frac{\partial}{\partial x}\left(h\rho_w\lambda_w\frac{\partial\Phi_w}{\partial x}\right)+\frac{\partial}{\partial y}\left(h\rho_w\lambda_w\frac{\partial\Phi_w}{\partial y}\right)+\frac{q_w\rho_w^*}{A}=h\frac{\partial}{\partial t}\left(\phi\rho_w S_w\right) \qquad (2.2)^1$$

for the water phase, and

$$\begin{aligned}&\frac{\partial}{\partial x}\left(h\rho_g\lambda_g\frac{\partial\Phi_g}{\partial x}+R_{so}h\rho_o\lambda_o\frac{\partial\Phi_o}{\partial x}+R_{sw}h\rho_w\lambda_w\frac{\partial\Phi_w}{\partial x}\right)\\&+\frac{\partial}{\partial y}\left(h\rho_g\lambda_g\frac{\partial\Phi_g}{\partial y}+R_{so}h\rho_o\lambda_o\frac{\partial\Phi_o}{\partial y}+R_{sw}h\rho_w\lambda_w\frac{\partial\Phi_w}{\partial y}\right)\\&+\frac{q_g\rho_g^*}{A}=h\frac{\partial}{\partial t}\left(\phi\rho_g S_g+R_{so}\phi\rho_o S_o+R_{sw}\phi\rho_w S_w\right)\end{aligned} \qquad (2.3)$$

for the gas phase.

where

$$\lambda_j = k \frac{k_{rj}}{\mu_j} \qquad , j = o, g, w \qquad (2.4)$$

These equations have been derived making the following assumptions:

1) Darcy's law applies

2) Composition of each fluid phase is constant

The generalized multiphase flow equation for the unsteady-state flow of oil, gas, and water in a porous medium is developed by combining the three single-phase equations into one basic equation. In order to do this, we need to express the following auxilury equations:

The potential terms are

$$\Phi_o = P_o + \rho_o g D \qquad (2.5)$$

$$\Phi_g = P_g + \rho_g g D \qquad (2.6)$$

$$\Phi_w = P_w + \rho_w g D \qquad (2.7)$$

the capillary pressure terms are

$$P_{cw} = P_o - P_w \qquad (2.8)$$

$$P_{cg} = P_g - P_o \qquad (2.9)$$

and the saturation equation is

$$S_o + S_g + S_w = 1 \qquad (2.10)$$

Thus:

$$\frac{\partial S_o}{\partial t} + \frac{\partial S_g}{\partial t} + \frac{\partial S_w}{\partial t} = 0 \qquad (2.11)$$

The derivation of the generalized multiphase equation is as follows:

Divide the oil equation (2.1) by ρ_o and expand by differentiation:

$$\frac{\partial}{\partial x}\left(h\lambda_o \frac{\partial P_o}{\partial x}\right) + \frac{\partial}{\partial x}\left\{h\lambda_o \frac{\partial(\rho_o g D)}{\partial x}\right\}$$

$$+ \frac{\partial}{\partial y}\left(h\lambda_o \frac{\partial P_o}{\partial y}\right) + \frac{\partial}{\partial y}\left\{h\lambda_o \frac{\partial(\rho_o g D)}{\partial y}\right\}$$

$$= \frac{h}{\rho_o}\left(\phi\rho_o \frac{\partial S_o}{\partial t} + \phi S_o \frac{\partial \rho_o}{\partial P_o}\frac{\partial P_o}{\partial t} + \rho_o S_o \frac{\partial \phi}{\partial P_o}\frac{\partial P_o}{\partial t}\right) - \frac{q_o \rho_o^*}{A\rho_o}$$

$$- \left\{\frac{(h\lambda_o)_x}{\rho_o}\left(\frac{\partial \Phi_o}{\partial x}\right)^2 \frac{\partial \rho_o}{\partial \Phi_o} + \frac{(h\lambda_o)_y}{\rho_o}\left(\frac{\partial \Phi_o}{\partial y}\right)^2 \frac{\partial \rho_o}{\partial \Phi_o}\right\} \qquad (2.12)$$

The water equation (2.2) is divided by ρ_w and expanded by differentiation:

$$\frac{\partial}{\partial x}\left(h\lambda_w \frac{\partial P_o}{\partial x}\right) - \frac{\partial}{\partial x}\left(h\lambda_w \frac{\partial P_{cw}}{\partial x}\right) + \frac{\partial}{\partial x}\left\{h\lambda_w \frac{\partial(\rho_w g D)}{\partial x}\right\}$$

$$+ \frac{\partial}{\partial y}\left(h\lambda_w \frac{\partial P_o}{\partial y}\right) - \frac{\partial}{\partial y}\left(h\lambda_w \frac{\partial P_{cw}}{\partial y}\right) + \frac{\partial}{\partial y}\left\{h\lambda_w \frac{\partial(\rho_w g D)}{\partial y}\right\}$$

$$= \frac{h}{\rho_w}\left(\phi\rho_w \frac{\partial S_w}{\partial t} + \phi S_w \frac{\partial \rho_w}{\partial P_o}\frac{\partial P_o}{\partial t} + \rho_w S_w \frac{\partial \phi}{\partial P_o}\frac{\partial P_o}{\partial t}\right) - \frac{q_w \rho_w^*}{A\rho_w}$$

$$- \left\{\frac{(h\lambda_w)_x}{\rho_w}\left(\frac{\partial \Phi_w}{\partial x}\right)^2 \frac{\partial \rho_w}{\partial \Phi_w} + \frac{(h\lambda_w)_y}{\rho_w}\left(\frac{\partial \Phi_w}{\partial y}\right)^2 \frac{\partial \rho_w}{\partial \Phi_w}\right\} \qquad (2.13)$$

The gas equation (2.3) is divided by ρ_g and expanded by differentiation:

$$\frac{\partial}{\partial x}\left(h\lambda_g \frac{\partial P_o}{\partial x}\right) + \frac{\partial}{\partial x}\left\{h\lambda_g \frac{\partial(\rho_g g D)}{\partial x}\right\} + \frac{\partial}{\partial x}\left(h\lambda_g \frac{\partial P_{cg}}{\partial x}\right)$$

$$+ \frac{\partial}{\partial y}\left(h\lambda_g \frac{\partial P_o}{\partial y}\right) + \frac{\partial}{\partial y}\left\{ h\lambda_g \frac{\partial(\rho_g g D)}{\partial y} \right\} + \frac{\partial}{\partial y}\left(h\lambda_g \frac{\partial P_{cg}}{\partial y}\right)$$

$$= \frac{h}{\rho_g}\left(\phi\rho_g \frac{\partial S_g}{\partial t} + \phi S_g \frac{\partial \rho_g}{\partial P_o}\frac{\partial P_o}{\partial t} + \rho_g S_g \frac{\partial \phi}{\partial P_o}\frac{\partial P_o}{\partial t}\right)$$

$$+ \frac{h\phi}{\rho_g}\left(S_o\rho_o \frac{\partial R_{so}}{\partial P_o} + S_w\rho_w \frac{\partial R_{sw}}{\partial P_o}\right)\frac{\partial P_o}{\partial t} + s$$

$$-\left\{ \frac{(h\lambda_g)_x}{\rho_g}\left(\frac{\partial \Phi_g}{\partial x}\right)^2 \frac{\partial \rho_g}{\partial \Phi_g} + \frac{(h\rho_o\lambda_o)_x}{\rho_g}\left(\frac{\partial \Phi_o}{\partial x}\right)^2 \frac{\partial R_{so}}{\partial \Phi_o} + \frac{(h\rho_w\lambda_w)_x}{\rho_g}\left(\frac{\partial \Phi_w}{\partial x}\right)^2 \frac{\partial R_{sw}}{\partial \Phi_w}\right\}$$

$$-\left\{ \frac{(h\lambda_g)_y}{\rho_g}\left(\frac{\partial \Phi_g}{\partial y}\right)^2 \frac{\partial \rho_g}{\partial \Phi_g} + \frac{(h\rho_o\lambda_o)_y}{\rho_g}\left(\frac{\partial \Phi_o}{\partial y}\right)^2 \frac{\partial R_{so}}{\partial \Phi_o} + \frac{(h\rho_w\lambda_w)_y}{\rho_g}\left(\frac{\partial \Phi_w}{\partial y}\right)^2 \frac{\partial R_{sw}}{\partial \Phi_w}\right\} \quad (2.14)$$

where

$$s = -\frac{q_g\rho_g^*}{A\rho_g} + \frac{R_{so}}{\rho_g}\left\{ h\frac{\partial}{\partial t}(\phi\rho_o S_o) - \frac{\partial}{\partial x}\left(h\rho_o\lambda_o \frac{\partial \Phi_o}{\partial x}\right) - \frac{\partial}{\partial y}\left(h\rho_o\lambda_o \frac{\partial \Phi_o}{\partial y}\right)\right\}$$

$$+ \frac{R_{sw}}{\rho_g}\left\{h\frac{\partial}{\partial t}(\phi\rho_w S_w) - \frac{\partial}{\partial x}\left(h\rho_w\lambda_w \frac{\partial \Phi_w}{\partial x}\right) - \frac{\partial}{\partial y}\left(h\rho_w\lambda_w \frac{\partial \Phi_w}{\partial y}\right)\right\} \quad (2.15)$$

Incorporating equations (2.1) and (2.2) into (2.15), we obtain

$$s = -\left(q_g\rho_g^* - R_{so}q_o\rho_o^* - R_{sw}q_w\rho_w^*\right) / (\rho_g A) \quad (2.16)$$

Combining equations (2.10), (2.11), (2.12), (2.15) and (2.16), we obtain

$$\frac{\partial}{\partial x}\left(h\lambda_t \frac{\partial P_o}{\partial x}\right) + \frac{\partial}{\partial y}\left(h\lambda_t \frac{\partial P_o}{\partial y}\right) = B_1 \frac{\partial P_o}{\partial t} + B_2 + B_3 + B_4 + B_5 \quad (2.17)$$

Equation (2.17) has been derived using unified SI units, but if practical units (Appendix A) are used, equation (2.17) becomes

$$\frac{\partial}{\partial x}\left(h\lambda_t \frac{\partial P_o}{\partial x}\right) + \frac{\partial}{\partial y}\left(h\lambda_t \frac{\partial P_o}{\partial y}\right) =$$

$$11.57407408\, B_1 \frac{\partial P_o}{\partial t} + 11.57407408\, B_2 + B_3 + 0.001\, B_4 + B_5 \quad (2.18)$$

where the total mobility term is

$$\lambda_t = \lambda_o + \lambda_g + \lambda_w \quad (2.19)$$

the PVT term is

$$B_1 = h\phi\,(S_o C_o + S_w C_w + S_g C_g + C_r) \quad (2.20)$$

$$C_o = \frac{1}{\rho_o}\frac{\partial \rho_o}{\partial P_o} + \frac{\rho_o}{\rho_g}\frac{\partial R_{so}}{\partial P_o} \quad (2.21)$$

$$C_w = \frac{1}{\rho_w}\frac{\partial \rho_w}{\partial P_o} + \frac{\rho_w}{\rho_g}\frac{\partial R_{sw}}{\partial P_o} \quad (2.22)$$

$$C_g = \frac{1}{\rho_g}\frac{\partial \rho_g}{\partial P_o} \simeq \frac{1}{P_o} \quad (2.23)$$

$$C_r = \frac{1}{\phi}\frac{\partial \phi}{\partial P_o} \quad (2.24)$$

the production term is

$$B_2 = \frac{-1}{A} \left\{ \frac{q_g \rho_g^*}{\rho_g} + q_o \rho_o^* \left(\frac{1}{\rho_o} - \frac{R_{so}}{\rho_g} \right) + q_w \rho_w^* \left(\frac{1}{\rho_w} - \frac{R_{sw}}{\rho_g} \right) \right\} \quad (2.25)$$

the capillary pressure term is

$$B_3 = - \left\{ \frac{\partial}{\partial x} \left(h\lambda_g \frac{\partial P_{cg}}{\partial x} \right) + \frac{\partial}{\partial y} \left(h\lambda_g \frac{\partial P_{cg}}{\partial y} \right) \right\}$$

$$+ \left\{ \frac{\partial}{\partial x} \left(h\lambda_w \frac{\partial P_{cw}}{\partial x} \right) + \frac{\partial}{\partial y} \left(h\lambda_w \frac{\partial P_{cw}}{\partial y} \right) \right\} \quad (2.26)$$

the gravity term is

$$B_4 = - \sum_j^{o,g,w} \left[\frac{\partial}{\partial x} \left\{ h\lambda_j \frac{\partial (\rho_j g D)}{\partial x} \right\} + \frac{\partial}{\partial y} \left\{ h\lambda_j \frac{\partial (\rho_j g D)}{\partial y} \right\} \right] \quad (2.27)$$

and the second degree derivative term is

$$B_5 = - \sum_j^{o,g,w} \left\{ \frac{(h\lambda_j)_x}{\rho_j} \left(\frac{\partial \Phi_j}{\partial x} \right)^2 \frac{\partial \rho_j}{\partial \Phi_j} + \frac{(h\lambda_j)_y}{\rho_j} \left(\frac{\partial \Phi_j}{\partial y} \right)^2 \frac{\partial \rho_j}{\partial \Phi_j} \right\}$$

$$- \sum_j^{o,w} \left\{ \frac{(h\rho_j\lambda_j)_x}{\rho_g} \left(\frac{\partial \Phi_j}{\partial x} \right)^2 \frac{\partial R_{sj}}{\partial \Phi_j} + \frac{(h\rho_j\lambda_j)_y}{\rho_g} \left(\frac{\partial \Phi_j}{\partial y} \right)^2 \frac{\partial R_{sj}}{\partial \Phi_j} \right\} \quad (2.28)$$

Potential gradients are assumed to be small and the square of such terms are neglected, therefore

$$B_5 = 0 \quad (2.29)$$

The following auxiliary set of equations are necessary to define

equation (2.18):

Capillary pressure equations

$$P_g - P_o = P_{cg} = f_1 (S_o, S_g, S_w) \tag{2.30}$$

$$P_o - P_w = P_{cw} = f_2(S_o, S_g, S_w) \tag{2.31}$$

Density equations

$$\rho_o = f_3 (P_o) \tag{2.32}$$

$$\rho_g = f_4 (P_g) \tag{2.33}$$

$$\rho_w = f_5 (P_w) \tag{2.34}$$

Relative permeability equations

$$k_{ro} = f_6 (S_o, S_g, S_w) \tag{2.35}$$

$$k_{rg} = f_7 (S_o, S_g, S_w) \tag{2.36}$$

$$k_{rw} = f_8 (S_o, S_g, S_w) \tag{2.37}$$

Reservoir geometry equations

$$D = f_9(x,y) \tag{2.38}$$

$$h = f_{10}(x,y) \tag{2.39}$$

Rock property equations

$$k = f_{11}\ (x,y) \tag{2.40}$$

$$\phi = f_{12}\ (P_o, \frac{1}{\phi}\ \frac{\partial \phi}{\partial P_o}) \tag{2.41}$$

Solubility equations

$$R_{so} = f_{13}\ (P_o) \tag{2.42}$$

$$R_{sw} = f_{14}\ (P_w) \tag{2.43}$$

Viscosity equations

$$\mu_o = f_{15}\ (P_o) \tag{2.44}$$

$$\mu_g = f_{16}\ (P_g) \tag{2.45}$$

$$\mu_w = f_{17}\ (P_w) \tag{2.46}$$

Initial condition equations

$$P_o(x,y,0) = P_{o_i}\ (x,y) \tag{2.47}$$

$$S_o(x,y,0) = S_{o_i}\ (x,y) \tag{2.48}$$

$$S_w(x,y,0) = S_{w_i}(x,y) \tag{2.49}$$

Boundary condition equations (Newmann type)

$$\frac{\partial P_o}{\partial x} = 0 \tag{2.50}$$

$$\frac{\partial P_o}{\partial y} = 0 \tag{2.51}$$

Equation (2.18) with the auxilury equations constitute a complete mathematical formulation of the two-dimensional three-phase simulation problem using implicit pressure explicit saturation (IMPES) technique.

The mathematical derivation is similar to that given by Peaceman and Rachford[11] and Crichlow[6] except for the SI units incorporated in this derivation.

CHAPTER III

FINITE DIFFERENCE SYSTEM

Equation (2.18) is far too complicated to be amenable to analytical solution. This nonlinear partial differential equation is approximated by finite differences where a variety of numerical techniques are adequate to solve it.

This chapter develops the finite difference system necessary to solve equation (2.18). In applying difference techniques, the continuum of space and time is divided into discrete intervals. Figure 3.1 shows the spatial discretized reservoir system. Thus, the difference system is defined only at the points (i,n) where i is the location of the cell and n is the time level. Figure 3.2 shows the definition of the cell system.

In this study, the implicit pressure - explicit saturation (IMPES) method will be used. The IMPES method solves equation (2.18) implicitly for pressure distribution. The saturation distribution is then explicitly calculated for each point.

3.1 Implicit Solution of Pressure

The finite difference formulation of equation (2.18) can be carried out by substituting for each term of the differential equation the appropriate finite difference approximation. Here, the noniterative

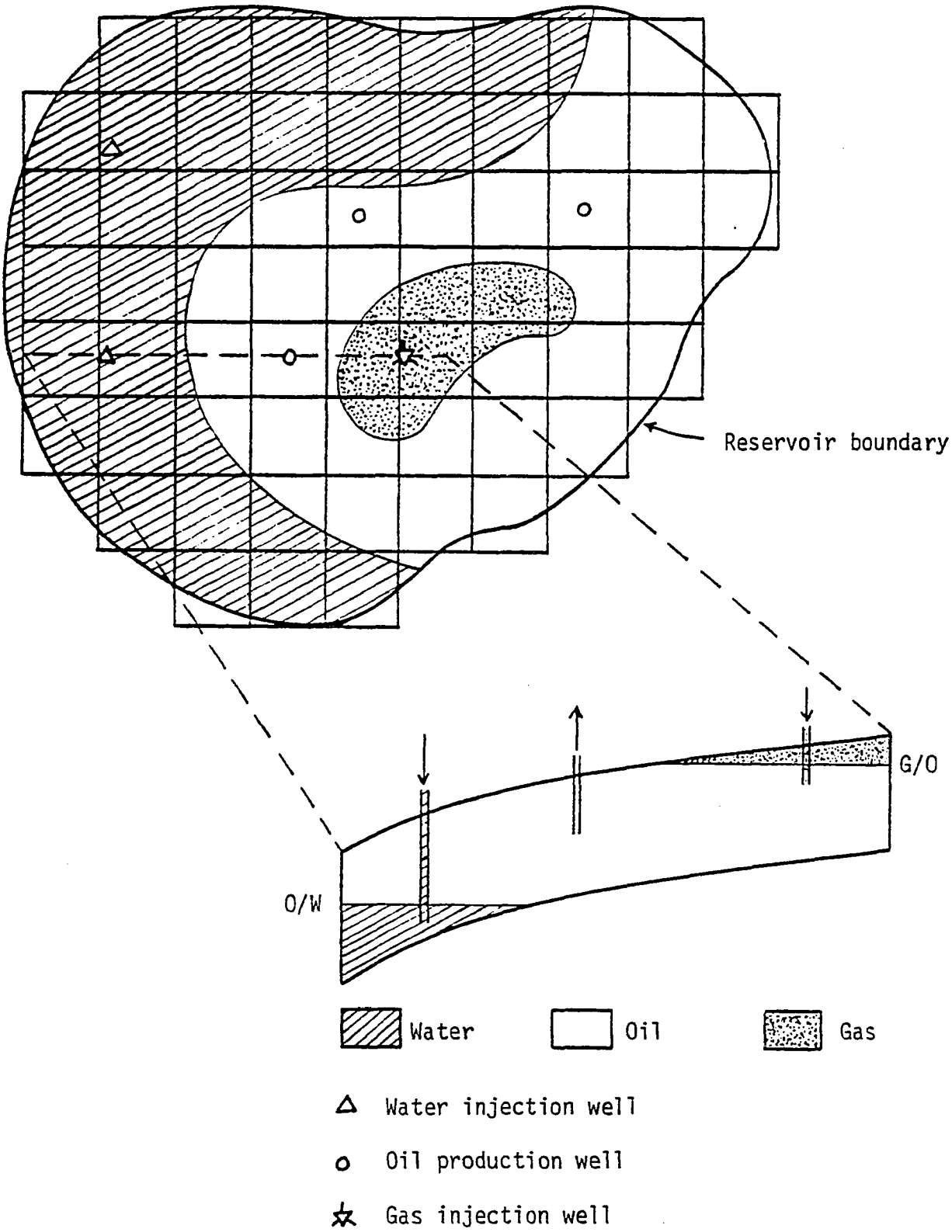
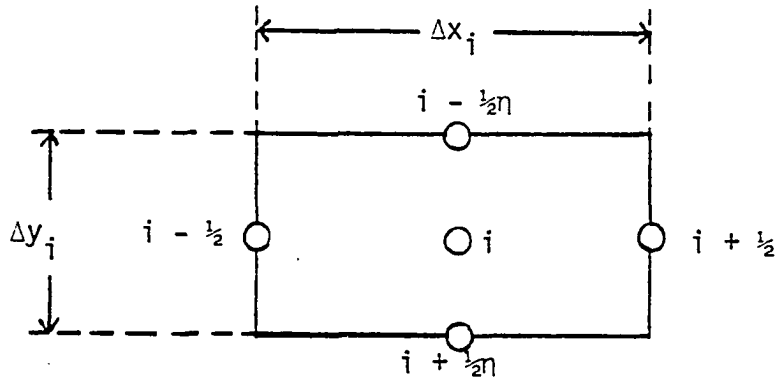
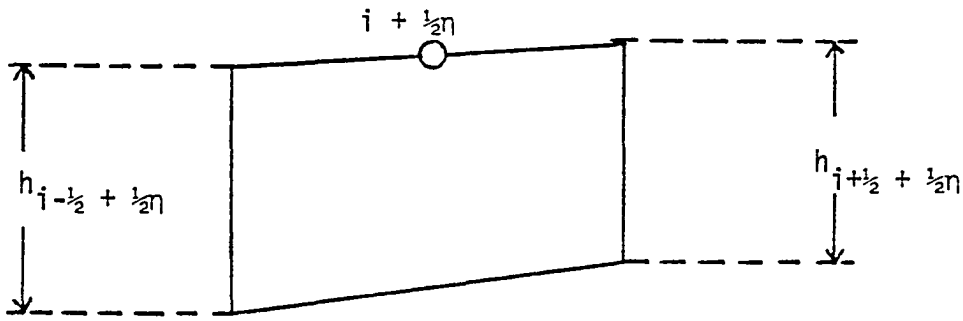


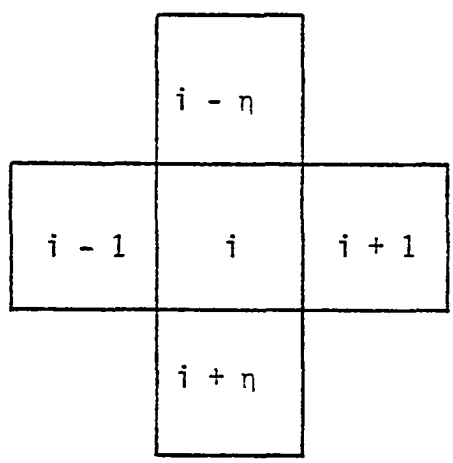
Figure 3.1 - Discretized Reservoir System



Typical cell - Top view



Typical cell - Side view



Typical cell - Arrangement

Figure 3.2 - Cell System Definition

IMPES approach will be used. Thus, the mobility, capillary and gravity terms will be evaluated at the previous time level. Hence equation (2.18) becomes

$$\frac{\partial}{\partial x}\left(h\lambda_t \frac{\partial P^{n+1}}{\partial x}\right) + \frac{\partial}{\partial y}\left(h\lambda_t \frac{\partial P^{n+1}}{\partial y}\right) = 11.57407408\, B_1^n \frac{\partial P^{n+1}}{\partial t} + QT \quad (3.1)$$

for simplicity the subscript "o" is dropped from P.

Each term of equation (3.1) can be differenced separately. The finite difference form of the first term on the left of equation (3.1) is:

$$\frac{\partial}{\partial x}\left(h\lambda_t \frac{\partial P^{n+1}}{\partial x}\right)_i = \frac{\theta}{\Delta x_i}\left\{(h\lambda_t)_{x_{i+\frac{1}{2}}} \frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x_{i+\frac{1}{2}}} - (h\lambda_t)_{x_{i-\frac{1}{2}}} \frac{P_i^{n+1} - P_{i-1}^{n+1}}{\Delta x_{i-\frac{1}{2}}}\right\}$$

$$+ \frac{(1-\theta)}{\Delta x_i}\left\{(h\lambda_t)_{x_{i+\frac{1}{2}}} \frac{P_{i+1}^{n} - P_i^{n}}{\Delta x_{i+\frac{1}{2}}} - (h\lambda_t)_{x_{i-\frac{1}{2}}} \frac{P_i^{n} - P_{i-1}^{n}}{\Delta x_{i-\frac{1}{2}}}\right\} \quad (3.2)$$

where the variables subscripted as $i-\frac{1}{2}$ or $i+\frac{1}{2}$ are the values of those variables evaluated at the boundary between the i'th and i-1, or i'th and i+1 cells, respectively. The parameter θ is a weighted factor such that if

$\theta = 0$, then the scheme is explicit

$\theta = \frac{1}{2}$, then the scheme is Crank-Nicholson

$\theta = 1$, then the scheme is fully implicit

Similarly the finite difference form of the second term on the left of equation (3.1) is

$$\frac{\partial}{\partial y}\left(h\lambda_t \frac{\partial P^{n+1}}{\partial y}\right)_i = \frac{\theta}{\Delta y_i}\left\{(h\lambda_t)_{y_{i+\frac{1}{2}\eta}} \frac{P_{i+\eta}^{n+1} - P_i^{n+1}}{\Delta y_{i+\frac{1}{2}\eta}} - (h\lambda_t)_{y_{i-\frac{1}{2}\eta}} \frac{P_i^{n+1} - P_{i-\eta}^{n+1}}{\Delta y_{i-\frac{1}{2}\eta}}\right\}$$

$$+ \frac{(1-\theta)}{\Delta y_i}\left\{(h\lambda_t)_{y_{i+\frac{1}{2}\eta}} \frac{P_{i+\eta}^{n} - P_i^{n}}{\Delta y_{i+\frac{1}{2}\eta}} - (h\lambda_t)_{y_{i-\frac{1}{2}\eta}} \frac{P_i^{n} - P_{i-\eta}^{n}}{y_{i-\frac{1}{2}\eta}}\right\} \quad (3.3)$$

where the variables subscripted as $i-\frac{1}{2}\eta$ or $i+\frac{1}{2}\eta$ are the values of those variables evaluated at the boundary between the i'th and $i-\eta$, or i'th and $i+\eta$ cells, respectively.

The time derivative of pressure is differenced as follows:

$$\left(\frac{\partial p^{n+1}}{\partial t}\right)_i = \frac{p_i^{n+1} - p_i^n}{\Delta t} \tag{3.4}$$

where

$$\Delta t = t^{n+1} - t^n \tag{3.5}$$

After the development of the finite difference equations for each term of equation (3.1), the linearized finite difference system can be written in the following rearranged form

$$e_i p_{i-\eta}^{n+1} + a_i p_{i-1}^{n+1} + b_i p_i^{n+1} + c_i p_{i+1}^{n+1} + f_i p_{i+\eta}^{n+1} = d_i \tag{3.6}$$

where the coefficients are

$$e_i = \frac{\theta(h\lambda_t)_{y_{i-\frac{1}{2}\eta}}}{\Delta y_i \, \Delta y_{i-\frac{1}{2}\eta}} \tag{3.7}$$

$$a_i = \frac{\theta(h\lambda_t)_{x_{i-\frac{1}{2}}}}{\Delta x_i \, \Delta x_{i-\frac{1}{2}}} \tag{3.8}$$

$$c_i = \frac{\theta(h\lambda_t)_{x_{i+\frac{1}{2}}}}{\Delta x_i \Delta x_{i+\frac{1}{2}}} \tag{3.9}$$

$$f_i = \frac{\theta(h\lambda_t)_{y_{i+\frac{1}{2}\eta}}}{\Delta y_i \, \Delta y_{i+\frac{1}{2}\eta}} \tag{3.10}$$

$$b_i = -e_i - a_i - c_i - f_i - 11.57407408 \, B_{1i}/\Delta t \tag{3.11}$$

the right hand side is

$$d_i = -11.57407408\ B_{1i}P_i^n/\Delta t + OT_i \tag{3.12}$$

where the variable OT_i consists of all of the values based on the last time step and is defined by

$$OT_i = 11.57407408\ B_{2i} + B_{3i} + 0.001\ B_{4i} + CN_i \tag{3.13}$$

the Crank-Nicholson term is:

$$CN_i = \frac{1-\theta}{\theta}\{-e_iP_{i-\eta}^n - a_iP_{i-1}^n + (e_i + a_i + c_i + f_i)\ P_i^n$$

$$-c_iP_{i+1}^n - f_iP_{i+\eta}^n\} \tag{3.14}$$

the weighted dimensions of the cells are

$$\Delta y_{i-\frac{1}{2}\eta} = \tfrac{1}{2}\ (\Delta y_i + \Delta y_{i-\eta}) \tag{3.15}$$

$$\Delta x_{i-\frac{1}{2}} = \tfrac{1}{2}\ (\Delta x_i + \Delta x_{i-1}) \tag{3.16}$$

$$\Delta x_{i+\frac{1}{2}} = \tfrac{1}{2}(\ \Delta x_{i+1} + \Delta x_i) \tag{3.17}$$

$$\Delta y_{i+\frac{1}{2}\eta} = \tfrac{1}{2}(\Delta y_{i+\eta} + \Delta y_i) \tag{3.18}$$

the PVT term is

$$B_{1i} = h_i\phi_i\ (S_oC_o + S_wC_w + S_gC_g + C_r)_i \tag{3.19}$$

The production term is

$$B_{2i} = -\frac{1}{\Delta x_i\Delta y_i}\left\{\frac{q_g\rho_g^*}{\rho_g} + q_o\rho_o^*\left(\frac{1}{\rho_o} - \frac{R_{so}}{\rho_g}\right) + q_w\rho_w^*\left(\frac{1}{\rho_w} - \frac{R_{sw}}{\rho_g}\right)\right\}_i \tag{3.20}$$

the capillary pressure term is

$$B_{3i} = B_{3wi} - B_{3gi} \tag{3.21}$$

$$B_{3ji} = (h\lambda_j)_{x_{i+\frac{1}{2}}} \frac{(P_{cj})_{i+1} - (P_{cj})_i}{\Delta x_i \, \Delta x_{i+\frac{1}{2}}}$$

$$- (h\lambda_j)_{x_{i-\frac{1}{2}}} \frac{(P_{cj})_i - (P_{cj})_{i-1}}{\Delta x_i \Delta x_{i-\frac{1}{2}}}$$

$$+ (h\lambda_j)_{y_{i+\frac{1}{2}\eta}} \frac{(P_{cj})_{i+\eta} - (P_{cj})_i}{\Delta y_i \, \Delta y_{i+\frac{1}{2}\eta}}$$

$$- (h\lambda_j)_{y_{i-\frac{1}{2}\eta}} \frac{(P_{cj})_i - (P_{cj})_{i-\eta}}{\Delta y_i \Delta y_{i-\frac{1}{2}\eta}} \tag{3.22}$$

and the gravity term is

$$B_{4i} = - \sum_{j}^{o,g,w} \{ (h\lambda_j)_{x_{i+\frac{1}{2}}} \frac{(\rho_j g D)_{i+1} - (\rho_j g D)_i}{\Delta x_i \quad \Delta x_{i+\frac{1}{2}}}$$

$$- (h\lambda_j)_{x_{i-\frac{1}{2}}} \frac{(\rho_j g D)_i - (\rho_j g D)_{i-1}}{\Delta x_i \quad \Delta x_{i-\frac{1}{2}}}$$

$$+ (h\lambda_j)_{y_{i+\frac{1}{2}\eta}} \frac{(\rho_j g D)_{i+\eta} - (\rho_j g D)_i}{\Delta y_i \quad \Delta y_{i+\frac{1}{2}\eta}}$$

$$- (h\lambda_j)_{y_{i-\frac{1}{2}\eta}} \frac{(\rho_j g D)_i - (\rho_j g D)_{i-\eta}}{\Delta y_i \quad \Delta y_{i-\frac{1}{2}\eta}} \} \tag{3.23}$$

If the finite difference approximation, equation (3.6), developed for cell i is written for all cells in the x-y rectangular mesh, a set of linear equations is obtained. Those equations can be written in the following matrix form:

$$\underline{\underline{A}} \, \underline{P} = \underline{d} \tag{3.24}$$

The form of matrix $\mathbb{A}$ depends upon the ordering scheme by which the cells of the x-y mesh are linearly indexed. This system of equations can be solved either by iterative methods or by direct methods. These methods will be discussed in chapter IV, V, and VI.

3.2 Explicit Solution of Saturations

The new oil saturation can be developed from the oil phase equation. Neglecting the second degree derivative term in equation (2.12), the saturation time derivative would be:

$$\frac{\partial S_o}{\partial t} = \frac{1}{h\phi}\left[\frac{\partial}{\partial x}\left(h\lambda_o \frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial x}\left\{h\lambda_o \frac{\partial(\rho_o g D)}{\partial x}\right\}\right.$$

$$\left. + \frac{\partial}{\partial y}\left(h\lambda_o \frac{\partial P}{\partial y}\right) + \frac{\partial}{\partial y}\left\{h\lambda_o \frac{\partial(\rho_o g D)}{\partial y}\right\}\right]$$

$$- \frac{S_o}{\rho_o}\frac{\partial \rho_o}{\partial t} - \frac{S_o}{\phi}\frac{\partial \phi}{\partial t} + \frac{q_o \rho_o^*}{h\phi A \rho_o} \qquad (3.25)$$

Similarly, the new water saturation can be developed from the water phase equation. Neglecting the second degree derivative term in equation (2.13), the saturation time derivative would be:

$$\frac{\partial S_w}{\partial t} = \frac{1}{h\phi}\left[\frac{\partial}{\partial x}\left(h\lambda_w \frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial x}\left\{h\lambda_w \frac{\partial(\rho_w g D)}{\partial x}\right\} - \frac{\partial}{\partial x}\left(h\lambda_w \frac{\partial P_{cw}}{\partial x}\right)\right.$$

$$\left. + \frac{\partial}{\partial y}\left(h\lambda_w \frac{\partial P}{\partial y}\right) + \frac{\partial}{\partial y}\left\{h\lambda_w \frac{\partial(\rho_w g D)}{\partial y}\right\} - \frac{\partial}{\partial y}\left(h\lambda_w \frac{\partial P_{cw}}{\partial y}\right)\right]$$

$$- \frac{S_w}{\rho_w}\frac{\partial \rho_w}{\partial t} - \frac{S_w}{\phi}\frac{\partial \phi}{\partial t} + \frac{q_w \rho_w^*}{h\phi A \rho_w} \qquad (3.26)$$

The finite difference approximation of the oil and water saturation equations is

$$S_{ji}^{n+1} = \Big[S_{ji}^{n} + \frac{0.0864\Delta t}{h_i \phi_i^{n+1}} \{ (h\lambda_j)_{x_{i+\frac{1}{2}}} \frac{p_{i+1}^{n+1} - p_i^{n+1}}{\Delta x_i \Delta x_{i+\frac{1}{2}}}$$

$$- (h\lambda_j)_{x_{i-\frac{1}{2}}} \frac{p_i^{n+1} - p_{i-1}^{n+1}}{\Delta x_i \Delta x_{i-\frac{1}{2}}} + (h\lambda_j)_{y_{i+\frac{1}{2}\eta}} \frac{p_{i+\eta}^{n+1} - p_i^{n+1}}{\Delta y_i \Delta y_{i+\frac{1}{2}\eta}}$$

$$- (h\lambda_j)_{y_{i-\frac{1}{2}\eta}} \frac{p_i^{n+1} - p_{i-\eta}^{n+1}}{\Delta y_i \Delta y_{i-\frac{1}{2}\eta}} \} + B_{ji}^{n} \Big] \Big/ \Big(3 - \frac{\phi_i^{n}}{\phi_i^{n+1}} - \frac{\rho_{ji}^{n}}{\rho_{ji}^{n+1}} \Big) \quad (3.27)$$

where

$$B_{oi} = \frac{\Delta t}{h_i \phi_i^{n+1} \Delta x_i \Delta y_i} \Big(\frac{q_{oi} \rho_o^*}{\rho_{oi}^{n+1}} \Big) + \frac{0.0864\Delta t}{h_i \phi_i^{n+1}} B_{4oi} \quad (3.28)$$

$$B_{wi} = \frac{\Delta t}{h_i \phi_i^{n+1} \Delta x_i \Delta y_i} \Big(\frac{q_{wi} \rho_w^*}{\rho_{wi}^{n+1}} \Big) + \frac{0.0864\Delta t}{h_i \phi_i^{n+1}} (-B_{3wi} + B_{4wi}) \quad (3.29)$$

$$B_{3wi} = (h\lambda_w)_{x_{i+\frac{1}{2}}} \frac{(P_{cw})_{i+1} - (P_{cw})_i}{\Delta x_i \Delta x_{i+\frac{1}{2}}} - (h\lambda_w)_{x_{i-\frac{1}{2}}} \frac{(P_{cw})_i - (P_{cw})_{i-1}}{\Delta x_i \Delta x_{i-\frac{1}{2}}}$$

$$+ (h\lambda_w)_{y_{i+\frac{1}{2}\eta}} \frac{(P_{cw})_{i+\eta} - (P_{cw})_i}{\Delta y_i \Delta y_{i+\frac{1}{2}\eta}} - (h\lambda_w)_{y_{i-\frac{1}{2}\eta}} \frac{(P_{cw})_i - (P_{cw})_{i-\eta}}{\Delta y_i \Delta y_{i-\frac{1}{2}\eta}} \quad (3.30)$$

$$B_{4ji} = 0.001\, g \{ (h\lambda_j)_{x_{i+\frac{1}{2}}} \frac{(\rho_j D)_{i+1} - (\rho_j D)_i}{\Delta x_i \Delta x_{i+\frac{1}{2}}} - (h\lambda_j)_{x_{i-\frac{1}{2}}} \frac{(\rho_j D)_i - (\rho_j D)_{i-1}}{\Delta x_i \Delta x_{i-\frac{1}{2}}}$$

$$+ (h\lambda_j)_{y_{i+\frac{1}{2}\eta}} \frac{(\rho_j D)_{i+\eta} - (\rho_j D)_i}{\Delta y_i \Delta y_{i+\frac{1}{2}\eta}} - (h\lambda_j)_{y_{i-\frac{1}{2}\eta}} \frac{(\rho_j D)_i - (\rho_j D)_{i-\eta}}{\Delta y_i \Delta y_{i-\frac{1}{2}\eta}} \} \quad (3.31)$$

The gas saturation is determined from equation (2.10)

$$S_{g_i} = 1 - S_{o_i} - S_{w_i} \tag{3.32}$$

3.3 Discretization of the Flow Coefficient ($h\lambda$)

The flow coefficient is broken into three different units as follows:

$$(h\lambda_j)_{x_{i-\frac{1}{2}}} = (k_x h)_{i-\frac{1}{2}} (k_{rj})_{i-\frac{1}{2}} / (\mu_j)_i \tag{3.33}$$

$$(h\lambda_j)_{y_{i-\frac{1}{2}\eta}} = (k_y h)_{i-\frac{1}{2}\eta} (k_{rj})_{i-\frac{1}{2}\eta} / (\mu_j)_i \tag{3.34}$$

where $j = o,g,w$

The coefficient $(k_x h)_{i-\frac{1}{2}}$ is evaluated by averaging k_x and h at cells i and i-1. The average k_x for series flow in linear bed is[4]

$$k_{x_{i-\frac{1}{2}}} = \frac{\Delta x_i + \Delta x_{i-1}}{\frac{\Delta x_i}{k_{x_i}} + \frac{\Delta x_{i-1}}{k_{x_{i-1}}}} \tag{3.35}$$

The value of $h_{i-\frac{1}{2}}$ is evaluated by considering a profile of two adjacent cells as shown in Figure 3.3. The shaded area in Figure 3.3 is equal to

$$\frac{h_{i-1} + h_i}{2} \left(\frac{\Delta x_i}{2} + \frac{\Delta x_{i-1}}{2} \right) \tag{3.36}$$

but, this area is also equal to

$$\frac{h_i + h_{i-\frac{1}{2}}}{2} \left(\frac{\Delta x_{i-1}}{2} \right) + \frac{h_{i-\frac{1}{2}} + h_i}{2} \left(\frac{\Delta x_i}{2} \right) \tag{3.37}$$

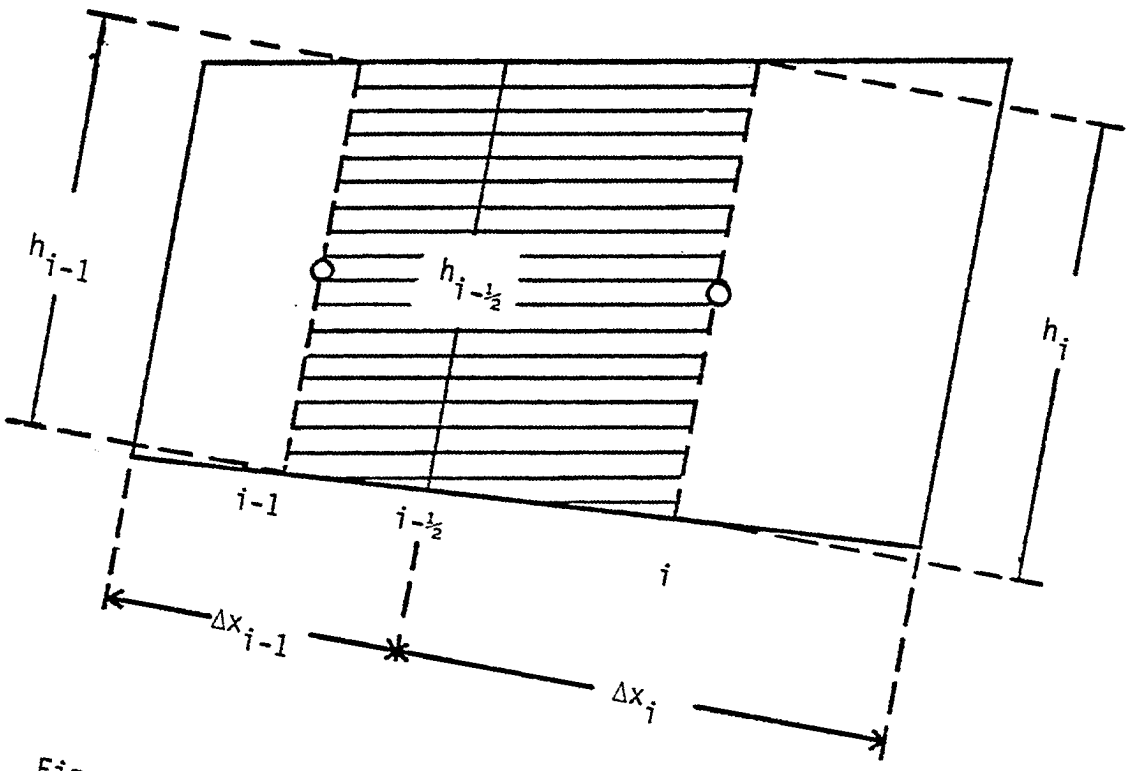


Figure 3.3 - Profile of Two Adjacent Cells

Equating equations (3.36) and (3.37) and solving for $h_{i-\frac{1}{2}}$, we obtain

$$h_{i-\frac{1}{2}} = \frac{h_{i-1}\Delta x_i + h_i \Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_i} \tag{3.38}$$

Combining equations (3.35) and (3.38), we get

$$(k_x h)_{i-\frac{1}{2}} = \frac{\Delta x_i h_{i-1} + \Delta x_{i-1} h_i}{\Delta x_i / k_{x_i} + \Delta x_{i-1} / k_{x_{i-1}}} \tag{3.39}$$

At the left boundary where i-1 is outside the reservoir

$$(k_x h)_{i-\frac{1}{2}} = (k_x h)_i \tag{3.40}$$

and at the right boundary where i is outside the reservoir

$$(k_x h)_{i-\frac{1}{2}} = (k_x h)_{i-1} \tag{3.41}$$

with similar equations for $(k_y h)_{i-\frac{1}{2}\eta}$.

The extrapolated realtive permeability method developed by Todd, et al.[16] is used to evaluate $(k_r)_{x_{i-\frac{1}{2}}}$ and $(k_r)_{y_{i-\frac{1}{2}\eta}}$ for oil, water and gas phases. For flow in the plus x-direction (ascending values of i)

$$(k_r)_{x_{i-\frac{1}{2}}} = k_{r_{i-1}} + \frac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_{i-2}} \left(k_{r_{i-1}} - k_{r_{i-2}}\right) \tag{3.42}$$

For flow in the minus x-direction (decreasing values of i)

$$(k_r)_{x_{i-\frac{1}{2}}} = k_{r_i} - \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}} \left(k_{r_{i+1}} - k_{r_i}\right) \tag{3.43}$$

The above two equations are subjected to the following constraints:

$$(k_r)_{x_{i-\frac{1}{2}}} \geq 0 \tag{3.44}$$

and

$$(k_r)_{x_{i-\frac{1}{2}}} \leq \max \{(k_r)_{x_i}, (k_r)_{x_{i-1}}\} \tag{3.45}$$

with similar equations and constraints for $(k_r)_{y_{i-\frac{1}{2}\eta}}$.

It is obvious from equation (3.42) and (3.43) that the method of Todd, et al.[16] is based on the extrapolation of relative permeability value at the interblock location based on the two upstream values. At the points where application of equation (3.42) and (3.43) would result in error, the single point upstream relative permeability is used instead. This happens at the boundary and near sink or source cells. The upstream equation for flow in the plus x-direction is

$$(k_r)_{x_{i-\frac{1}{2}}} = k_{r_{i-1}} \tag{3.46}$$

and for flow in the minus x-direction is

$$(k_r)_{x_{i-\frac{1}{2}}} = k_{r_i} \tag{3.47}$$

with similar equations for $(k_r)_{y_{i-\frac{1}{2}\eta}}$. The flow chart for the calculation of interblock relative permeability is given in Figure 3.4.

The viscosity terms are evaluated at the cell i because the viscosity is a weak function of pressure.

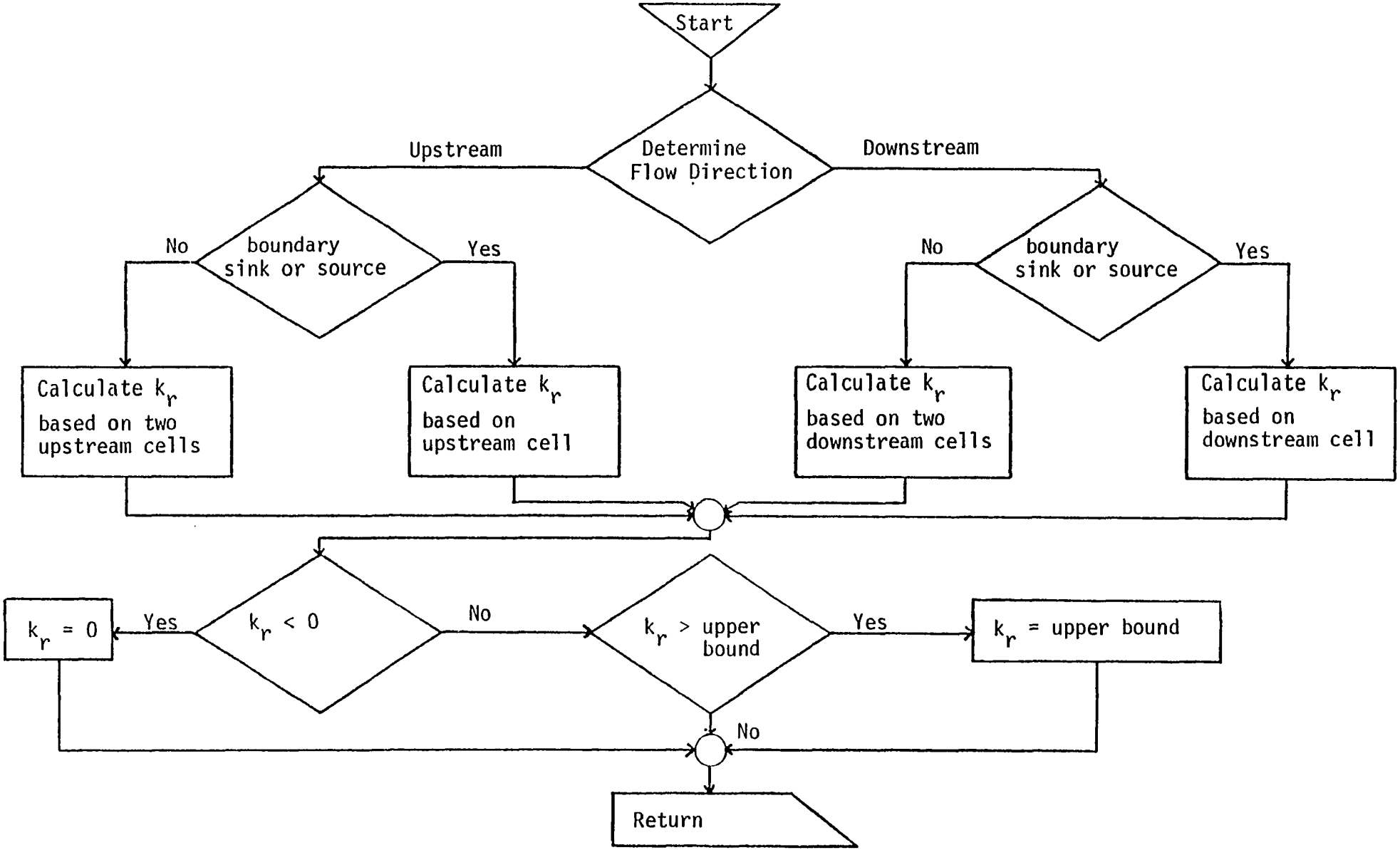


Figure 3.4 - Flow chart for the calculation of interblock relative permeability

3.4 Boundary Conditions

Equations (2.50) and (2.51) are satisfied by the use of reflection principle.

$$P_i - P_{i-\eta} = 0 \quad \forall i \ni h_{i-\eta} = 0 \tag{3.48}$$

$$P_i - P_{i-1} = 0 \quad \forall i \ni h_{i-1} = 0 \tag{3.49}$$

$$P_{i+1} - P_i = 0 \quad \forall i \ni h_{i+1} = 0 \tag{3.50}$$

$$P_{i+\eta} - P_i = 0 \quad \forall i \ni h_{i+\eta} = 0 \tag{3.51}$$

It is noticeable that these terms are associated with certain flow coefficients all the time. Equations (3.48) through (3.51) can be replaced by

$$(h\lambda_j)_{i-\frac{1}{2}\eta} = 0 \quad \forall\, i \ni h_{i-\eta} = 0 \tag{3.52}$$

$$(h\lambda_j)_{i-\frac{1}{2}} = 0 \quad \forall\, i \ni h_{i-1} = 0 \tag{3.53}$$

$$(h\lambda_j)_{i+\frac{1}{2}} = 0 \quad \forall\, i \ni h_{i+1} = 0 \tag{3.54}$$

$$(h\lambda_j)_{i+\frac{1}{2}\eta} = 0 \quad \forall\, i \ni h_{i+\eta} = 0 \tag{3.55}$$

where $j = o,w,g$.

CHAPTER IV

ITERATIVE METHODS

The iterative methods are based on solving the system of equations by successive approximations until the procedure converges to a pre-defined tolerance or an action due to the failure of convergence is taken.

The most popular and effective iterative methods used in petroleum reservoir simulation are

1. Successive Overrelaxation
2. Alternating Direction Implicit
3. Strongly Implicit

These iterative methods will be used to solve the system of linearized equations whose general form is

$$e_i P_{i-\eta} + a_i P_{i-1} + b_i P_i + c_i P_{i+1} + f_i P_{i+\eta} = d_i \qquad (4.1)$$

In this study, the maximum absolute relative error of successive iterations is used as the convergence criterion. The maximum absolute relative error r_{max} must be within the pre-defined tolerance ε as follows:

$$r_{max} \leq \varepsilon \qquad (4.2)$$

where

$$r_{max} = \max \left| r_i \right| \qquad \forall i \ni h_i \neq 0 \tag{4.3}$$

and

$$r_i = \frac{p_i^k - p_i^{k-1}}{p_i^k} \tag{4.4}$$

4.1 Successive Overrelaxation

The most popular versions of successive overrelaxation methods used in two dimensional petroleum reservoir simulation are the point and line relaxation.

Point Successive Overrelaxation (PSOR)

The PSOR method is an iterative technique in which improved estimate is computed by applying the following formula at each cell:

$$\tilde{p}_i^{k+1} = \frac{1}{b_i} \left(d_i - e_i p_{i-\eta}^{k+1} - a_i p_{i-1}^{k+1} - c_i p_{i+1}^{k} - f_i p_{i+\eta}^{k} \right) \tag{4.5}$$

After each point, the value of $\tilde{p}_i^{k+1}$ is relaxed

$$p_i^{k+1} = p_i^k + \omega \left(\tilde{p}_i^{k+1} - p_i^k \right) \tag{4.6}$$

The factor ω is called the relaxation parameter, and its presence accelerates the convergence process. The PSOR flow chart is given in Figure 4.1.

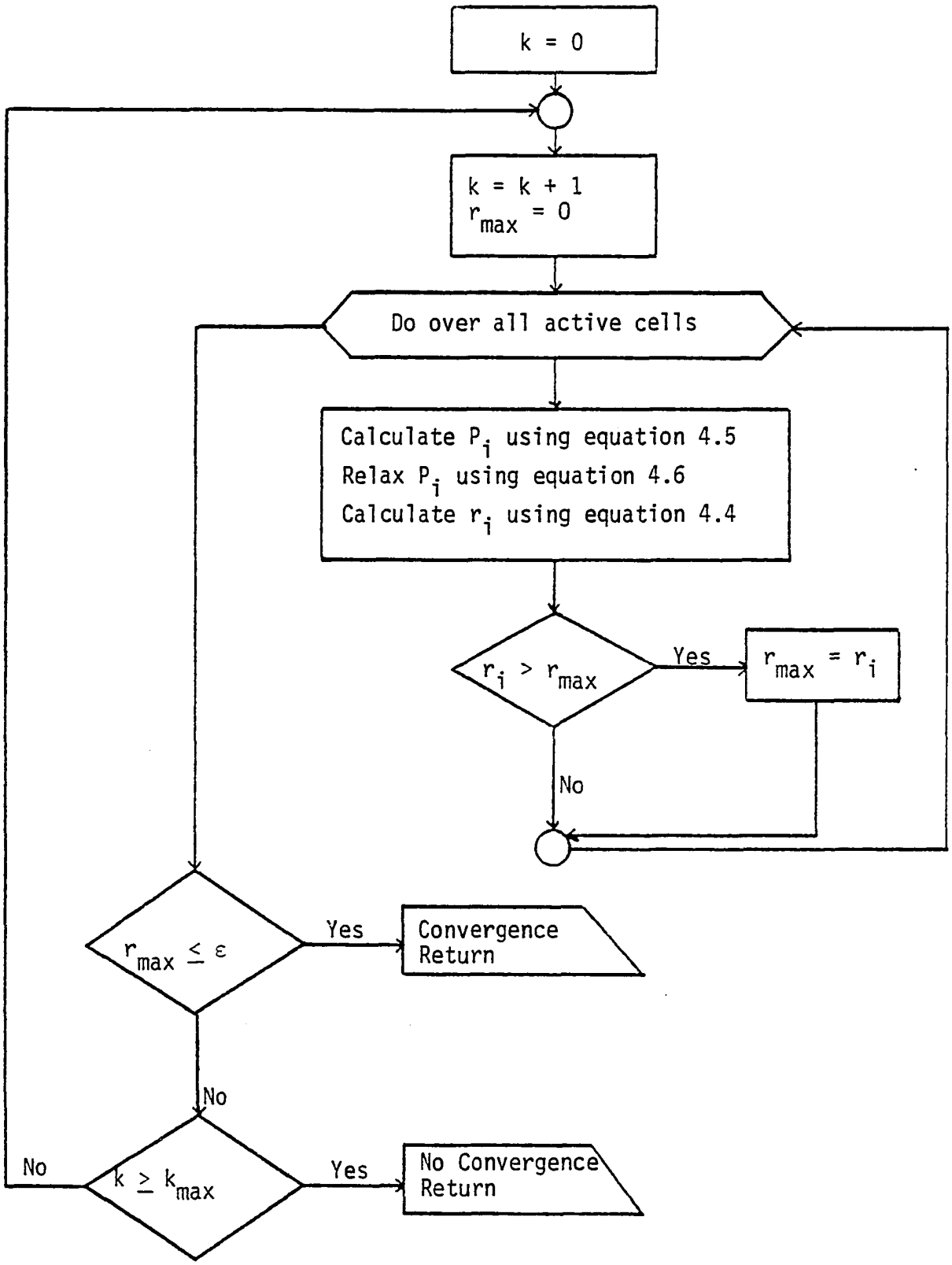


Figure 4.1 - PSOR flow chart

Line Successive Overrelaxation (LSOR)

This technique resolves the two-dimensional problem to a succession of temporary one-dimensional problems which can be solved rather efficiently. Equation (4.1) is written for a general row as

$$a_i \tilde{P}^{k+1}_{i-1} + b_i \tilde{P}^{k+1}_{i} + c_i \tilde{P}^{k+1}_{i+1} = d'_i \tag{4.7}$$

where

$$d'_i = d_i - e_i P^{k+1}_{i-\eta} - f_i P^{k}_{i+\eta} \tag{4.8}$$

This equation forms a close-band tridiagonal matrix which can be solved by the close-band Thomas algorithm outlined in the next section. The values of P_i obtained for each row at a time are relaxed

$$P^{k+1}_i = P^k_i + \omega (\tilde{P}^{k+1}_i - P^k_i) \tag{4.9}$$

where i is for all the elements in that given row. The LSOR flow chart is given in Figure 4.2.

Relaxation Parameter

An optimum relaxation parameter ω is necessary for maximum efficiency of overrelaxation methods. The selection of this optimum value requires the development of the relationship between the number of iterations required for convergence against a constant tolerance and the relaxation parameter. Several abbreviated simulation runs are usually made with various relaxion parameters. This data is plotted

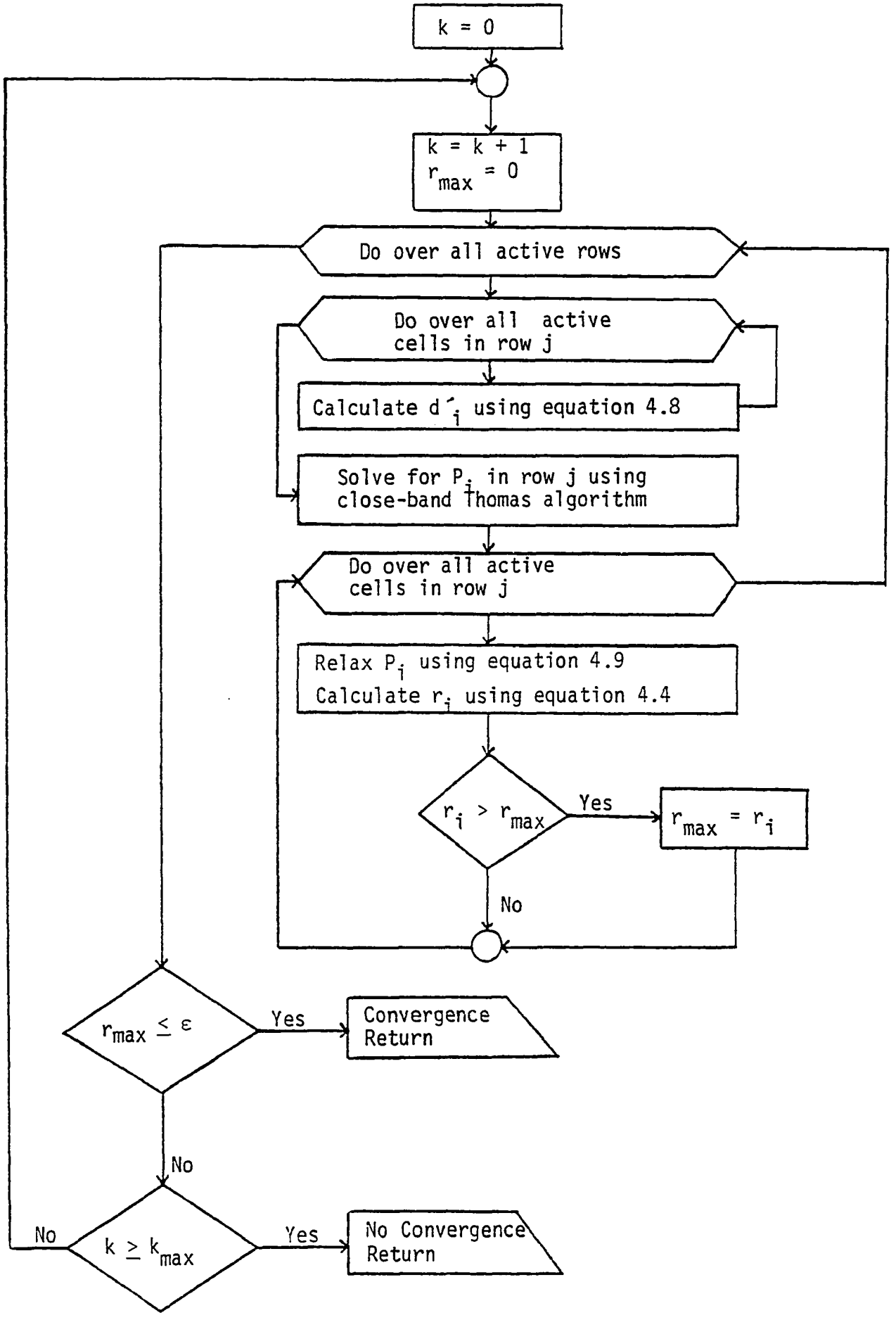


Figure 4.2 - LSOR flow chart

graphically as shown in Figure 4.3. The curve has a characteristic shape, and the best value of ω is the lowest point on the curve. Generally, the more homogeneous the system is, the closer the value of ω to 1.0, while the value of ω approaches 2.0 as the degree of anisotropy of the system increases.

The slope of ω curve is close to linear beyond the optimum ω. This implies that it is generally better to overestimate ω than to underestimate it.

4.2 Alternating Direction Implicit (ADI)

In the ADI[11] method, each iteration step is divided into two equal substeps. "During the first substep the cells are swept in the x-direction one row at a time solving for the unknown pressures. In the second substep the system is swept in the y-direction one column at a time solving for the unknown pressures."[6] The ADI flow chart is given in Figure 4.6.

X-Direction Sweep

Here only those terms contributed by $\frac{\partial}{\partial x}\ (h\lambda_t \frac{\partial P}{\partial x})$ and $\frac{\partial P}{\partial t}$ will be assumed unknowns. Hence at the k'th iteration, the general equation for the x-direction sweep is

$$a_i P_{i-1}^{k+\frac{1}{2}} + bx_i P_i^{k+\frac{1}{2}} + c_i P_{i+1}^{k+\frac{1}{2}} = dx_i \qquad (4.10)$$

where

$$bx_i = -(a_i + c_i + 11.57407408\ B_{1i}/\Delta t + \gamma_i \beta)$$

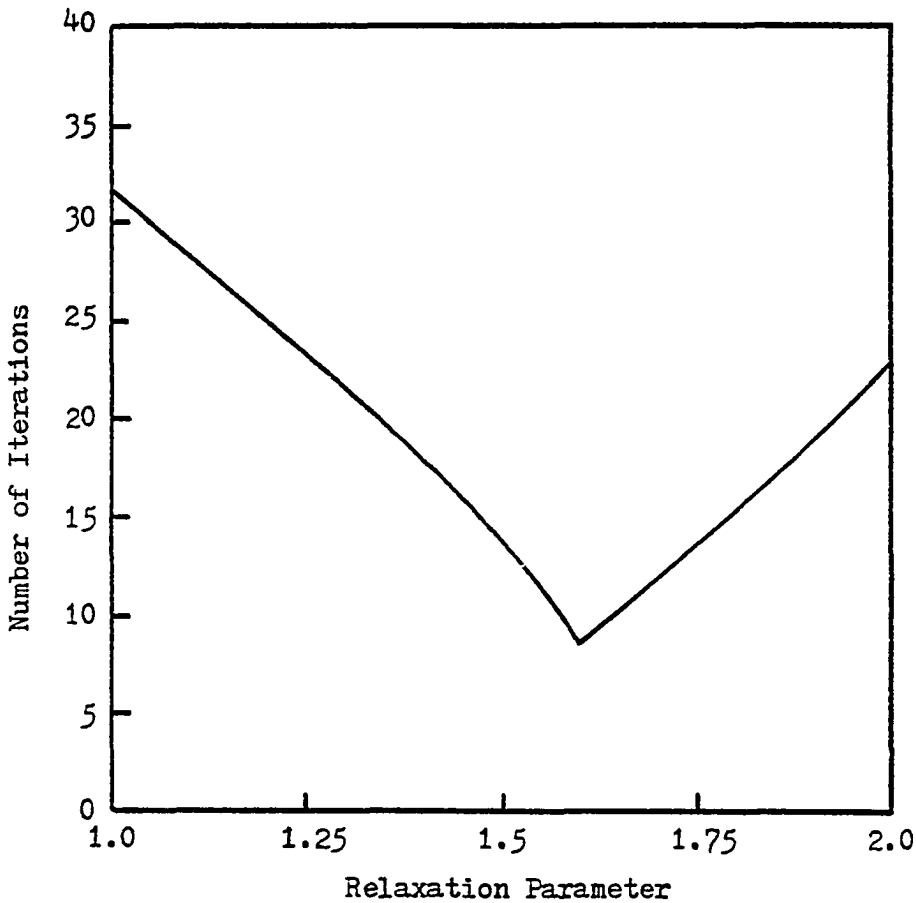


Figure 4.3 - Typical Plot of Relaxation Parameter at Fixed Tolerance

or

$$bx_i = b_i + e_i + f_i - \gamma_i\beta \tag{4.11}$$

$$dx_i = d_i - e_i P^k_{i-\eta} + (e_i + f_i - \gamma_i\beta)\, P^k_i - f_i P^k_{i+\eta} \tag{4.12}$$

and β is an acceleration parameter added to speed up convergence, γ_i is the normalizing factor to account for possible differences in areal size of the cell and for reservoir heterogeneity. The normalizing factor is defined in this study as

$$\gamma_i = \frac{(hk)_{avg}}{\Delta x_i\, \Delta y_i} \tag{4.13}$$

where

$$(hk)_{avg} = \frac{1}{N} \sum_{i=1}^{N} h_i \sqrt{k_{x_i} k_{y_i}} \tag{4.14}$$

Equation (4.10) forms a close-band tridiagonal matrix which can be solved by close-band Thomas algorithm.

Close-Band Thomas Algorithm

The equations are

$$a_i P_{i-1} + b_i P_i + c_i P_{i+1} = d_i$$

$$\text{for } 1 \leq i \leq N \qquad \text{with } a_1 = a_N = 0 \tag{4.15}$$

The algorithm is as follows:

First, for $1 \leq i \leq N$, compute

$$\zeta_i = b_i - \frac{a_i c_{i-1}}{\zeta_{i-1}} \qquad \text{with } \zeta_1 = b_1 \tag{4.16}$$

and

$$\xi_i = \frac{d_i - a_i \xi_{i-1}}{\zeta_i} \qquad \text{with } \xi_1 = \frac{d_1}{b_1} \tag{4.17}$$

The values of the unknowns (P's) are then computed from

$$P_N = \xi_N \qquad \text{and} \qquad P_i = \xi_i - \frac{c_i P_{i+1}}{\zeta_i}$$

$$\text{for } i = N-1, N-2, \ldots, 1 \tag{4.18}$$

the flow chart for close-band Thomas algorithm is given in Figure 4.4.

y-Direction Sweep

Here only those terms contributed by $\frac{\partial}{\partial y}(h\lambda_t \frac{\partial P}{\partial y})$ and $\frac{\partial P}{\partial t}$ will be assumed unknowns. Hence at the k'th iteration, the general equation for the y-direction sweep is

$$e_i P_{i-\eta}^{k+1} + by_i P_i^{k+1} + f_i P_{i+\eta}^{k+1} = dy_i \tag{4.19}$$

where

$$by_i = -(e_i + f_i + 11.57407408\ B_{1i}/\Delta t + \gamma_i \beta)$$

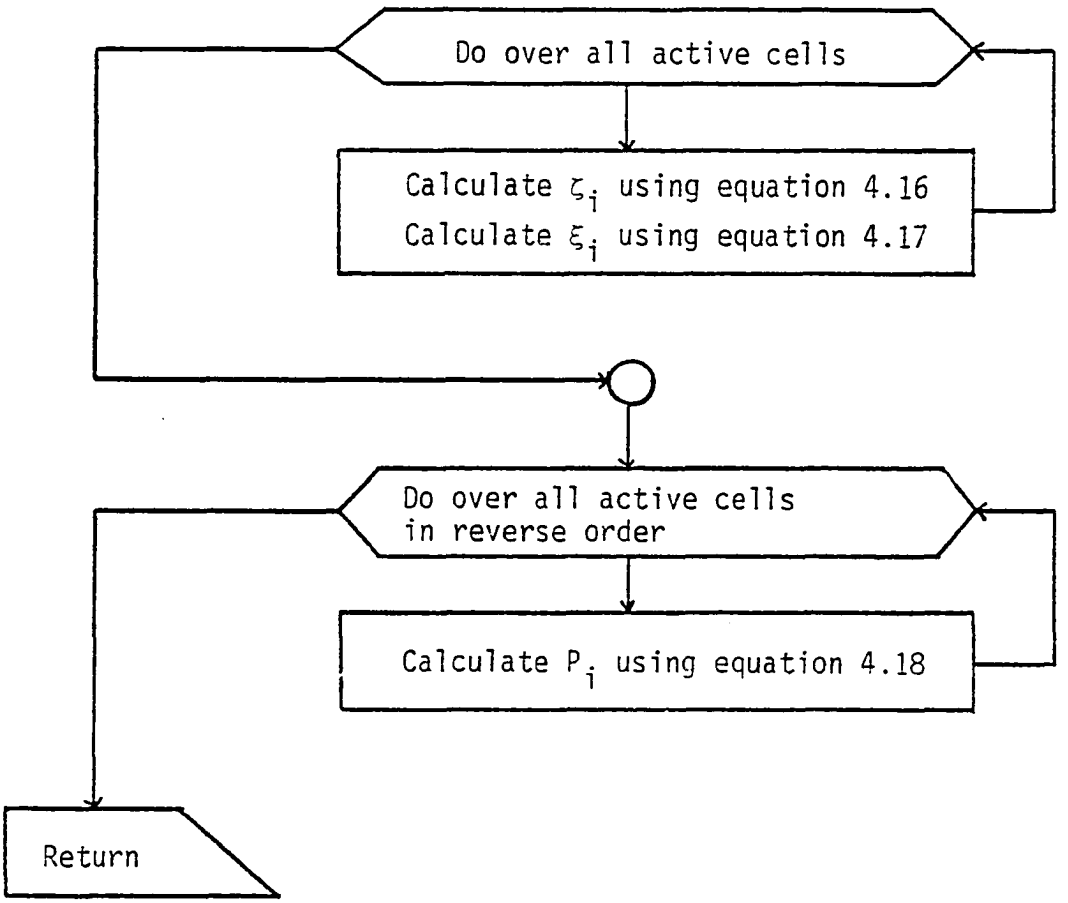


Figure 4.4 - Flow Chart for Close-Band Thomas Algorithm

or

$$by_i = b_i + a_i + c_i - \gamma_i\beta \tag{4.20}$$

$$dy_i = d_i - a_i P_{i-1}^{k+\frac{1}{2}} + (a_i + c_i - \gamma_i\beta) P_i^{k+\frac{1}{2}} - c_i P_{i+1}^{k+\frac{1}{2}} \tag{4.21}$$

Equation (4.19) forms a wide-band tridiagonal matrix which can be solved by wide-band Thomas Algorithm.

Wide-Band Thomas Algorithm

The equations are

$$e_i P_{i-\eta} + b_i P_i + f_i P_{i+\eta} = d_i$$

$$\text{for } 1 \leq i \leq N \quad \text{with } e_i = f_{N-i} = 0 \quad \text{for } 1 \leq i \leq \eta \tag{4.22}$$

The algorithm is as follows:

First for $1 \leq i \leq N$, compute

$$\zeta_i = b_i - \frac{e_i f_{i-\eta}}{\zeta_{i-\eta}} \qquad \text{with } \zeta_i = b_i \quad \text{for } 1 \leq i \leq \eta \tag{4.23}$$

and

$$\xi_i = \frac{d_i - e_i \xi_{i-\eta}}{\zeta_i} \qquad \text{with } \xi_i = \frac{d_i}{b_i} \quad \text{for } 1 \leq i \leq \eta \tag{4.24}$$

The values of the unknowns are then computed from

$$P_i = \xi_i \qquad \text{for } i=N, N-1, \ldots, N-\eta+1$$

and

$$P_i = \xi_i - \frac{f_i P_{i+\eta}}{\zeta_i} \qquad \text{for } i=N-\eta\ ,\ N-\eta-1,\ldots,1 \tag{4.25}$$

The flow chart for wide-band Thomas algorithm is given in Figure 4.5.

ADI Acceleration Parameters

To achieve a faster rate of convergence a series of parameters used in sequence is employed. Crichlow[6] outlined the procedure of Peaceman and Rachford[11] to determine the upper and lower limits of the parameter range from the reservoir data.

Four parameters are defined as functions of the x- and y-direction interblock transmissibilities (T_x and T_y) and the number of cells in the x- and y-direction (N_x and N_y) in the following manner

$$M_1 = \frac{2T_x}{T_x + T_y} \quad \frac{\pi^2}{4N_x^2} \tag{4.26}$$

$$M_2 = \frac{2T_x}{T_x + T_y} \tag{4.27}$$

$$M_3 = \frac{2T_y}{T_x + T_y} \quad \frac{\pi^2}{4N_y^2} \tag{4.28}$$

$$M_4 = \frac{2T_y}{T_x + T_y} \tag{4.29}$$

where

$$T_x = \frac{\sum_{i=1}^{N} k_{x_i} \Delta y_i/\Delta x_i}{N} \tag{4.30}$$

$$T_y = \frac{\sum_{i=1}^{N} k_{y_i} \Delta x_i/\Delta y_i}{N} \tag{4.31}$$

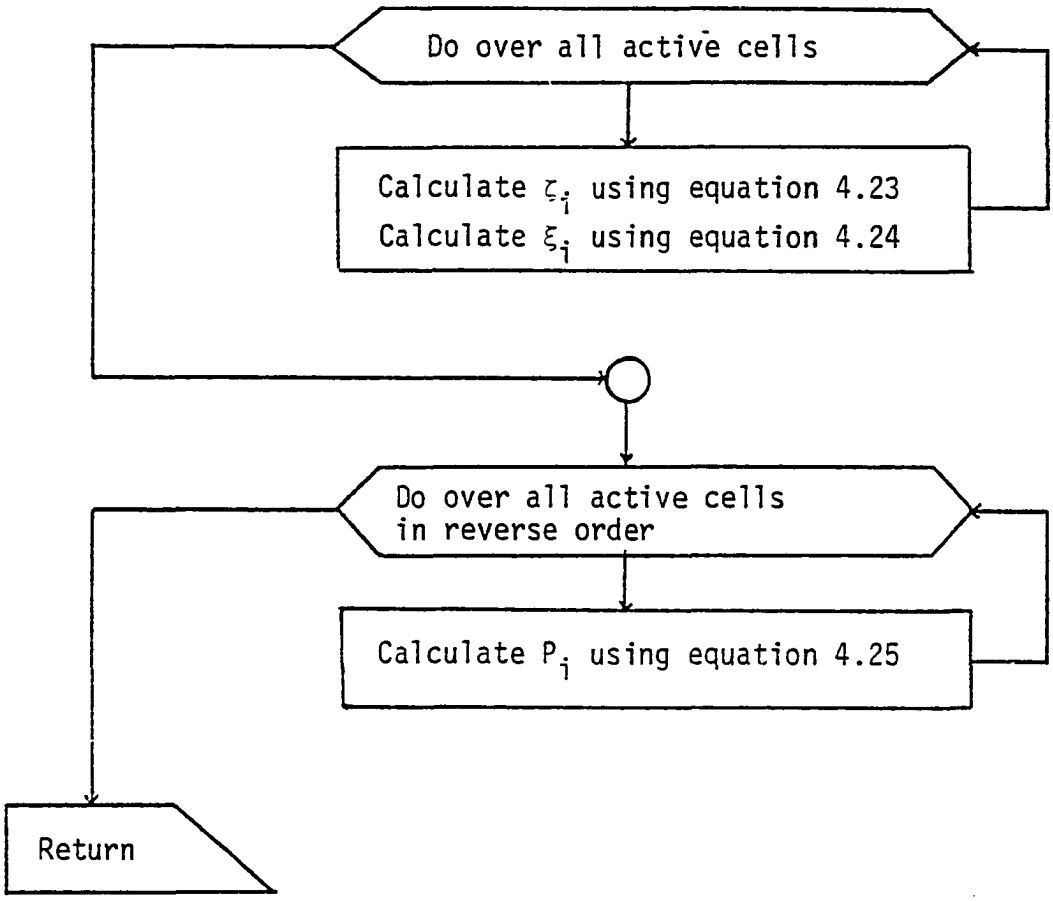


Figure 4.5 - Flow Chart for Wide-Band Thomas Algorithm

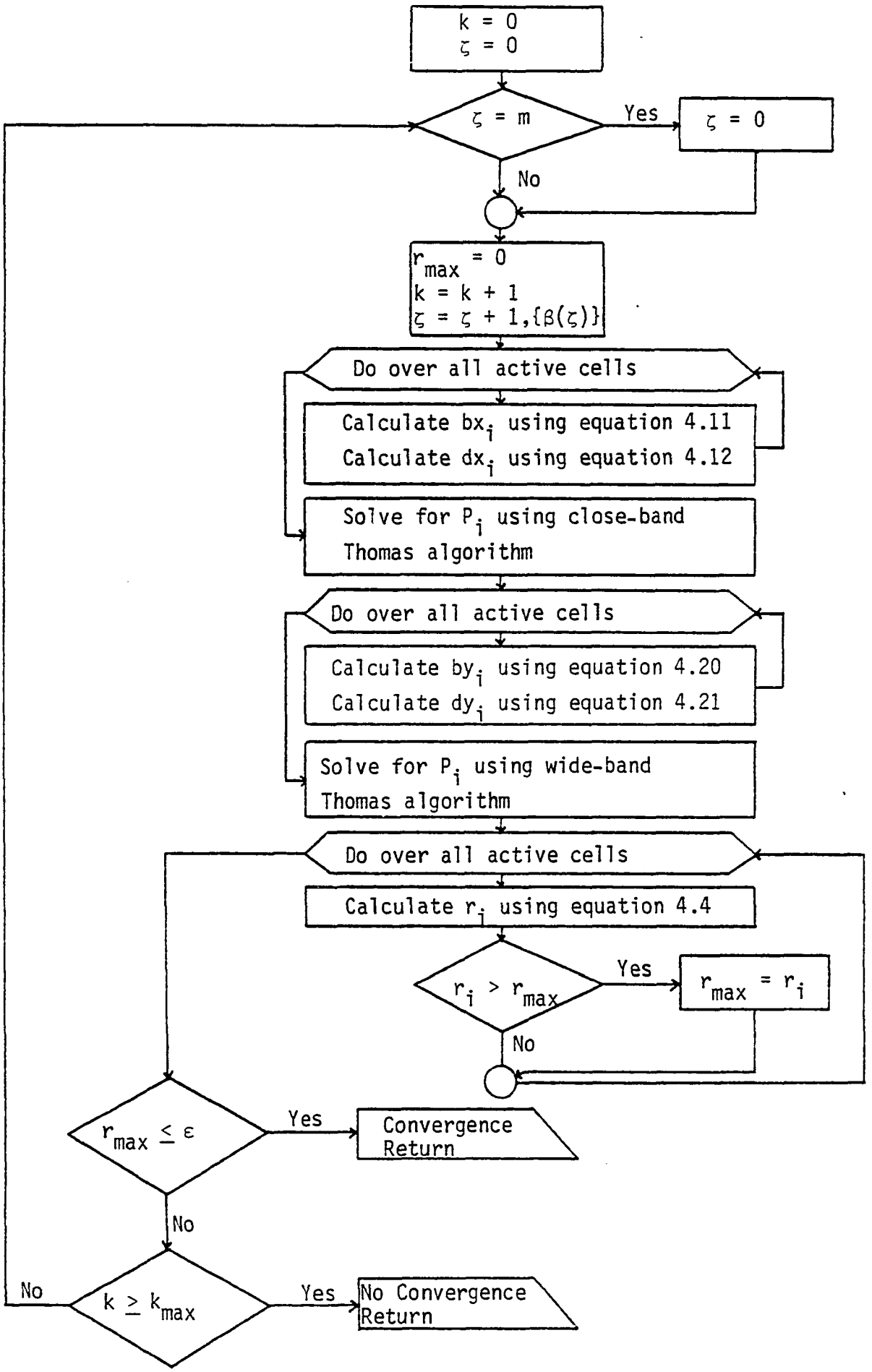


Figure 4.6 - ADI flow chart

The lower limit of the acceleration parameter is calculated from

$$\beta_{min} = \min(M_1, M_3) \tag{4.32}$$

and the upper limit is calculated from

$$\beta_{max} = \max(M_2, M_4) \tag{4.33}$$

Several authors[16,17] have pointed out that the acceleration parameters should be in a geometric series such that

$$\beta_{k+1} = r\beta_k \tag{4.34}$$

where r is the constant multiplier defined by

$$r = (\beta_{max}/\beta_{min})^{\frac{1}{m-1}} \tag{4.35}$$

where m is the number of acceleration parameters in the system.

These equations are only a guide and the user must experiment in order to obtain an adequate lower value for β. Since the convergence is very sensitive to β_{min}.

4.3 Strongly Implicit Procedure (SIP)

The strongly implicit procedure is an iterative technique developed by Stone[14]. SIP involves the solution of the system of simultaneous linear equations by an elimination process working on a modified version of the original matrix system. The original pentadiagonal matrix $\mathbb{A}$ is

transformed to a heptadiagonal matrix B which is more easily factorable into an LU product. The L and U matrices have only three nonzero elements in each row in contrary to the factorization of matrix $\mathbb{A}$. All SIP associated matrices are shown in Figure 4.8 through 4.12. The SIP flow chart is given in Figure 4.13.

The elements of L and U cannot be selected in such a way that the original $\mathbb{A}$ matrix is identical with matrix B. Two matrices are equal if their corresponding elements are equal. This would lead to the following set of relations for each cell point i:

$$e_i' = e_i \tag{4.36}$$

$$g_i' = 0 \tag{4.37}$$

$$a_i' = a_i \tag{4.38}$$

$$b_i' = b_i \tag{4.39}$$

$$c_i' = c_i \tag{4.40}$$

$$h_i' = 0 \tag{4.41}$$

$$f_i' = f_i \tag{4.42}$$

The elements of B are defined in Figure (4.10) and (4.12). The above seven relationships (equation 4.36 through 4.42) cannot all be satisfied. Hence, the transformation of $\mathbb{A}$ to B involves the additional of two new elements in equation (4.1). These two elements are associated with the cells $(i-\eta+1)$ and $(i+\eta-1)$whose coefficients we shall call g_i and h_i respectively. These associated cells are shown in Figure 4.7.

	$i-\eta$	$i-\eta+1$
$i-1$	i	$i+1$
$i+\eta-1$	$i+\eta$	

Figure 4.7 - Cell Arrangement

The new coefficients are multiplied by ε_1 and ε_2, which are obtained by Taylor's series expansions.

$$\varepsilon_1 = P_{i-\eta+1} - \alpha(-P_i + P_{i-\eta} + P_{i+1}) \tag{4.43}$$

$$\varepsilon_2 = P_{i+\eta-1} - \alpha(-P_i + P_{i-1} + P_{i+\eta}) \tag{4.44}$$

where α is the iteration parameter which accelerates the convergence process. The influence of the new coefficients is minimal since ε_1 and ε_2 are very small.

The addition of the new elements to equation (4.1) results in

$$\begin{aligned} & e_i P_{i-\eta} + a_i P_{i-1} + b_i P_i + c_i P_{i+1} + f_i P_{i+\eta} \\ & + g_i \{P_{i-\eta+1} - \alpha(-P_i + P_{i-\eta} + P_{i+1})\} \\ & + h_i \{P_{i+\eta-1} - \alpha(-P_i + P_{i-1} + P_{i+\eta})\} = d_i \end{aligned} \tag{4.45}$$

or

$$\begin{aligned} & (e_i - \alpha g_i) P_{i-\eta} + g_i P_{i-\eta+1} + (a_i - \alpha h_i) P_{i-1} \\ & + (b_i + \alpha g_i + \alpha h_i) P_i + (c_i - \alpha g_i) P_{i+1} \\ & + h_i P_{i+\eta-1} + (f_i - \alpha h_i) P_{i+\eta} = d_i \end{aligned} \tag{4.46}$$

The modified coefficient matrix $\mathbb{A}$ is shown in Figure 4.8.

The strongly implicit procedure consists of an algorithm for odd-numbered iterations and another for even-numbered iterations. Suarez and Farouq Ali[15] have presented a comparison among the three versions of SIP method. They concluded that the two equations of SIP (odd and even) when used give the best results. Hence, a combination of odd and even algorithms will be used in this study.

Odd-Numbered Iteration Algorithm

In odd-numbered iterations, the L and U matrices are chosen as shown in Figure 4.9. With this choice, the coefficients of matrix B(=LU) would be as shown in Figure 4.10. Furthermore, the modified heptadiagonal $\mathbb{A}$ matrix is identical to B. This is done by the following relations for each cell i:

$$E_i = e_i - \alpha g_i \tag{4.47}$$

$$E_i C_{i-\eta} = g_i \tag{4.48}$$

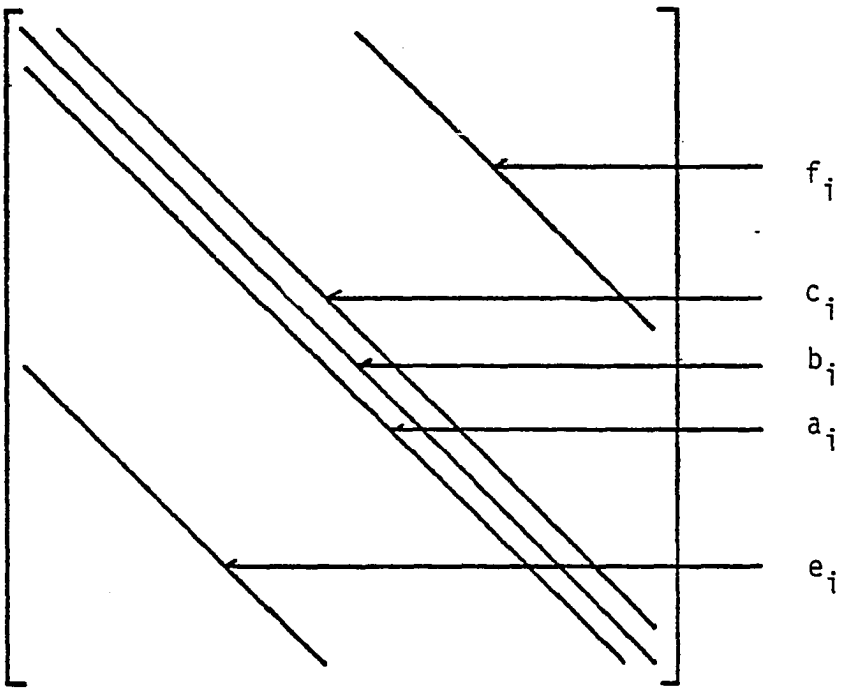
$$A_i = a_i - \alpha h_i \tag{4.49}$$

$$B_i + E_i F_{i-\eta} + A_i C_{i-1} = b_i + \alpha g_i + \alpha h_i \tag{4.50}$$

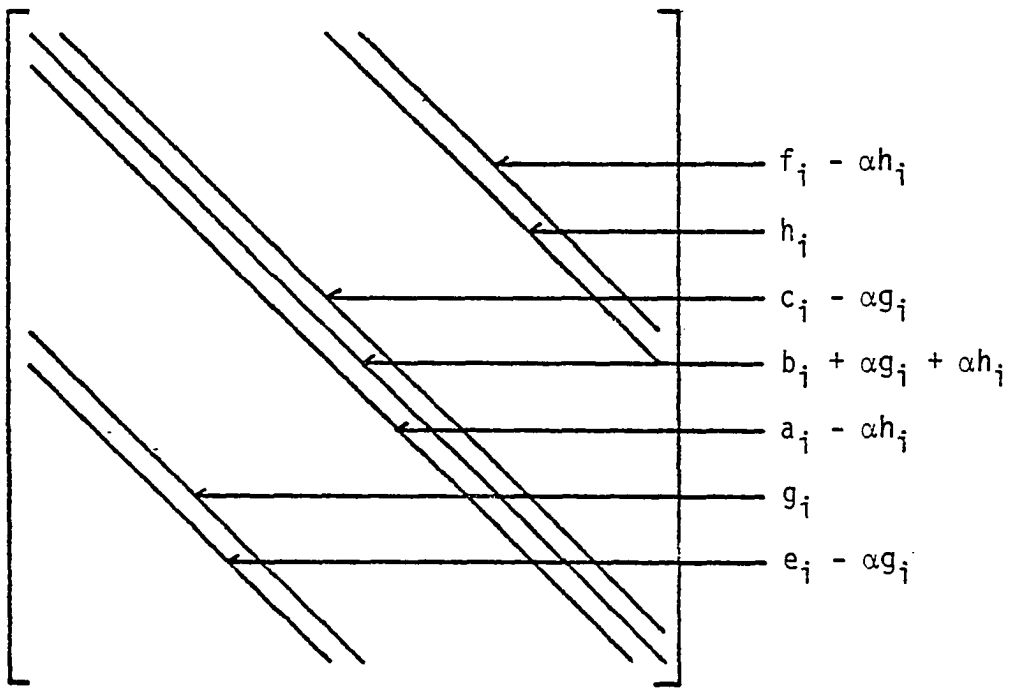
$$B_i C_i = c_i - \alpha g_i \tag{4.51}$$

$$A_i F_{i-1} = h_i \tag{4.52}$$

$$B_i F_i = f_i - \alpha h_i \tag{4.53}$$

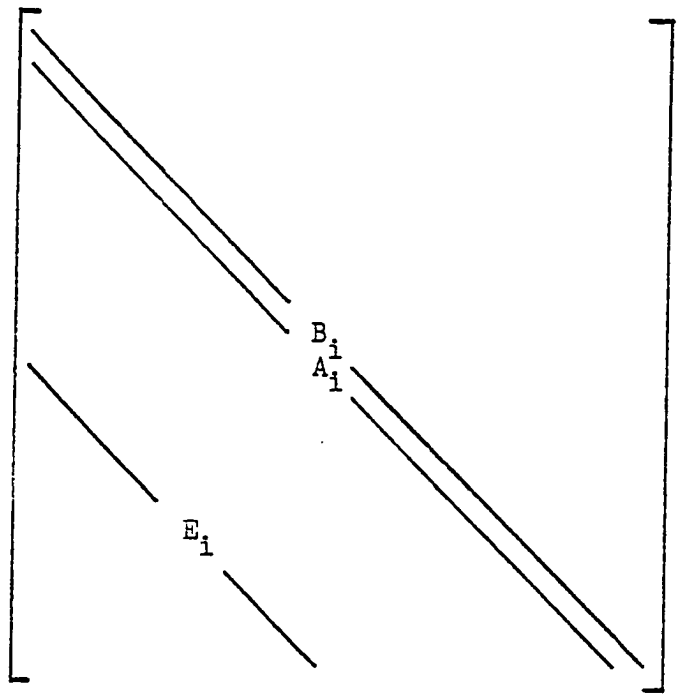


Original $\mathbb{A}$ matrix

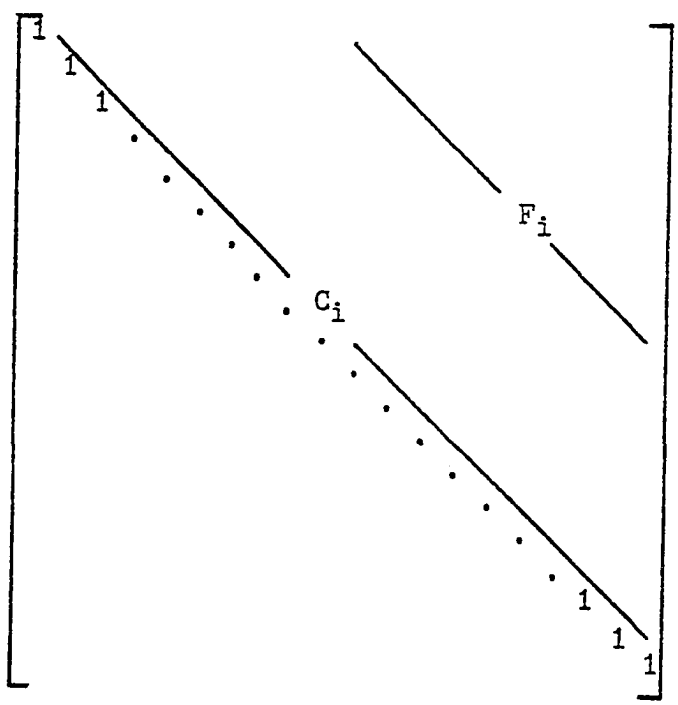


Modified $\mathbb{A}$ matrix

Figure 4.8 - Matrix $\mathbb{A}$ in SIP

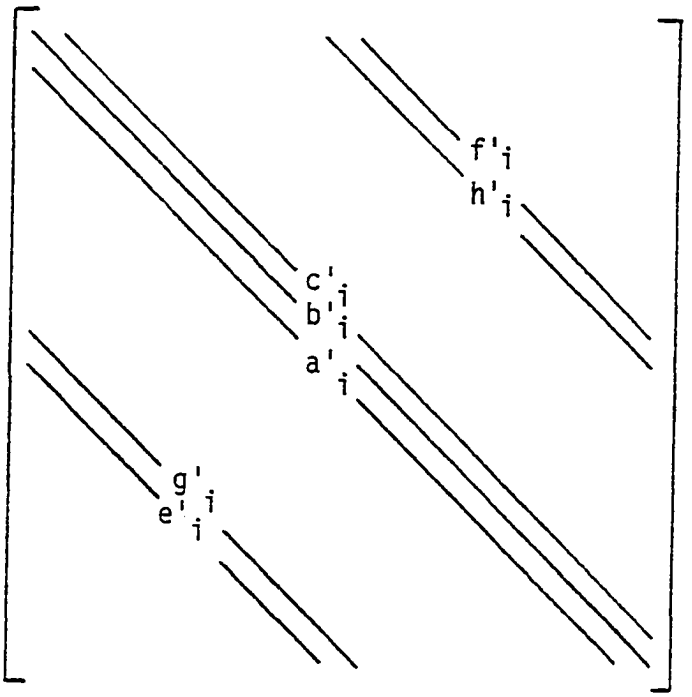


L - Matrix



U - Matrix

Figure 4.9 - L and U matrices for odd-numbered iterations



$$e'_i = E_i$$

$$g'_i = E_i C_{i-\eta}$$

$$a'_i = A_i$$

$$b'_i = B_i + E_i F_{i-\eta} + A_i C_{i-1}$$

$$c'_i = B_i C_i$$

$$h'_i = A_i F_{i-1}$$

$$f'_i = B_i F_i$$

Figure 4.10 - B matrix for odd-numbered iterations
(Coefficients in terms of L and U elements)

The algorithm of the odd-numbered iterations is as follows:

Given

$$B\underline{P} = \underline{D} \tag{4.54}$$

where

$$B = LU \tag{4.55}$$

Therefore

$$LU\underline{P} = \underline{D} \tag{4.56}$$

Letting

$$U\underline{P} = \underline{y} \tag{4.57}$$

Then

$$L\underline{y} = \underline{D} \tag{4.58}$$

Solving equations (4.58) and (4.57) successively for $\underline{y}$ and $\underline{P}$ respectively:

$$\underline{y} = L^{-1}\underline{D} \tag{4.59}$$

and

$$\underline{P} = U^{-1}\underline{y} \tag{4.60}$$

Equations (4.47) through (4.53) can be used to compute the elements of L and U matrices recursively for i=1,2,...,N as follows:

$$g_i = \frac{e_i C_{i-\eta}}{\alpha C_{i-\eta}^{+1}} \tag{4.61}$$

$$h_i = \frac{a_i F_{i-1}}{\alpha F_{i-1}^{+1}} \tag{4.62}$$

$$E_i = e_i - \alpha g_i \tag{4.63}$$

$$A_i = a_i - \alpha h_i \tag{4.64}$$

$$B_i = b_i + \alpha g_i + \alpha h_i - E_i F_{i-\eta} - A_i C_{i-1} \quad (4.65)$$

$$C_i = (c_i - \alpha g_i)/B_i \quad (4.66)$$

$$F_i = (f_i - \alpha h_i)/B_i \quad (4.67)$$

The intermediate vector $\underline{y}$ is computed by forward substitution.

$$y_i = (d_i - E_i y_{i-\eta} - A_i y_{i-1})/B_i \quad , i = 1,2,\ldots,N \quad (4.68)$$

The pressure vector is calculated by backward substitution.

$$P_i = y_i - C_i P_{i+1} - F_i P_{i+\eta} \quad , i = N, N-1,\ldots,1 \quad (4.69)$$

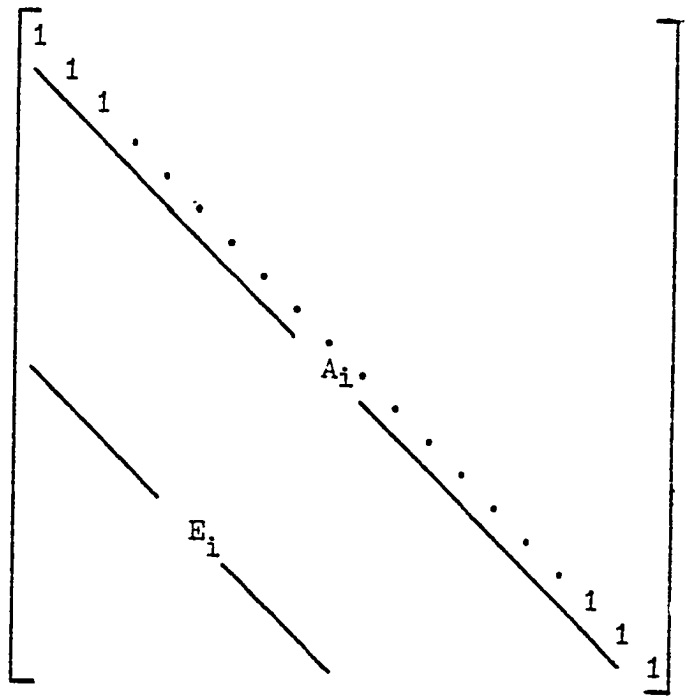
Even-Numbered Iteration Algorithm

In even-numbered iterations, the L and U matrices are chosen as shown in Figure 4.11. Whith this choice, the coefficients of matrix B(=UL) would be as shown in Figure 4.12. Furthermore, the modified $\underline{A}$ matrix is identical to B. This is done by the following relations for each cell i:

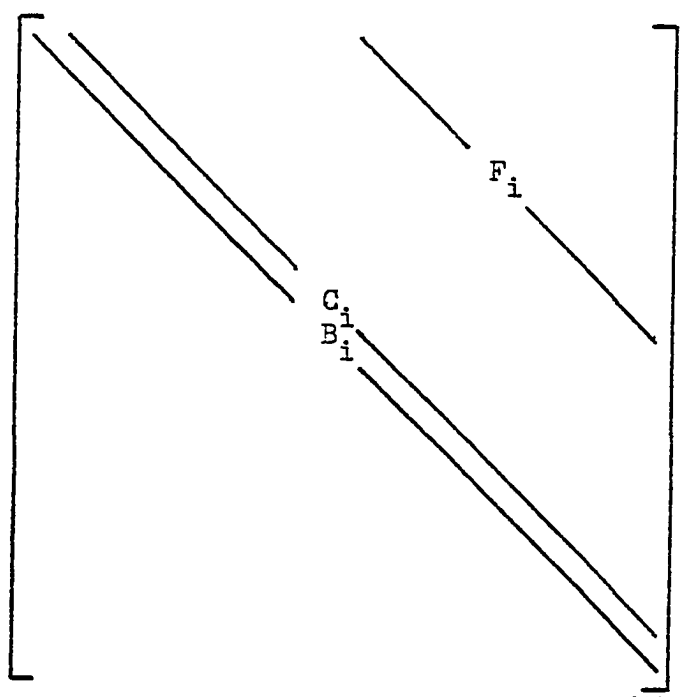
$$E_i B_i = e_i - \alpha g_i \quad (4.70)$$

$$C_i E_{i+1} = g_i \quad (4.71)$$

$$A_i B_i = a_i - \alpha h_i \quad (4.72)$$

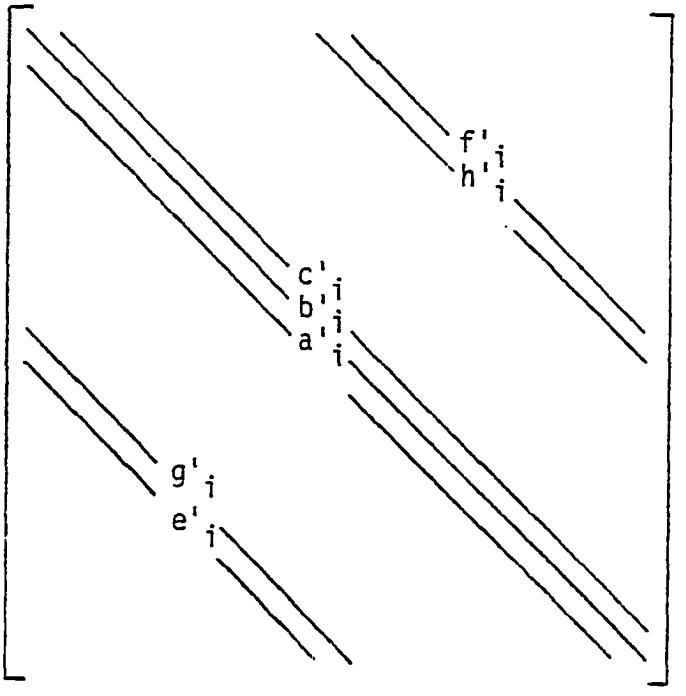


L - Matrix



U - Matrix

Figure 4.11 - L and U matrices for even-numbered iterations



$$e'_i = E_i \quad B_i$$

$$g'_i = C_i \quad E_{i+1}$$

$$a'_i = A_i \quad B_i$$

$$b'_i = B_i + C_i \quad A_{i+1} + F_i \quad E_{i+\eta}$$

$$c'_i = C_i$$

$$h'_i = F_i \quad A_{i+\eta}$$

$$f'_i = F_i$$

Figure 4.12 - B matrix for even-numbered iterations
(Coefficients in terms of L and U elements)

$$B_i + C_i A_{i+1} + F_i E_{i+\eta} = b_i + \alpha g_i + \alpha h_i \tag{4.73}$$

$$C_i = c_i - \alpha g_i \tag{4.74}$$

$$F_i A_{i+\eta} = h_i \tag{4.75}$$

$$F_i = f_i - \alpha h_i \tag{4.76}$$

The algorithm of the even-numbered iterations is as follows:

Given

$$B\underline{P} = \underline{D} \tag{4.77}$$

Where

$$B = UL \tag{4.78}$$

Therefore

$$UL\underline{P} = \underline{D} \tag{4.79}$$

Letting

$$L\underline{P} = \underline{y} \tag{4.80}$$

Then

$$U\underline{y} = \underline{D} \tag{4.81}$$

Solving equations (4.81) and (4.80) successively for $\underline{y}$ and $\underline{P}$ respectively:

$$\underline{y} = U^{-1}\underline{D} \tag{4.82}$$

and

$$\underline{P} = L^{-1}\underline{y} \tag{4.83}$$

Equations (4.70) through (4.76) can be used to compute the elements of L and U matrices recursively for i=N, N-1,...,1 as follows:

$$g_i = \frac{c_i E_{i+1}}{\alpha E_{i+1}+1} \tag{4.84}$$

$$h_i = \frac{f_i A_{i+\eta}}{\alpha A_{i+\eta}+1} \tag{4.85}$$

$$F_i = f_i - \alpha h_i \tag{4.86}$$

$$C_i = c_i - \alpha g_i \tag{4.87}$$

$$B_i = b_i + \alpha g_i + \alpha h_i - C_i\, A_{i+1} - F_i\, E_{i+\eta} \tag{4.88}$$

$$A_i = (a_i - \alpha h_i)/B_i \tag{4.89}$$

$$E_i = (e_i - \alpha g_i)/B_i \tag{4.90}$$

The intermediate vector $\underline{y}$ is computed by backward substitution.

$$y_i = (d_i - C_i\, y_{i+1} - F_i\, y_{i+\eta})/B_i \quad , \; i=N,N-1,\ldots,1 \tag{4.91}$$

The pressure vector is calculated by forward substitution.

$$P_i = y_i - E_i\, P_{i-\eta} - A_i\, P_{i-1} \quad , \; i = 1,2,..,N \tag{4.92}$$

Iteration Parameter α

The minimum parameter used is zero and the maximum (α_{max}) is determined by[20]

$$\alpha_{max} = 1 - \frac{1}{N} \sum_{i=1}^{N} \min \left[\frac{\pi^2}{2N_x^2 \ (1+ \zeta)} \ , \ \frac{\pi^2}{2N_y^2 \ (1 + 1/\zeta)} \right] \qquad (4.93)$$

where

$$\zeta = \frac{k_y \Delta x^2}{k_x \ \Delta y^2}$$

Stone [14] concluded that a minimum of four parameters is desirable and each parameter is to be used twice in succession for odd and even numbered iterations respectively. Also, the individual parameters should be geometrically spaced as follows:

$$\alpha_i = 1 - (1 - \alpha_{max})^{\frac{i-1}{m-1}} \qquad , i = 1,2,\ldots,m \qquad (4.94)$$

where m is the number of parameters in the system.

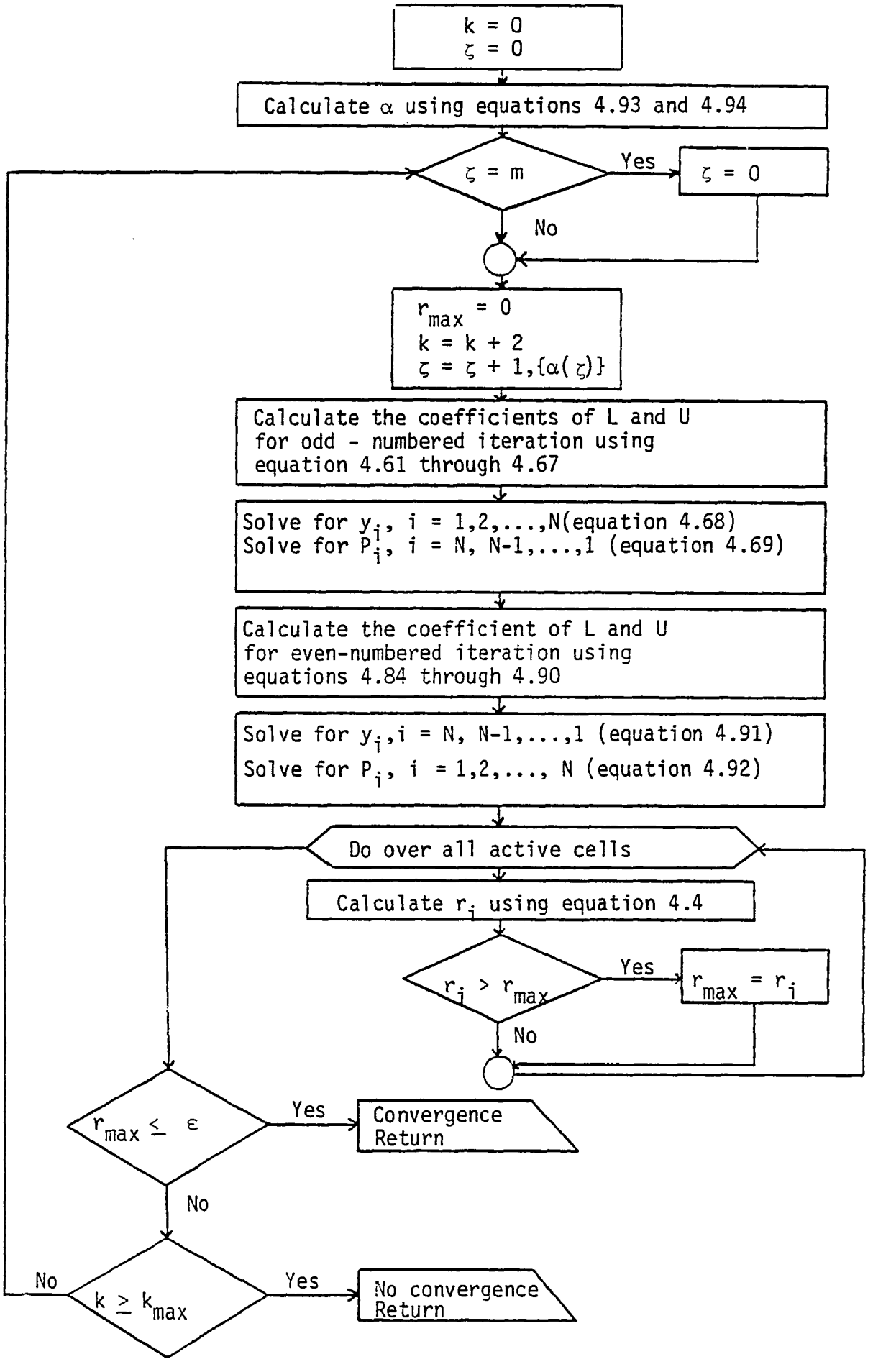


Figure 4.13 - SIP flow chart

CHAPTER V

DIRECT METHODS

Direct methods are those in which the solution to the system of equations is obtained upon the completion of a finite number of arithmetic operations. All direct methods require the same number of arithmetic operations[21], but the most popular direct method used in reservoir simulation is the LU factorization. The LU factorization method will be outlined to solve the following system of equations (see Figure 5.1):

$$\underline{\underline{A}}\,\underline{P} = \underline{d} \tag{5.1}$$

Thereafter, the band and sparsity matrix techniques will be reviewed.

5.1 LU Factorization

This algorithm has been developed by Crout[7] and it involves the decomposition of the coefficient matrix into a lower triangular matrix and an upper triangular matrix. This decomposition is followed by two consecutive substitution steps to compute the answer. The scheme of this technique is as follows:

Given

$$\underline{\underline{A}}\,\underline{P} = \underline{d}$$

then

$$\underline{\underline{A}} = LU \tag{5.2}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{1N} \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{2N} \\ \cdot & \cdot & \cdot & & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & & & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & & & & \cdot \\ \cdot & \cdot & \cdot & & & & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & & & & & \cdot & \cdot \\ a_{N1} & \cdot & \cdot & & & & & & & a_{NN} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{bmatrix}$$

Figure 5.1 - $\underline{\underline{A}}\ \underline{P} = \underline{d}$ System of Equations

The L and U matrices are given in Figure 5.2.

Therefore

$$LU\underline{P} = \underline{d} \tag{5.3}$$

Letting

$$U\underline{P} = \underline{y} \tag{5.4}$$

Then

$$L\underline{y} = \underline{d} \tag{5.5}$$

Solving equations (5.5) and (5.4) successively for $\underline{y}$ and $\underline{P}$ respectively:

$$\underline{y} = L^{-1}\underline{d} \tag{5.6}$$

and

$$\underline{P} = U^{-1}\underline{y} \tag{5.7}$$

Hence, the LU factorization technique consists of two major steps, the computation of L and U and the back substitution.

Computation of L and U

$$l_{ij} = 0 \quad \text{if} \quad i < j \tag{5.8}$$

$$u_{ij} = 0 \quad \text{if} \quad i = j \tag{5.9}$$

$$u_{ij} = 0 \quad \text{if} \quad i > j \tag{5.10}$$

for $i = 1,2,\ldots, N$ and $j = 1,2,\ldots, N$

$$l_{i1} = a_{i1} \quad , i = 1,2,\ldots,N \tag{5.11}$$

$$u_{1j} = a_{1j}/l_{11} \quad , j = 2,3,\ldots,N \tag{5.12}$$

$$
\begin{bmatrix}
l_{11} & & & & & & & & & \\
l_{21} & l_{22} & & & & & & & & \\
l_{31} & l_{32} & l_{33} & & & & & & & \\
\cdot & \cdot & \cdot & \cdot & & & & & & \\
\cdot & \cdot & \cdot & & \cdot & & & & & \\
\cdot & \cdot & \cdot & & & \cdot & & & & \\
\cdot & \cdot & \cdot & & & & \cdot & & & \\
\cdot & \cdot & \cdot & & & & & \cdot & & \\
\cdot & \cdot & \cdot & & & & & & \cdot & \\
l_{N1} & l_{N2} & l_{N3} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & l_{NN}
\end{bmatrix}
$$

L - Matrix

$$
\begin{bmatrix}
1 & u_{12} & u_{13} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_{1N} \\
 & 1 & u_{23} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_{2N} \\
 & & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_{3N} \\
 & & & \cdot & & & & & & \cdot \\
 & & & & \cdot & & & & & \cdot \\
 & & & & & \cdot & & & & \cdot \\
 & & & & & & \cdot & & & \cdot \\
 & & & & & & & \cdot & & \cdot \\
 & & & & & & & & \cdot & \cdot \\
 & & & & & & & & & 1
\end{bmatrix}
$$

U - Matrix

Figure 5.2 - L U Factorization of $\bar{A}$

The rest of the elements are calculated for m = 2,3,...,N as follows:

$$l_{im} = a_{im} - \sum_{k=1}^{m-1} l_{ik}u_{km} \qquad , i = m,m+1,...,N \quad (5.13)$$

Then compute successively

$$u_{mj} = \frac{1}{l_{mm}} \left(a_{mj} - \sum_{k=1}^{m-1} l_{mk}u_{kj}\right) \qquad , j = m+1,m+2,...,N \quad (5.14)$$

The matrices L and U can be designated by a composite matrix $\mathcal{C}$ such that $\mathcal{C} = L\backslash U$ meaning

$$\mathcal{C}_{ij} = l_{ij},\ i \geq j\ ;\quad \mathcal{C}_{ij} = u_{ij},\ i < j \qquad (5.15)$$

Back Substitution

$$y_1 = d_1/l_{11} \qquad (5.16)$$

$$y_i = \frac{1}{l_{ii}} \left(d_i - \sum_{k=1}^{i-1} l_{ik}y_k\right) \qquad , i = 2,3,...,N \quad (5.17)$$

$$P_N = y_N \qquad (5.18)$$

$$P_i = y_i - \sum_{k=i+1}^{N} u_{ik}P_k \qquad , i = N-1,N-2,...,1 \quad (5.19)$$

5.2 Ordering Schemes

In reservoir simulators using implicit pressure explicit saturation technique, the coefficient matrix (matrix $\mathbb{A}$ in equation 3.24) is a sparse matrix. The order of the unknowns of such a sparse system of linear algebric equations drastically affects the amount of computation and storage requirements.

Price and Coats[12] presented four different ordering schemes based on optimal ordering. The work and storage involved are given as functions of the size of coefficient matrix. These ordering

schemes are standard, diagonal, alternating point, and alternating diagonal. McDonald and Trimble[9] presented a fifth ordering scheme that we will call alternating line. The work requirement for this scheme is also presented.

1. Standard Ordering: In this scheme, the cells are numbered consecutively along the rows or columns starting with the shortest direction to reduce the numerical computation. Figure 5.3 depicts such an ordering scheme in two dimensions.

The resulting coefficient matrix corresponding to standard ordering is of a pentadiagonal form as shown in Figure 5.5.

2. Diagonal Ordering: In this ordering, the cells are numbered consecutively along the diagonals starting with the shortest direction. This ordering scheme is shown in Figure 5.4.

The resulting coefficient matrix corresponding to diagonal ordering is shown in Figure 5.6.

3. Alternating Point Ordering: In this ordering, the cells are numbered such that each point is separated from its adjacent number by at least one other point.

The resulting coefficient matrix corresponding to alternating point ordering scheme of Figure 5.7 is shown in Figure 5.9.

4. Alternating Diagonal Ordering: In this scheme, the cells are numbered consecutively along alternating diagonals. This ordering scheme is shown in Figure 5.8.

Figure 5.10 shows the resulting coefficient matrix corresponding to alternating diagonal ordering scheme.

	1	2	3	4	5	6 = I
1	1	6	11	16	21	26
2	2	7	12	17	22	27
3	3	8	13	18	23	28
4	4	9	14	19	24	29
J = 5	5	10	15	20	25	30

Figure 5.3 - Standard Ordering

	1	2	3	4	5	6 = I
1	1	3	6	10	15	20
2	2	5	9	14	19	24
3	4	8	13	18	23	27
4	7	12	17	22	26	29
J = 5	11	16	21	25	28	30

Figure 5.4 - Diagonal Ordering

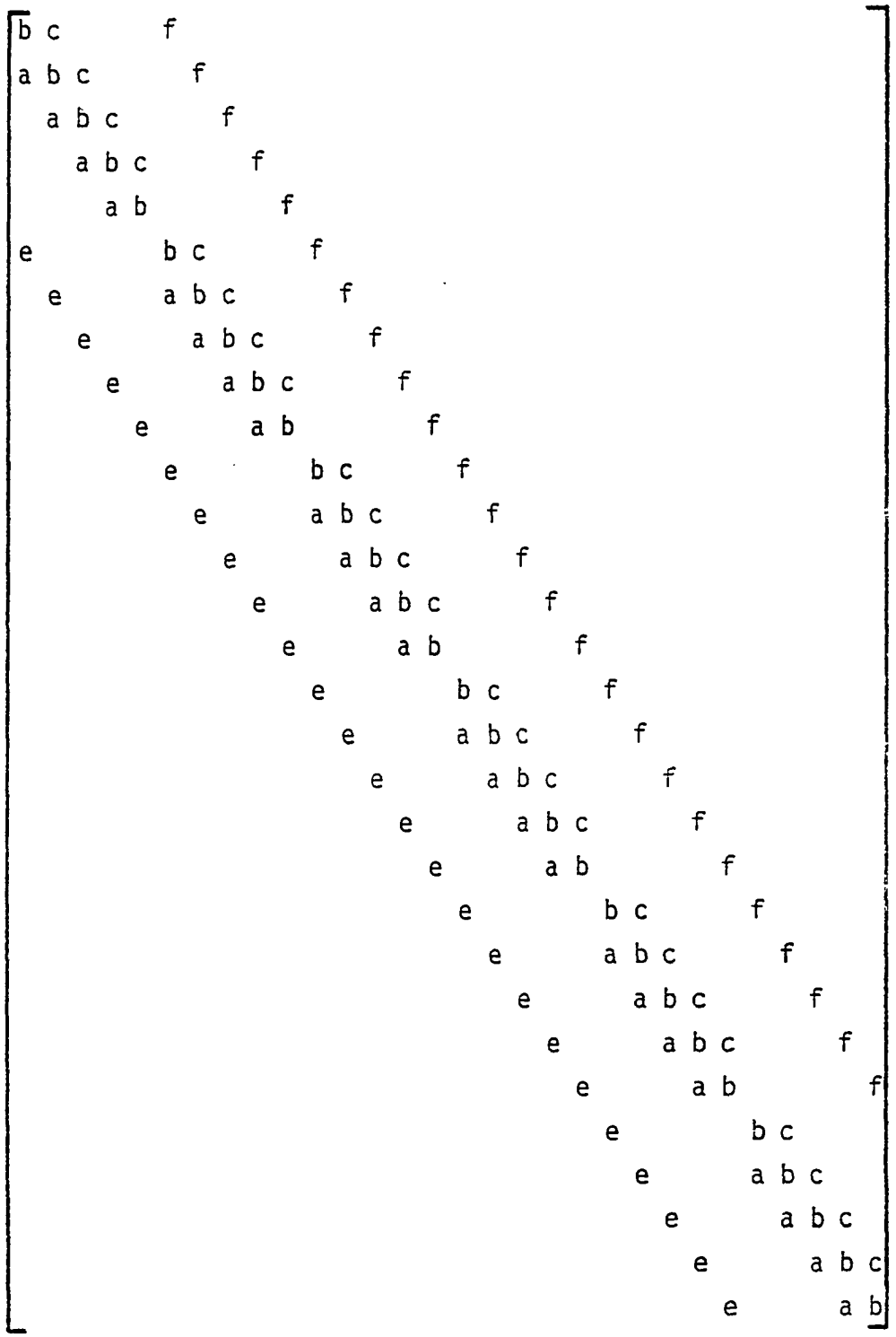


Figure 5.5-Matrix $\mathbb{A}$ Corresponding to Standard Ordering

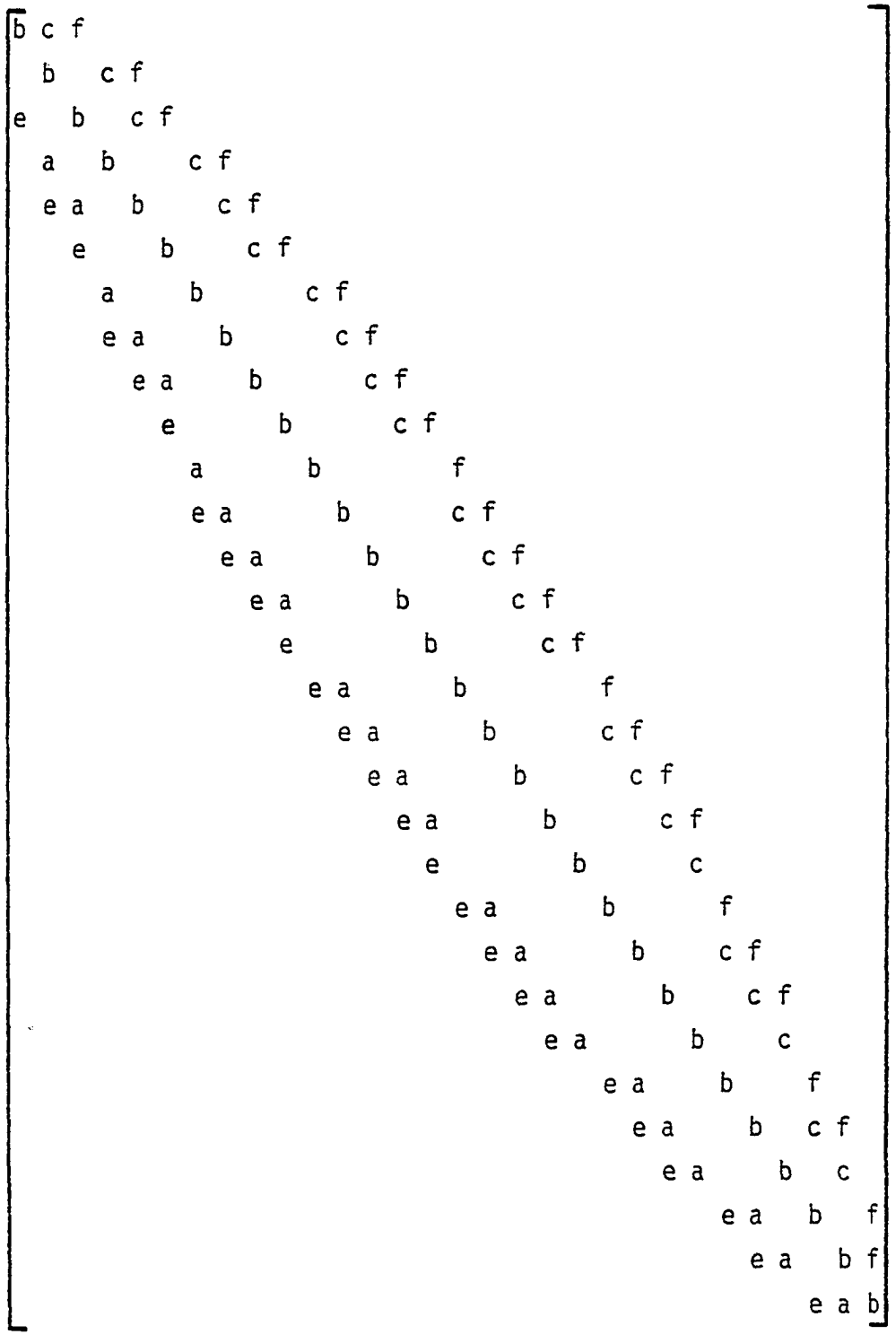


Figure 5.6-Matrix $\mathbb{A}$ Corresponding to Diagonal Ordering

	1	2	3	4	5	6	= I
1	1	16	2	17	3	18	
2	19	4	20	5	21	6	
3	7	22	8	23	9	24	
4	25	10	26	11	27	12	
J = 5	13	28	14	29	15	30	

Figure 5.7 - Alternating Point Ordering

	1	2	3	4	5	6	= I
1	1	17	4	21	9	26	
2	16	3	20	8	25	13	
3	2	19	7	24	12	29	
4	18	6	23	11	28	15	
J = 5	5	22	10	27	14	30	

Figure 5.8 - Alternating Diagonal Ordering

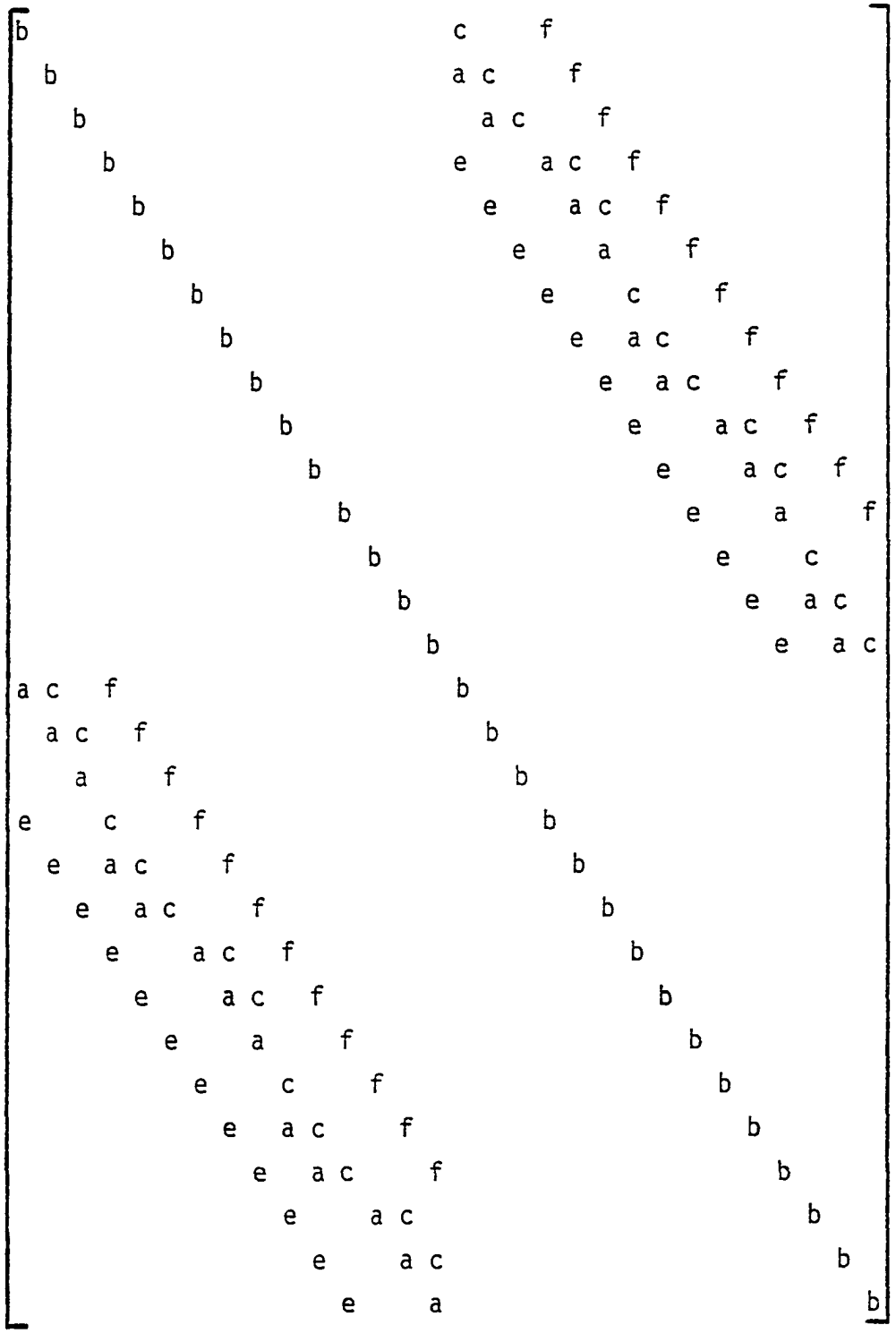


Figure 5.9-Matrix $\mathbf{A}$ Corresponding to Alternating Point Ordering

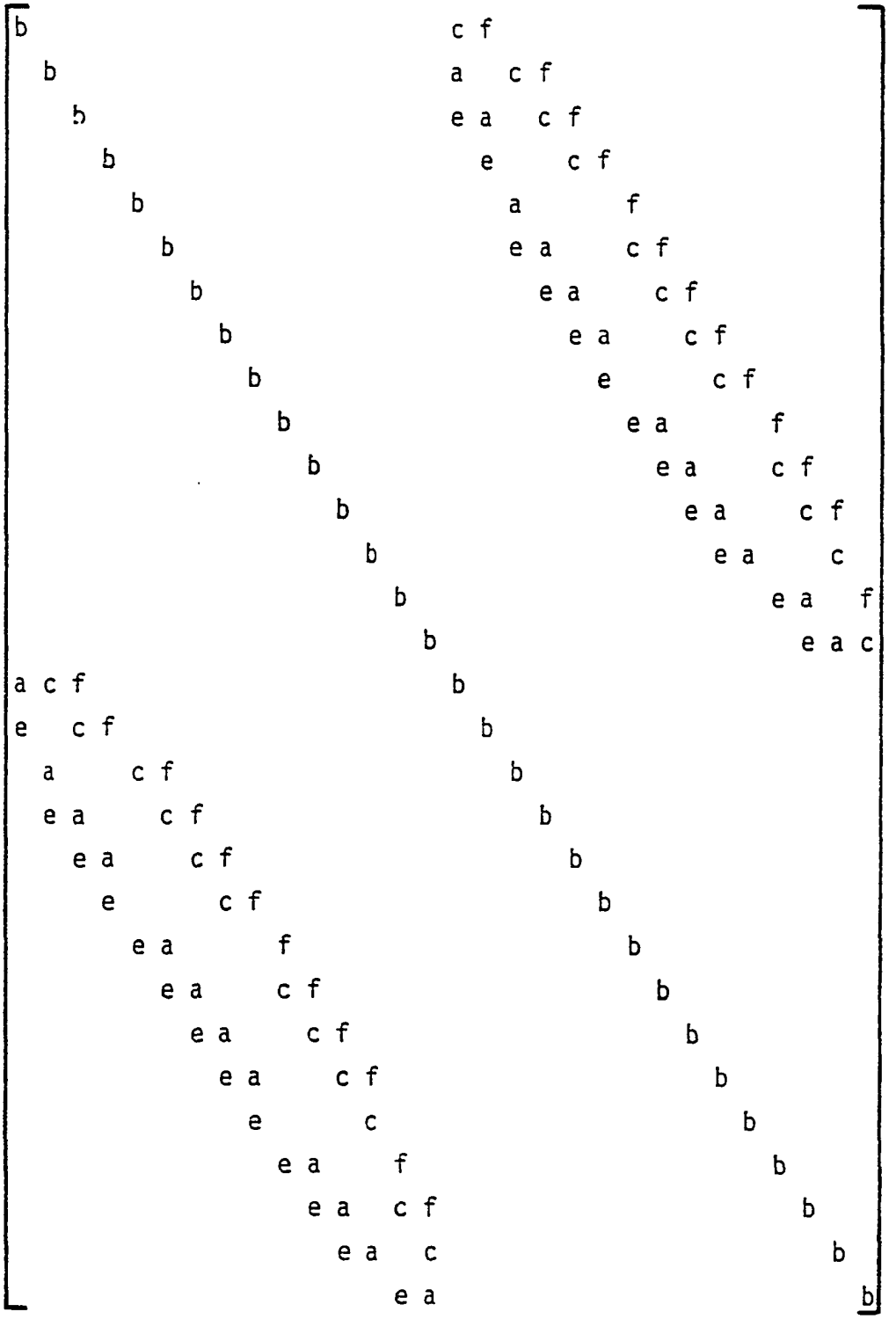


Figure 5.10-Matrix $\underline{A}$ Corresponding to Alternating Diagonal Ordering

5. Alternating Line Ordering: In this ordering, the cells are numbered consecutively along the shortest rows or colums, in an alternating pattern. Figure 5.11 depicts such an ordering scheme.

Figure 5.12 shows the resulting coefficient matrix corresponding to alternating line ordering scheme.

Here a summary of the work and storage requirements of an I x J two-dimensional mesh is presented with large I and J such that $J \leq I$. The work is defined as the number of multiplications and divisions necessary to solve for the unknowns without operating on zero elements. The storage is required only for nonzero elements.

Ordering Scheme	Work	Storage
Standard	IJ^3	IJ^2
Diagonal	$IJ^3 - \frac{J^4}{2}$	$IJ^2 - \frac{J^3}{3}$
Alternating Point	$\frac{IJ^3}{2}$	$\frac{IJ^2}{2}$
Alternating Line	$\frac{IJ^3}{2}$	
Alternating Diagonal	$\frac{IJ^3}{2} - \frac{J^4}{4}$	$\frac{IJ^2}{2} - \frac{J^3}{6}$

Table 5.1 - Work and Storage Requirements for Ordering Schemes

	1	2	3	4	5	6 = I
1	1	16	6	21	11	26
2	2	17	7	22	12	27
3	3	18	8	23	13	28
4	4	19	9	24	14	29
J = 5	5	20	10	25	15	30

Figure 5.11 - Alternating Line Ordering

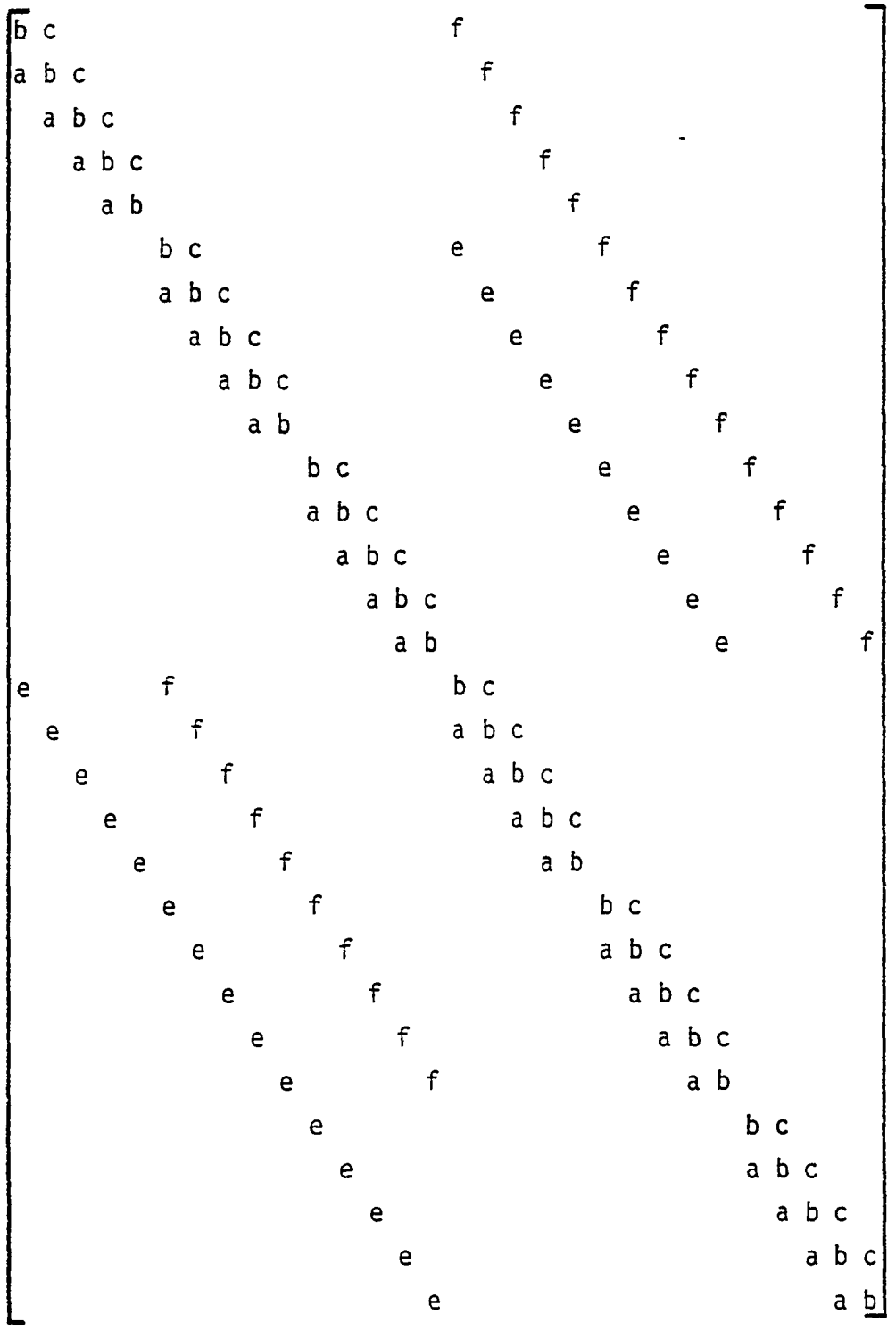


Figure 5.12-Matrix $\mathbb{A}$ Corresponding to Alternating Line Ordering

5.3 Generate - and - Solve Algorithms

The optimal ordering technique can be improved by the use of processes which are geared specifically to the direct solution. One method is the generate - and - solve algorithms used by Woo, et al[21] in reservoir simulation. Their method makes use of the computer architecture in developing the algorithm. The basis of the method is summarized by Crichlow[6] as follows:

1. In a given matrix there is a fixed number of nontrivial operations required to solve the system by some given direct scheme.
2. The solution process produces some nonzero elements which will always be present at a given (i,j) location in the matrix.
3. If the location of all nonzero quantities is fixed a priori both in the matrix and in its solution, then a straightforward, nonbranching, nonlooping computer code can be developed to solve this system very efficiently.

The major drawback of this technique is the fact that it requires large storage and overhead work to store and generate the nonlooping computer code.

5.4 Band Matrix Technique

Direct processes, although they are conceptually easy, are rarely used without extensive modification to solve the pressure system in simulation because these processes usually require large computer storage space and long computation times, unless special techniques such as band matrix or sparsity are employed.

The treatment of zeros within the bands as non-zeros is called band matrix technique. The band matrix algorithm will be applied on standard ordering which we will call STBAND.

STBAND

For band matrix technique applied on standard ordering, all elements between F-vector and E-vector in Figure 5.5 are assumed to be nonzero.

By letting

$$G_{\eta,j} = e_j \tag{5.20}$$

$$G_{1,j} = a_j \tag{5.21}$$

$$H_{1,j} = c_j \tag{5.22}$$

$$H_{\eta,j} = f_j \tag{5.23}$$

the general equation of STBAND would be

$$G_{1,j}P_{j-\eta} + G_{2,j}P_{j-\eta+1} + \cdots + G_{\eta,j}P_{j-1} + b_jP_j$$

$$+ H_{\eta,j}P_{j+1} + H_{\eta-1,j}P_{j+2} + \cdots + H_{1,j}P_{j+\eta} = d_j$$

$$1 \leq j \leq N,\ N \geq \eta \tag{5.24}$$

The algorithm used is as follows[18]:

$$\alpha_{i,j} = G_{i,j} = 0 \qquad \text{for } i \geq j \tag{5.25}$$

$$H_{i,j} = 0 \qquad \text{for } i \geq N + 1 - j \tag{5.26}$$

Forward Solution (j = 1,2,...,N)

$$\alpha_{i,j} = G_{i,j} - \sum_{k=i+1}^{k=\eta} \alpha_{k,j} y_{k-i,j-k} \quad , i = \eta, \eta-1, \ldots, 1 \tag{5.27}$$

$$\beta_j = b_j - \sum_{k=1}^{k=\eta} \alpha_{k,j} y_{k,j-k} \tag{5.28}$$

$$y_{i,j} = \frac{1}{\beta_j}\{ H_{i,j} - \sum_{k=i+1}^{k=\eta} \alpha_{k-i,j} y_{k,j-k+i} \} \quad , i = 1,2,\ldots,\eta \tag{5.29}$$

$$\gamma_j = \frac{1}{\beta_j} \{ d_j - \sum_{k=1}^{k=\eta} \alpha_{k,j} \gamma_{j-k} \} \tag{5.30}$$

Backward Solution (j = N, N-1,..., 1)

$$P_j = \gamma_j - \sum_{k=1}^{k=\eta} y_{k,j} P_{j+k} \tag{5.31}$$

CHAPTER VI

A NEW APPROACH TO THE APPLICATION OF DIRECT METHODS IN PETROLEUM RESERVOIR SIMULATION

The yet known systematic optimal ordering in petroleum reservoir simulation is alternating diagonal as shown in Table 5.1. However, there is a major drawback to this ordering scheme. In alternating diagonal scheme, the elements in the upper right and lower left corners of matrix $\mathbb{A}$ do not form a perfect secondary diagonals as shown in Figure 5.10. This problem forced us to use band algorithm which treats the zeros within the band as nonzeros. The band algorithm was discussed in the preceding chapter. The treatment of the zeros as nonzeros makes the band matrix technique require a larger storage space and more computational labor.

6.1 Restricted Alternating Diagonal Ordering

Here a modification to the alternating diagonal ordering that produces perfect secondary diagonals in the orginal $\mathbb{A}$ matrix is proposed. This ordering will be called restricted alternating diagonal (RAD). Figures 6.1 and 6.2 depict the RAD ordering. Figures 6.3 and 6.4 show the resulting coefficient matrix corresponding to RAD ordering scheme.

The numbering of this scheme must go along the shortest diagonal to reduce the band width of the new nonzero elements that will be

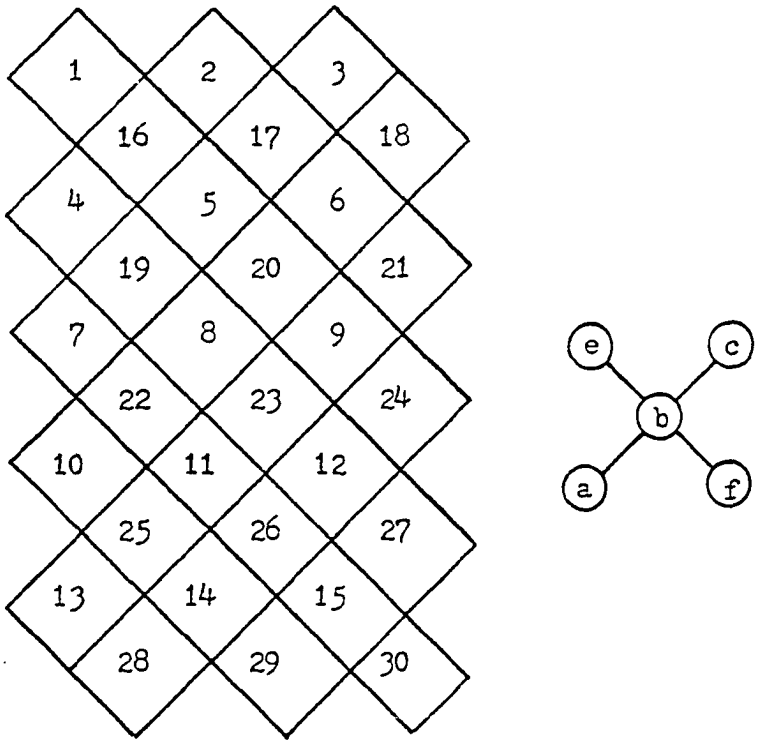


Figure 6.1 - RAD Ordering Along the Shortest Diagonal

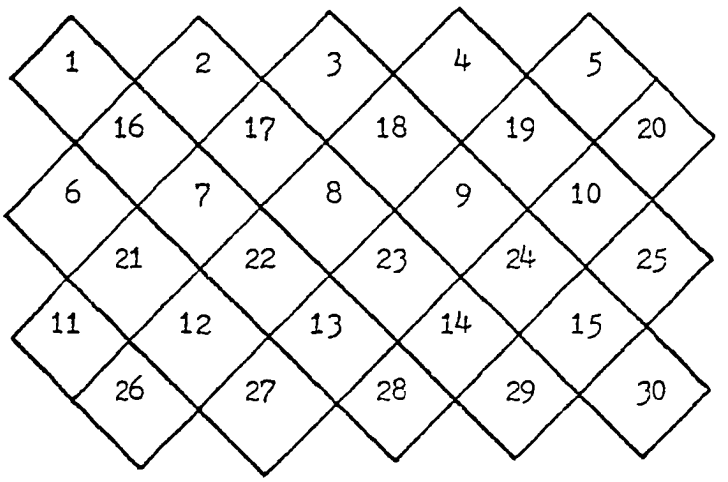


Figure 6.2 - RAD Ordering Along the Longest Diagonal

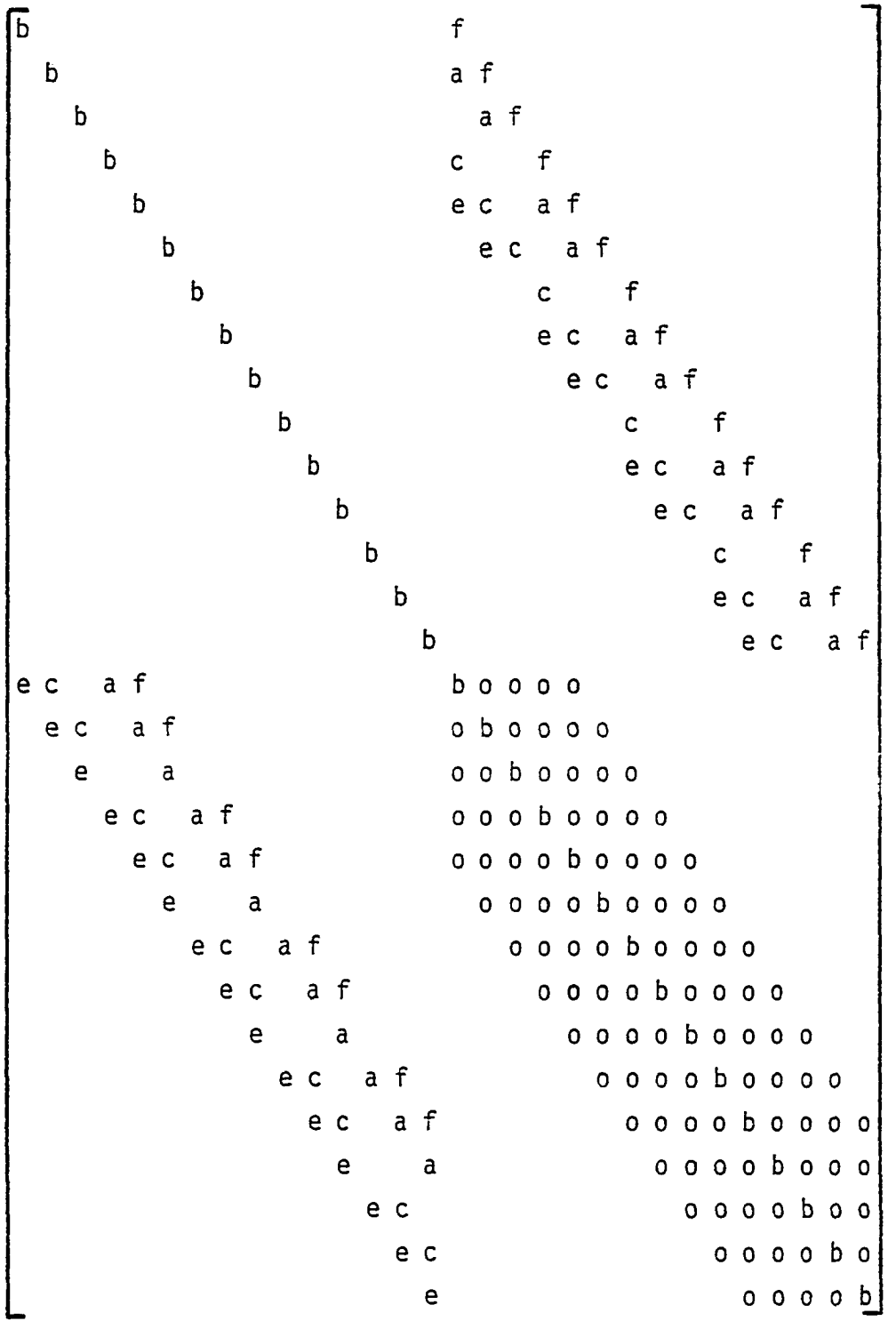


Figure 6.3-Matrix $\mathbb{A}$ Corresponding to RAD Along the Shortest Diagonal

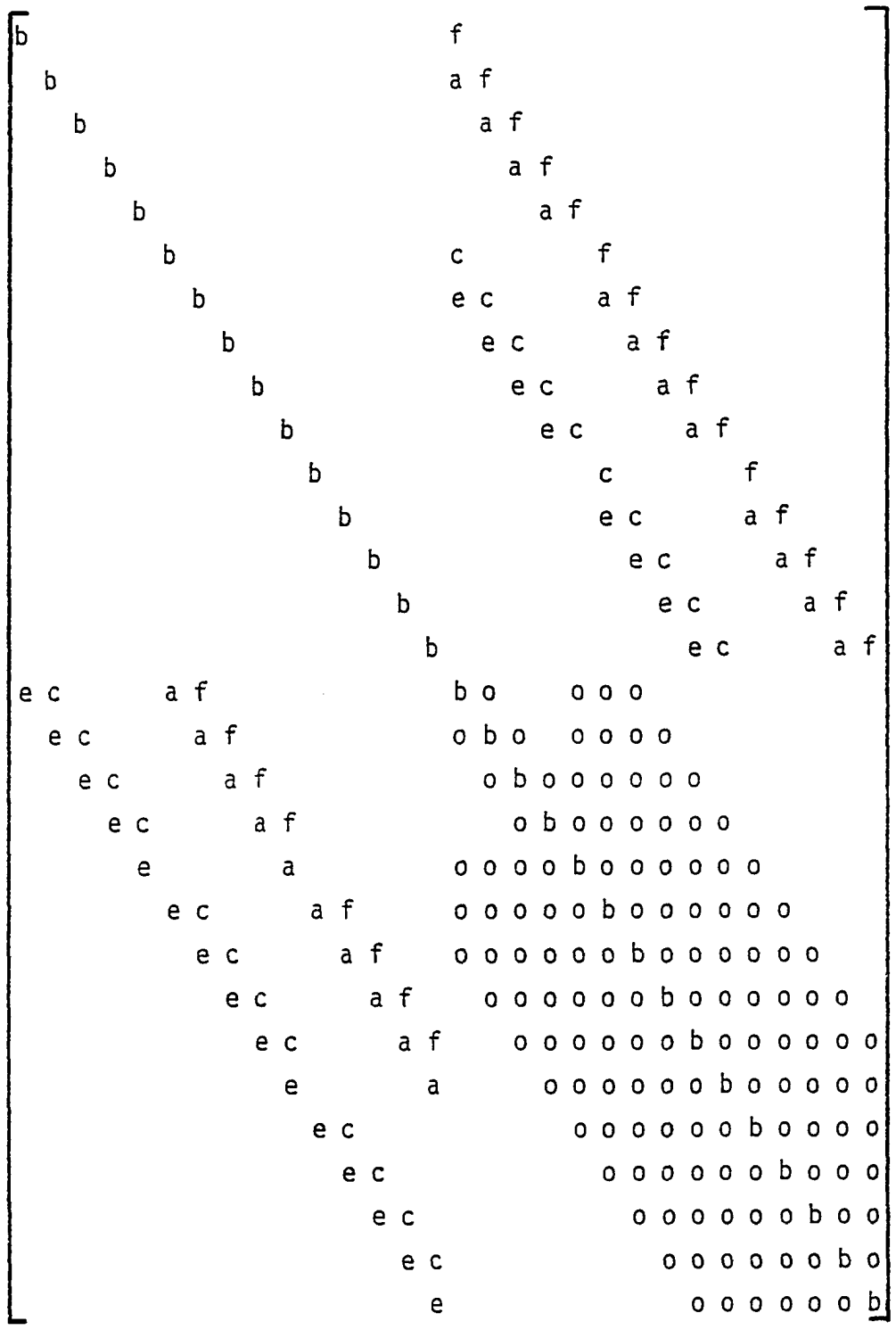


Figure 6.4 - Matrix $\underline{A}$ Corresponding to RAD Along the Longest Diagonal

generated later in the process of factorization. This will be obvious when we compare Figures 6.3 and 6.4.

6.2 LU Factorization Applied to RAD

In the process of LU factorization, no new elements will be generated except for the elements in the lower right corner of the matrix as shown in Figures 6.3 and 6.4 where the new elements are entered as small circles. The new elements form a band of width equals (2m-1), where m is the semi-bandwidth of matrix $\bar{\bar{A}}$.

The semi-bandwidth m for RAD scheme is defined as follows:

$$m = L_d + 2 \tag{6.1}$$

where L_d is the length of the diagonal along which numbering is applied.

The upper section of matrix $\bar{\bar{A}}$ and $\bar{G}$ has a length of N_u, while the length of the lower section of $\bar{\bar{A}}$ and $\bar{G}$ is N_l. These terms are defined as follows:

$$\left.\begin{aligned} N_u &= N/2 \quad , \text{ if } N_d \text{ is even} \\ N_u &= (N-m)/2 + m \quad , \text{ if } N_d \text{ is odd} \end{aligned}\right\} \tag{6.2}$$

and

$$N_l = N - N_u \tag{6.3}$$

where N_d is the number of diagonals in the system.

The work requirement, when LU factorization is applied to RAD to solve the pressure equations is

$$W \simeq 2N + 5N_u + (16 + m^2)N_l \tag{6.4}$$

and the storage requirement is

$$S < 5N + (2m-2)N_l \tag{6.5}$$

The application of LU factorization to the RAD of Figure 6.5 results in the composite matrix $\mathbf{G}$ of Figure 6.6. The RAD scheme consists of the computation of $\mathbf{G}$ and the back substitution. These two steps are outlined next.

Computation of the Composite Matrix $\mathbf{G}(=\mathbf{L}\backslash\mathbf{U})$

The elements in the left half of matrix $\mathbf{G}$ are the same as those of matrix $\mathbf{A}$.

In the process of factorization, there are no new elements introduced in the upper right corner of matrix $\mathbf{G}$, but the existing elements are modified for $j = 1,2,\ldots,N_u$ as follows:

$$a_j' = a_j/b_j \tag{6.6}$$

$$c_j' = c_j/b_j \tag{6.7}$$

$$e_j' = e_j/b_j \tag{6.8}$$

$$f_j' = f_j/b_j \tag{6.9}$$

There are 2m-2 new vectors introduced in the lower right quarter of matrix $\mathbf{G}$ in the process of factorization as shown in Figure 6.6. These vectors and the principal diagonal vector are calculated into two steps. The first step is the computation of the contribution of the mutual product of vectors A,C,E and F. There are 8 vectors out of the 2m-2 vectors which receive such contribution. This contribution is computed for $j = 1,2,\ldots,N_l$ by the following equations:

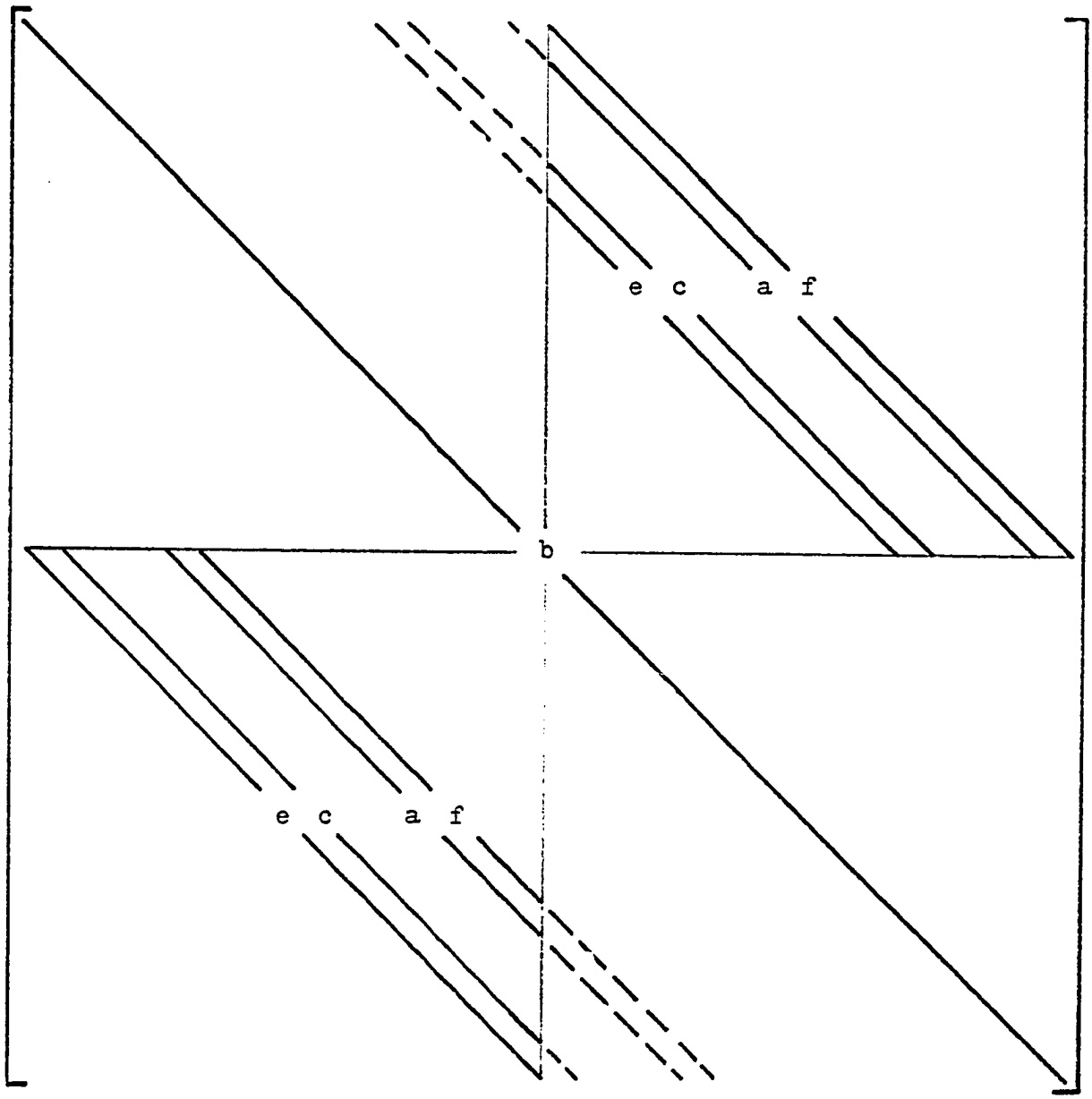


Figure 6.5 - General form of Matrix $\bar{A}$ of RAD

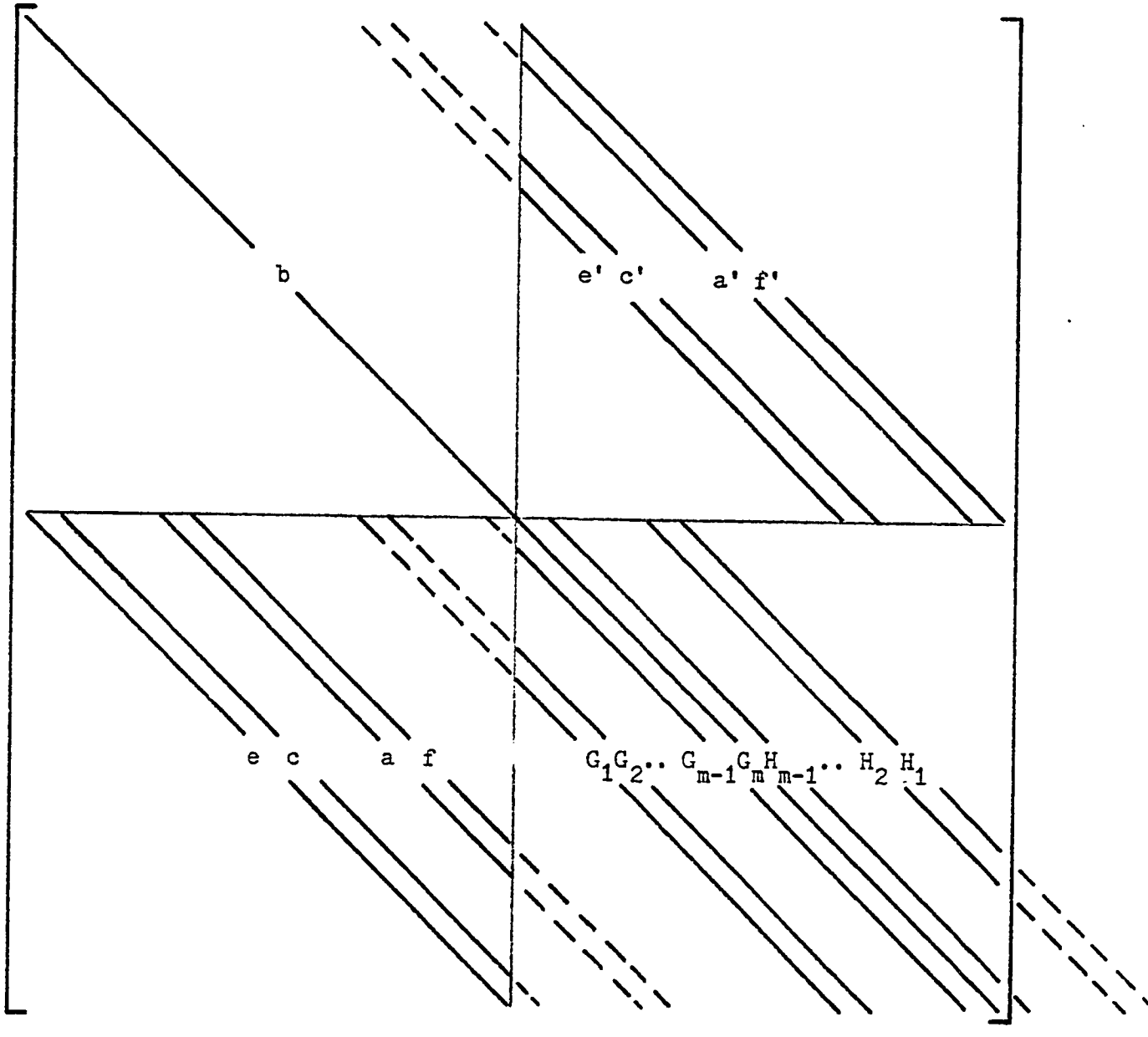


Figure 6.6 - The Composite Matrix $\mathcal{C}$ of RAD

$$G'_{1,j} = -e_{j+N_u}\, e'_j \tag{6.10}$$

$$G'_{2,j} = -e_{j+N_u}\, c'_j - c_{j+N_u}\, e'_{j+1} \tag{6.11}$$

$$G'_{3,j} = -c_{j+N_u}\, c'_{j+1} \tag{6.12}$$

$$G'_{i,j} = 0 \qquad , i = 4,5,\ldots,m-2 \tag{6.13}$$

$$G'_{m-1,j} = -e_{j+N_u}\, a'_j - a_{j+N_u}\, e'_{j+m-2} \tag{6.14}$$

$$G'_{m,j} = b_{j+N_u} - e_{j+N_u}\, f'_j - c_{j+N_u}\, a'_{j+1} - a_{j+N_u}\, c'_{j+m-2} - f_{j+N_u}\, e'_{j+m-1} \tag{6.15}$$

$$H'_{1,j} = -f'_{j+m-1}\, f_{j+N_u} \tag{6.16}$$

$$H'_{2,j} = -f'_{j+m-2}\, a_{j+N_u} - a'_{j+m-1}\, f_{j+N_u} \tag{6.17}$$

$$H'_{3,j} = -a'_{j+m-2}\, a_{j+N_u} \tag{6.18}$$

$$H'_{i,j} = 0 \qquad , i = 4,5,\ldots,m-2 \tag{6.19}$$

$$H'_{m-1,j} = -f'_{j+1}\, c_{j+N_u} - c'_{j+m-1}\, f_{j+N_u} \tag{6.20}$$

In the second step, all the computation is within the lower right quarter of matrix G. In this step the elements are subdivided into three groups as shown in Figure 6.7. These groups are chosen in such a way that the indices can be precalculated and easily programmed.

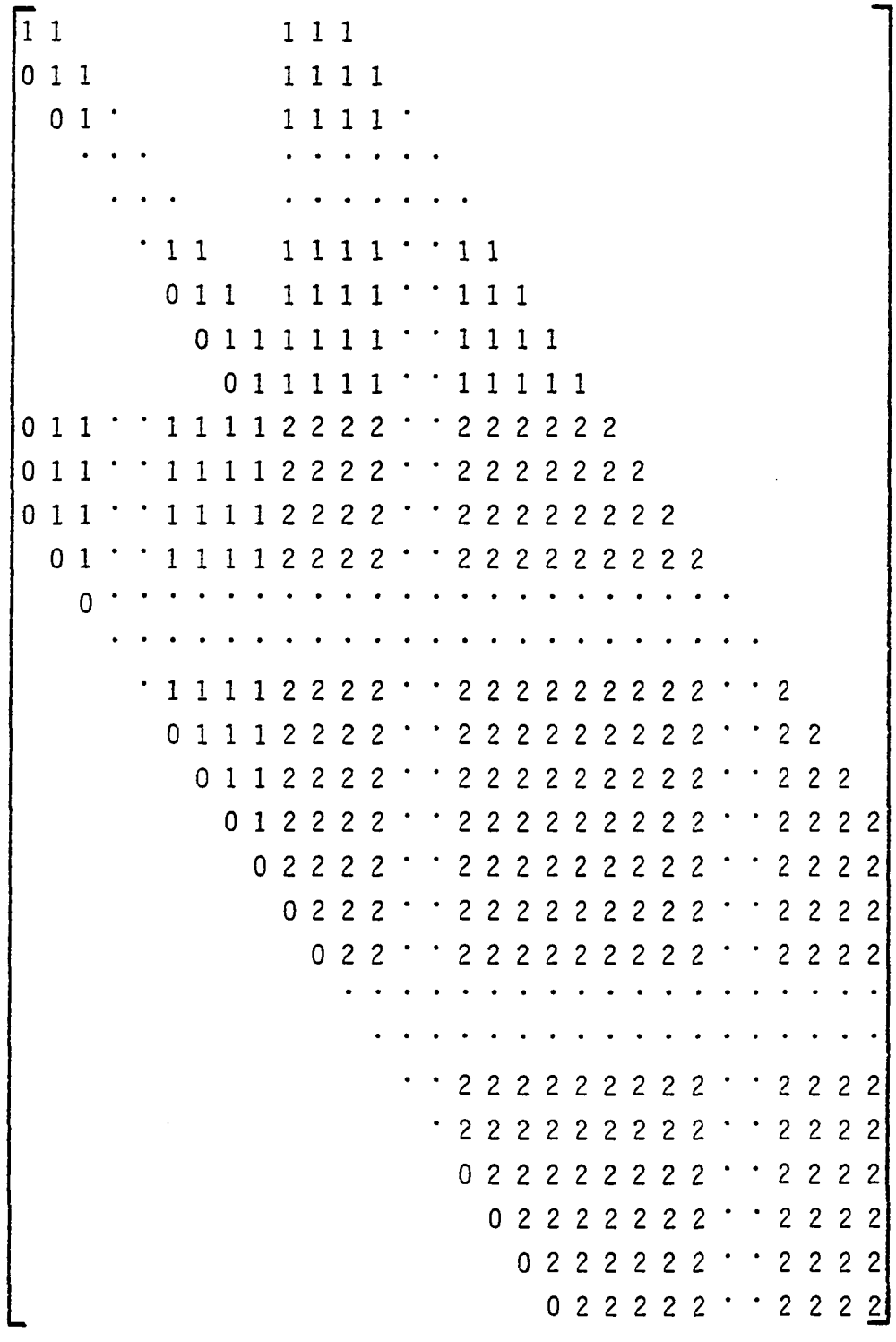


Figure 6.7 - The lower right quarter of matrix $\hat{G}$ of semi-bandwidth m
(The number indicates the group of the element)

Furthermore, this choice provides a programming procedure for this complex problem without using any if statement which results in a reduction of computer time. The computations within these groups are outlined next.

Group 0

No more computation is needed

Group 1

For $i = 1,2,3,\ldots,m$ calculate

$$H_{i,1} = H'_{i,1} / G_{m,1} \tag{6.21}$$

Then calculate for $j = 2,3,\ldots,m-3$

$$G_{m,j} = G'_{m,j} - G_{m-1,j} H_{m-1,j-1} \tag{6.22}$$

$$H_{i,j} = \frac{1}{G_{m,j}} (H'_{i,j} - H_{i-1,j-1} G_{m-1,j}) \quad , i=2,3,..,j+2 \tag{6.23}$$

$$H_{m-1,j} = H'_{m-1,j} / G_{m,j} \tag{6.24}$$

Group 2

For $j = m-2,m-1,\ldots, N_l - 1$ calculate

$$G_{i,j} = G'_{i,j} - \sum_{k=1}^{i-1} G_{i-k,j} H_{m-k,j-k-m+i} \quad , i=2,3,\ldots,m \tag{6.25}$$

$$H_{1,j} = H'_{1,j} / G_{m,j} \tag{6.26}$$

$$H_{i,j} = \frac{1}{G_{m,j}} \left(H'_{i,j} - \sum_{k=1}^{i-1} H_{i-k,j-k}\, G_{m-k,j}\right) \quad , i=2,3,..,m-1 \tag{6.27}$$

Then calculate

$$G_{m,N_l} = G'_{m,N_l} - \sum_{k=1}^{m-1} G_{m-k,N_l}\, H_{m-k,N_l-k} \tag{6.28}$$

Back Substitution

$$y_j = d_j/b_j \qquad , j = 1,2,..,N_u \tag{6.29}$$

$$y_j = \frac{1}{G_{m,j-N_u}} (d_j - e_j y_{j-N_u} - c_j y_{j-N_u+1} - a_j y_{j-N_u+m-2}$$

$$-f_j y_{j+N_u+m-1} - \sum_{k=1}^{m-1} G_{k,j-N_u}\, y_{j-m+k}) \quad , j = N_u+1,\ldots,N \tag{6.30}$$

$$P_N = y_N \tag{6.31}$$

$$P_j = y_j - \sum_{k=1}^{m-1} H_{k,j-N_u}\, P_{j+m-k} \quad , j = N-1,N-2,\ldots,N_u+1 \tag{6.32}$$

$$P_j = y_j - f'_j P_{j+N_u} - a'_j P_{j-1+N_u} - c'_j P_{j-m+2+N_u} - e'_j P_{j-m+1+N_u}$$

$$, j = N_u, N_u-1,\ldots,1 \tag{6.33}$$

6.3 Implementation of RAD Scheme into the Simulation Package

To implement the restricted alternating diagonal ordering scheme into the simulation package, the following steps are performed:

1. The standard ordering is reordered to RAD scheme. This step is executed once in the entire simulation. This reordering can be accomplished by finding a RAD scheme that will enclose the standard ordering under consideration as shown in Figure 6.8.
2. The coefficient matrix is transformed into RAD scheme.
3. Then the pressure equations are solved. Thereafter, the pressure vector obtained is reordered into standard ordering to go along with the rest of the simulation package.

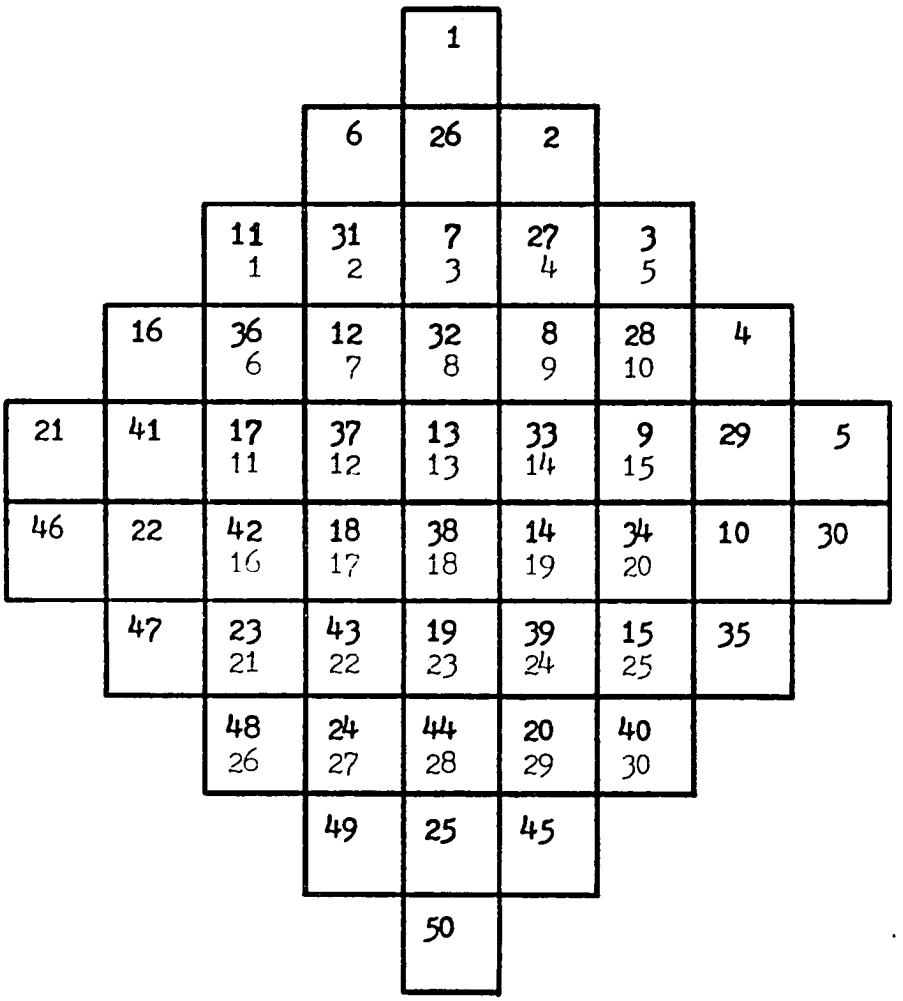


Figure 6.8 - **RAD Ordering Enclosing** Standard Ordering

CHAPTER VII

SIMULATION PACKAGE

This simulation package is a complex collection of computer subroutines tied together by the simulator mainline. The flow chart for the simulator mainline is given in Figure 7.1. These subroutines are developed either out of necessity or simply for the convenience of the engineer.

In this simulation package, a new feature is introduced to save CPU execution time. A chain of imaginary cells is introduced around the active cells. For sealed reservoirs, this new feature allows the execution of active cells only without checking for boundary cells.

There are nine sections in this simulation package. Every section consists of closely related subroutines. Each of these sections will be discussed in detail.

7.1 Input Section

This section of the simulation package consists of the following subroutines:

1. INPUT
2. READ

and requires the PRINT subroutine.

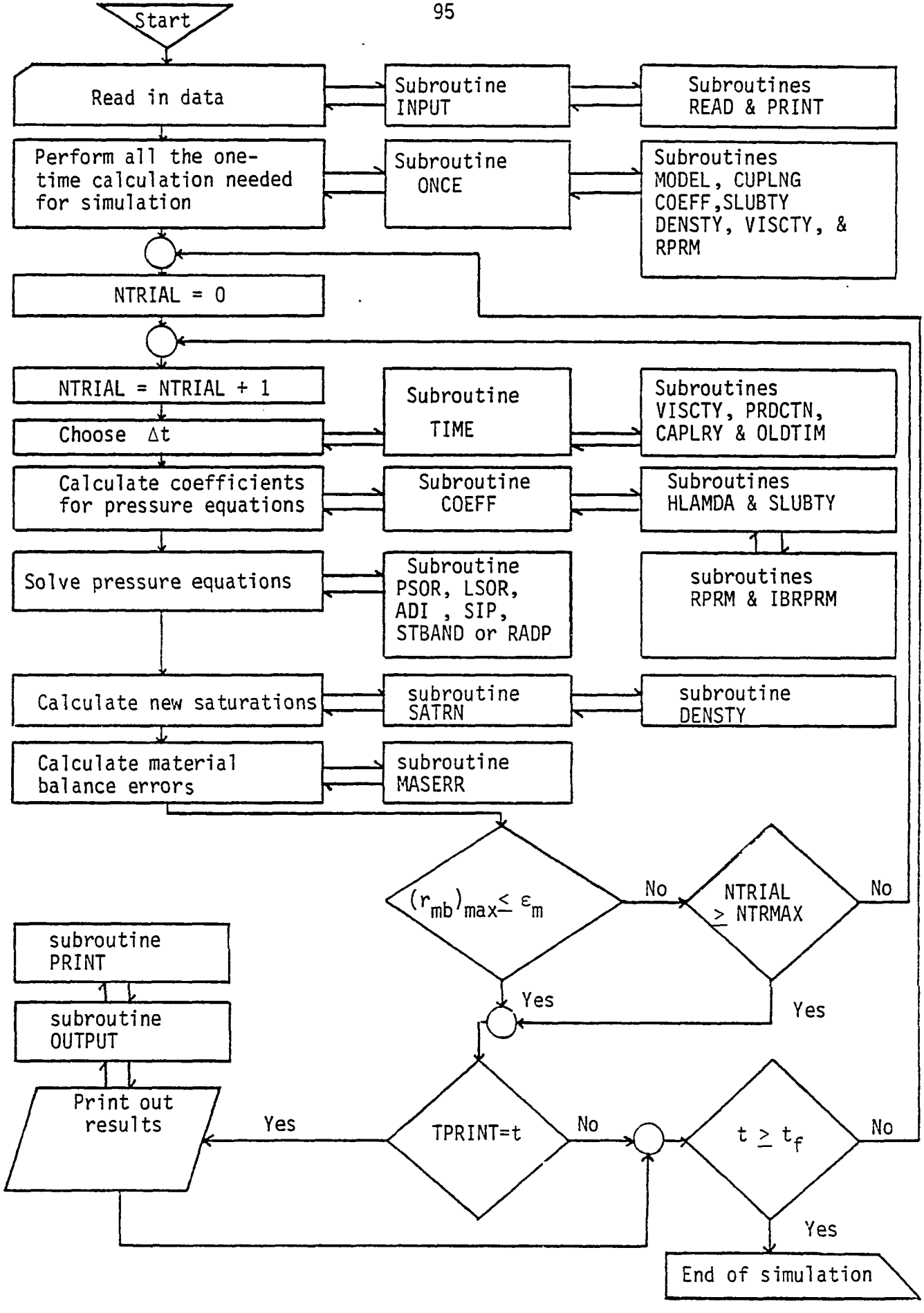


Figure 7.1 - Flow Chart of Simulator Mainline

This section provides a means of reading all input data required in simulation. For input data entered as matrices or vectors, this section will check that the value assigned for each cell is within the range specified for all the matrix or vector values. In addition the entire matrix or vector can be multiplied by a given constant. Also a given constant can be algebrically added to each element of the matrix or vector. If all the elements of the matrix are the same, then the matrix can be entered as a single value. These possible modifications of the elements are very important for history matching. All input data are printed out. Furthermore, the values which lie outside the range specified for the system are printed out along with its cell number and its matrix or vector title so that it can be corrected.

The input data are grouped as follows:

1. Single entries

   Number of cells (N_x,N_y)

   Compressibilities (C_o,C_w,C_r)

   Irreducible saturations (S_{wc},S_{or},S_{gc})

   Densities at standard conditions (ρ_o^*, ρ_w^*, ρ_g^*)

   Bubble point pressure (P_{BP})

   Time control parameters (t_i,t_f, DTMIN, DTMAX, DPLIM, DSLIM)

   Crank-Nicholson parameter (θ)

   Print out interval (OUTMOD)

   Material balance control parameters (ε_m, NTRMAX)

   Pressure control parameters (ε,k_{max},m,ω)

2. Production-Injection data

   Flowing bottom hole pressure (P_{wf})

   Radius of wells (r_w)

   Location of wells and their flow rates

According to the values of the flow rates read, the type of well is determined as shown in Figure 7.2. The vector identifier IQ defines the various types of wells. IQ is defined as follows:

IQ = -2 , if the well is water injection at constant rate

IQ = -1 , if the well is gas injection at constant rate

IQ = 0 , if the cell is not a source or sink

IQ = 1 , if the well produces gas at constant rate

IQ = 2 , if the well produces oil at constant rate

IQ = 3 , if the well produces oil at constant pressure

3. Capillary pressure data (P_{cw},P_{cg})

4. Reservoir geometry

   Dimensions (Δx_i,Δy_i)

   Thickness (h_i)

   Depth (D_i)

5. Rock properties

   Porosity (ϕ_i)

   Permeability (k_{xi},k_{yi})

6. Initial conditions

   Initial oil pressure (P_{o_i})

   Initial oil saturation (S_{o_i})

   Initial water saturation(S_{w_i})

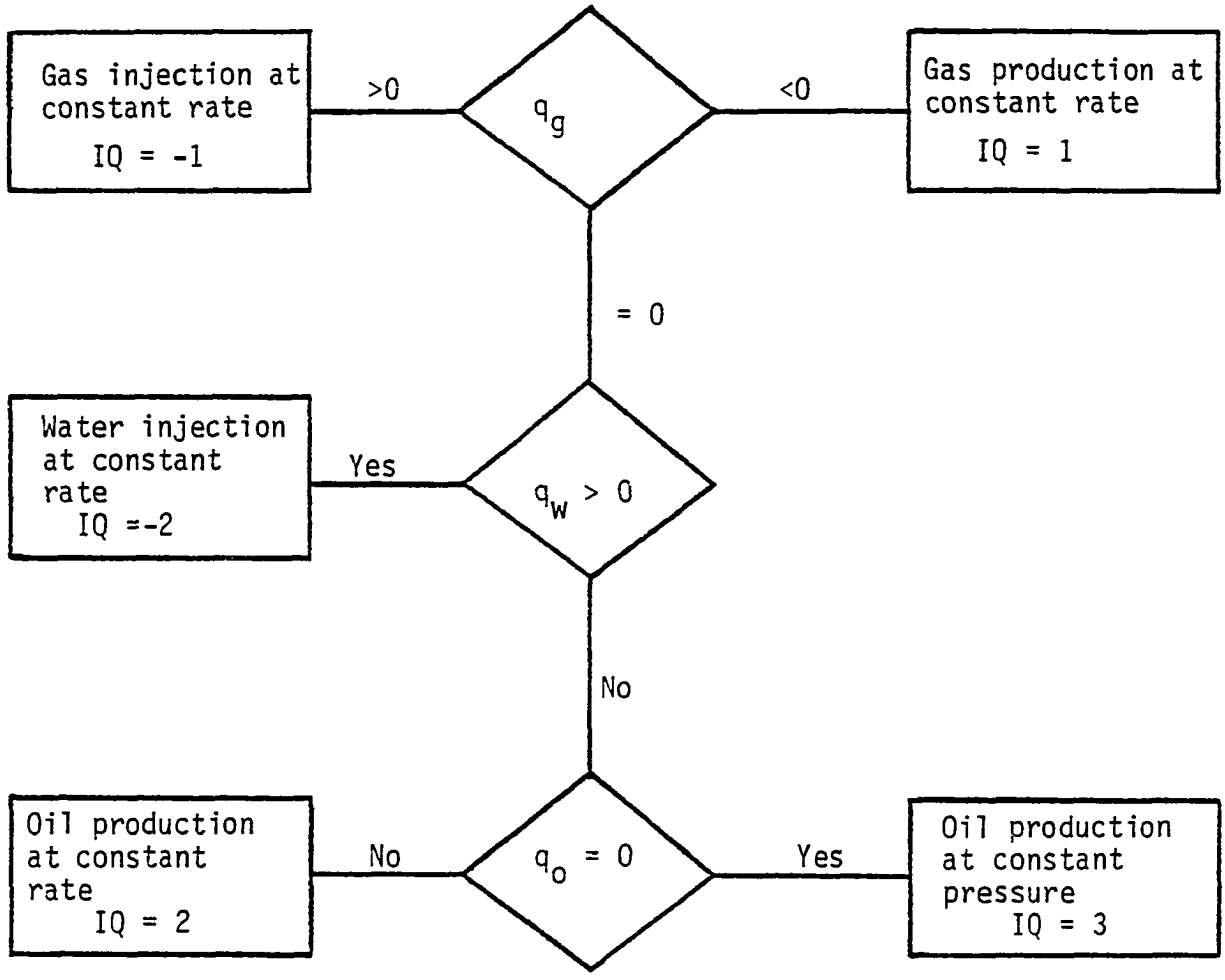


Figure 7.2 - Type of Well Vector Identifier

7.2 One-Time Calculations Section

This section of the simulation package consists of the following subroutines:

1. ONCE
2. MODEL
3. CUPLNG

and requires the following subroutines:

1. SLUBTY
2. DENSTY
3. VISCTY
4. COEFF
5. PRINT

The subroutine ONCE performs the following one-time calculations:

a - Identification of the active cells

b - Initial oil, water, and gas in place

c - Initial bulk volume for each active cell

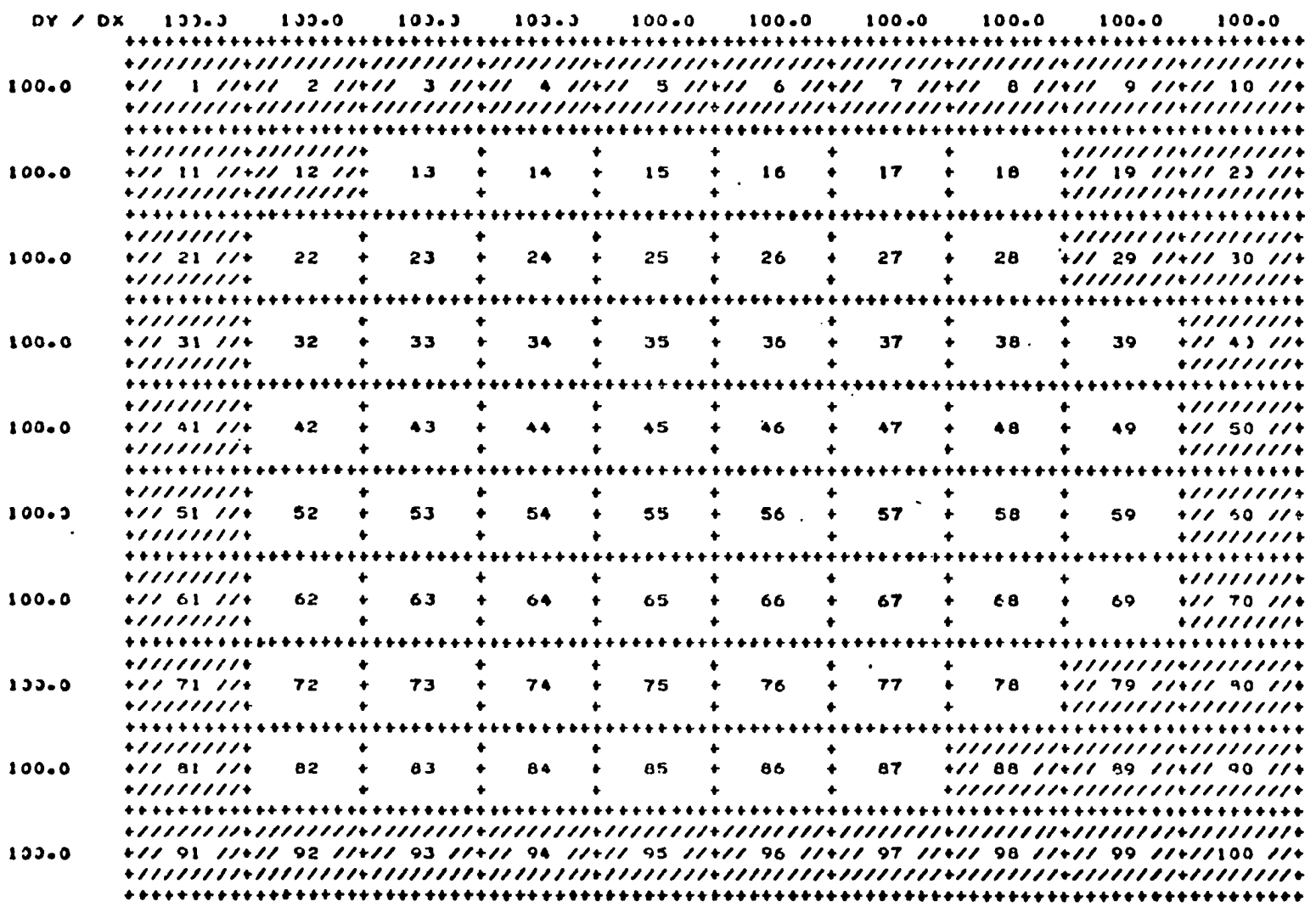
d - Weighted dimensions

e - Calculation of normalizing factor and iteration parameters for ADI and SIP

The subroutine CUPLNG calculates the couplings among cells using equations (3.39,3.40, and 3.41) for horizontal coupling and similar equations for vertical coupling.

The subroutine MODEL produces a sketch, similar to that presented in Figure 7.3. When a reservoir is discretized for simulation, as was discussed in chapter III, often some of the cells are not active. In addition, the cells in the sketch are numbered as they are identified

FIGURE 7.3 - NUMBERING SCHEME OF RESERVOIR MODEL (IMAGINARY CELLS SHADED) - DISTANCE IN METERS



by the simulator. The x-direction of the cells is shown across the top of the sketch and the y-direction of the cells is shown down the left side. In case the number of cells in the x-direction exceeds 10, then the sketch will be slashed after 10 cells and completed afterwards. This routine serves as a check for the shape of the reservoir simulated. Information required for this routine is the number of cells in the x and y directions, their dimensions, and the thickness of each cell as a check for inactive cells.

7.3 Time Control Section

This section of the simulation package consists of the following subroutines:

1. TIME
2. OLDTIM

and requires the following subroutines:

1. VISCTY
2. PRDCTN
3. CAPLRY

This section updates the time and chooses an appropriate time step (Δt) for time level n+1. The choice of Δt is based on the desire to use the maximum Δt possible provided that there is no single cell overdepleted and the truncation and round-off errors are within acceptable tolerance. The following criteria are applied in determining Δt:

1. Select a time step such that the producing cell is not overdepleted[6]. This can be accomplished by selecting a time step such that there is less than 15% change in the fluid saturation in the cell containing the well.

$$\Delta t_{max} = \min (f_i) \qquad \forall\, i \ni IQ > 0 \tag{7.1}$$

where

$$f_i = 0.15(VS_o\rho_o/q_o\rho_o^*)_i \qquad \text{for oil producing wells} \tag{7.2}$$

$$f_i = 0.15\ (VS_g\rho_g/q_g\rho_g^*)_i \qquad \text{for gas producing wells} \tag{7.3}$$

thus, the maximum time increment is less than the time required to produce 15% of a pore volume of the smallest cell containing a well.

2. Todd, at al[16] suggested the following method to control the truncation error. They suggested to limit the saturation and pressure changes per time step. This can be accomplished if Δt does not exceed Δt_{max} defined by

$$\Delta t_{max}^{n+1} = \Delta t^n \frac{DPLIM}{DPMAX} \tag{7.4}$$

$$\Delta t_{max}^{n+1} = \Delta t^n \frac{DSLIM}{DSMAX} \tag{7.5}$$

where

DPLIM = maximum pressure change desired

DSLIM = maximum saturation change desired

for the three phase model

$$DPMAX = \max \left| P_i^{n+1} - P_i^n \right| \qquad \forall\, i \ni h \neq 0 \tag{7.6}$$

$$DSMAX = \max_{j\,=\,o,w,g} \left| (S_j^{n+1} - S_j^n)_i \right| \forall\, i \ni h \neq 0 \tag{7.7}$$

3. To make the input of the engineer more substantial, the range of Δt is specified

$$DTMIN \leq \Delta t \leq DTMAX \tag{7.8}$$

4. To avoid the round-off error, the chosen Δt is truncated to two decimals.

The subroutine OLDTIM calculates the production, capillary pressure, and gravity terms using equations (3.20) through (3.22) based on the last time step.

7.4 Coefficient Matrix Calculation Section

This section of the simulation package consists of the following subroutines:

1. COEFF
2. HLAMDA
3. IBRPRM

and requires the following subroutines:

1. RPRM
2. SLUBTY

The subroutine COEFF calculates the coefficient matrix using equations (3.7) through (3.12).

The HLAMDA and IBRPRM subroutines calculate the interblock flow coefficient ($h\lambda$) for all active cells and for the three phases. The equations and procedures outlined in section 3.3 are incorporated in these subroutines.

The boundary conditions is incorporated into the coefficient matrix through HLAMDA subroutine.

7.5 Pressure Solution Section

This section can be any of the subroutines that performs iterative or direct methods discussed in chapters IV, V, and VI to solve the pressure equations, namely PSOR, LSOR, ADI, SIP of chapter IV, STBAND of chapter V, and RADP of chapter VI.

7.6 Saturation Calculation Section

This section of the simulation package consists of one subroutine called SATRN and it requires the DENSTY subroutine.

This section calculates the new oil, water, and gas saturations using the equations developed in section 3.2.

The calculated saturations are subjected to the logical constraints.

$$S_{wc} \leq S_w \leq 1 \tag{7.9}$$

$$S_{or} \leq S_o \leq 1 - S_w \tag{7.10}$$

The porosity is updated in this section as follows:

$$\phi_i^{n+1} = \phi_i^n \{1 + C_r(P_i^{n+1} - P_i^n)\} \tag{7.11}$$

7.7 Material Balance Section

This section of the simulation package consists of one subroutine called MASERR and it does not require any other subroutine.

This section calculates the oil, water, gas, and total mass balance relative errors.

$$r_j = \frac{1}{MIP_j} (M_j - MIP_j) \quad , j = o,w,g \tag{7.12}$$

$$r_t = \frac{1}{TM} (M_o + M_w + M_g - TM) \tag{7.13}$$

where

$$M_j = \sum_{i=1}^{N} \{V\rho_j S_j - \rho_j^* (q_{jc} + q_j \Delta t)\}_i \quad , j = o,w,g \tag{7.14}$$

$$TM = MIP_o + MIP_w + MIP_g \tag{7.15}$$

In order for a model to maintain stability, it should conserve mass at all time[6]. In this study, $(r_{mb})_{max}$ is used as a convergence criterion on mass.

$$(r_{mb})_{max} = \max \left| r_j \right| \quad , j = o,w,g,t \tag{7.16}$$

This material balance error must be within the predefined tolerance ε_m as follows:

$$(r_{mb})_{max} \leq \varepsilon_m \tag{7.17}$$

This check is done in the mainline of the simulator as shown in Figure 7.1.

When the mass balance converges or the number of trials exceeds the maximum allowed, then the cumulative production is updated as follows:

$$q_{oc}^{n+1} = q_{oc}^{n} + q_{o}^{n+1}(t^{n+1} - t^{n}) \tag{7.18}$$

$$q_{gc}^{n+1} = q_{gc}^{n} + q_{g}^{n+1}(t^{n+1} - t^{n}) \tag{7.19}$$

$$q_{wc}^{n+1} = q_{wc}^{n} + q_{w}^{n+1}(t^{n+1} - t^{n}) \tag{7.20}$$

Also the maximum and minimum Δt used since the last print out time is updated.

7.8 Output Section

This section of the simulation package consists of the following subroutines:

1. OUTPUT
2. PRINT

and requires no other subroutine.

This section prints out the results accumulated since the last print out time.

The subroutine PRINT prints out a given matrix along with its title. The x-direction of the cells is numbered across the top of the matrix and the y-direction of the cells is numbered down the left side. In case the number of cells in the x-direction exceeds 10, the matrix will be slashed after 10 cells and completed afterwards.

7.9 Auxiliary Section

This section of the simulation package consists of the following subroutines:

1. RPRM
2. DENSTY
3. VISCTY
4. SLUBTY
5. CAPLRY
6. PRDCTN

and requires no other subroutine.

This section calculates the fluid properties and the production-injection flow rates. The data for this section can be entered as tables or empirical relationships developed from the reduction of field data.

The subroutine RPRM calculates the realtive permeabilities.

The subroutine DENSTY calculates the oil, water, and gas densities.

The subroutine VISCTY calculates the oil, water, and gas viscosities.

The subroutine CAPLRY calculates the capillary pressure P_{cw} and P_{cg}.

The subroutine SLUBTY calculates the gas-oil ratios and gas-water ratios.

The subroutine PRDCTN determines the flow rate at source or sink cells. The two popular types of flow are considered.

1. Constant flow rate
2. Flow rate at constant pressure at the wellbore.

This simulation package is designed to utilize the explicit production procedure which requires that the oil, gas and water flow rates be evaluated at last time level. The flow rates are evaulated according to the well type designated by IQ vector defined in section 7.1.

1. Water injection well (IQ = -2)

$$q_o = 0 \tag{7.21}$$

$$q_g = 0 \tag{7.22}$$

$$q_w = + \text{ const.} \tag{7.23}$$

2. Gas injection well (IQ = -1)

$$q_o = 0 \tag{7.24}$$

$$q_g = + \text{ const.} \tag{7.25}$$

$$q_w = 0 \tag{7.26}$$

3. Gas producing well at constant rate (IQ = 1)

$$q_o = 0 \tag{7.27}$$

$$q_g = - \text{ const.} \tag{7.28}$$

$$q_w = \frac{k_{rw}\rho_w}{\mu_w\rho_w^*} \frac{\mu_g\rho_g^*}{k_{rg}\rho_g} q_g \tag{7.29}$$

4. Oil producing well at constant rate (IQ = 2)

$$q_o = - \text{ const.} \tag{7.30}$$

$$q_w = \frac{k_{rw}\rho_w}{\mu_w\rho_w^*} \frac{\mu_o\rho_o^*}{k_{ro}\rho_o} q_o \tag{7.31}$$

$$q_g = \frac{k_{rg}\rho_g}{\mu_g\rho_g^*} \frac{\mu_o\rho_o^*}{k_{ro}\rho_o} q_o + \frac{1}{\rho_g^*} (\rho_o^* q_o R_{so} + \rho_w^* q_w R_{sw}) \tag{7.32}$$

5. Oil producing well at constant pressure at the wellbore (IQ = 3)

$$q_o = \frac{-0.0864\sqrt{k_x k_y}\ k_{ro} h \rho_o (P - P_{wf})}{\mu_o \rho_o^* \left\{\ln\left(\frac{1}{r_w}\sqrt{\frac{\Delta x \Delta y}{\pi}}\right) - \frac{3}{4}\right\}} \tag{7.33}$$

$$q_w = \frac{k_{rw}\rho_w}{\mu_w \rho_w^*} \frac{\mu_o \rho_o^*}{k_{ro}\rho_o} q_o \tag{7.34}$$

$$q_g = \frac{k_{rg}\rho_g}{\mu_g \rho_g^*} \frac{\mu_o \rho_o^*}{k_{ro}\rho_o} q_o + \frac{1}{\rho_g^*}\left(\rho_o^* q_o R_{so} + \rho_w^* q_w R_{sw}\right) \tag{7.35}$$

CHAPTER VIII

COMPARATIVE EVALUATION

The simulator outlined in chapter VII was developed and used to simulate Little Axe Unit cf S.E. Denver Pool of Cleveland County in Oklahoma with two sets of permeabilities:

a) Constant permeability for homogeneous case

b) Variable permeability for heterogeneous case

The geometry and properties of this reservoir are given in Appendix D.

The pressure equations are solved with six different numberical procedures, namely PSOR, LSOR, ADI, SIP of chapter IV, STBAND of chapter V, and the new method (RADP) of chapter VI. The computer listing of these numerical procedures is given in Appendix E. A sample of computer simulation output is given in Appendix F.

The comparative evaluation of these numerical procedures includes difficulties of iterative methods, storage requirements, CPU time, mass balance errors, and average reservoir pressure.

8.1 Difficulties of Iterative Methods

Iterative methods have a number of difficulties not present in direct methods. Some of these difficulties are:

1 - Nonreliability: It is well known that the iterative methods do not work for every system of equations. Sometimes direct methods do not converge and sometimes they converge, but not to the right answer.

The last three months of the simulation for homogeneous case, the strongly implicit procedure did not converge within the pre-defined tolerance with the maximum number of iterations allowed. However, we obtained all needed results with higher pressure errors.

2 - Iteration parameters: The estimation of iteration parameters is a serious problem that experience and experimental runs are needed to get a suitable parameter.

3 - Convergence criterion: The relative error of successive approximations are used as convergence criterion. Such a criterion gives no indication of how well the computed solution actually satisfies the original equation. Consequently, the above criterion can be misleading.

8.2 - Storage Requirements

The computer storage requirements of the numerical procedures are given in Table 8.1.

Numerical Procedure	Computer Storage (Bytes)	Computer Storage for one cell (Bytes)	Computer Storage Relative to PSOR
PSOR	13632	39.50	1.00
LSOR	18714	54.24	1.37
ADI	34132	98.93	2.50
SIP	34404	99.72	2.52
STBAND	55778	161.68	4.09
RADP	50894	147.52	3.73

Table 8.1 - Computer Storage Requirements

The direct methods require larger computer storage than iterative methods. However, in direct methods, RADP requires 91% of the computer storage required for STBAND.

8.3 - CPU Time Comparison

The CPU time consumed in simulating the reservoir model as a function of real time is given in Tables 8.2 and 8.3. These two tables are plotted on Figure 8.1 and 8.2 for visual comparison.

From Figures 8.1 and 8.2, it is obvious that iterative methods are faster than direct methods.

In direct methods, the new method (RADP) is substantially faster than the band matrix algorithm applied to standard ordering (STBAND). In a 600 days simulation and 345 cell system, RADP requires 65.5% of the computer time required for STBAND to perform the same simulation.

Among the iterative methods used, the PSOR is faster than LSOR, ADI, and SIP. The ADI is slightly better for homogeneous case while SIP is considerably better for heterogeneous run.

It is noticeable that the CPU time curve is linear for direct methods and concaved upward for iterative methods. Therefore, the CPU time for direct methods is predictable in contrary to iterative methods. This feature is due to the nature of direct methods in which the solution to the system of equations is obtained upon the completion of a finite number of arithmetic operations.

8.4 - Mass Balance Errors

Four types of mass balance relative errors for homogeneous and heterogeneous cases are presented in Tables 8.4 through 8.11. The differences among all types of errors for all six methods are too small to draw any general conclusions.

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.97	1.57	2.23	2.50	5.13	3.38
60	1.91	3.17	4.71	5.00	10.17	6.78
90	2.87	4.68	7.18	7.51	15.26	10.14
120	3.81	6.27	9.60	10.01	20.56	13.53
150	4.71	7.91	11.93	12.51	25.89	16.92
180	5.62	9.54	14.27	15.02	31.27	20.30
210	6.52	11.21	16.68	17.52	36.64	23.78
240	7.44	12.94	19.18	20.00	41.87	27.20
270	8.41	14.75	21.73	22.53	47.09	30.58
300	9.43	16.58	24.29	25.03	52.28	33.99
330	10.48	18.51	26.85	27.52	57.45	37.39
360	11.54	20.50	29.39	30.02	62.62	40.82
390	12.65	22.54	31.95	32.49	67.78	44.29
420	13.82	24.66	34.60	34.99	72.94	47.76
450	15.11	26.79	37.37	37.48	78.11	51.20
480	16.46	28.91	40.11	39.98	83.29	54.62
510	17.82	31.07	43.00	43.52	88.59	58.07
540	19.22	33.28	45.81	46.50	93.78	61.57
570	20.80	35.66	49.60	50.42	98.97	65.04
600	22.82	38.25	53.51	56.00	104.14	68.52

TABLE 8.2 - CPU TIME FOR HOMOGENEOUS CASE IN SECONDS

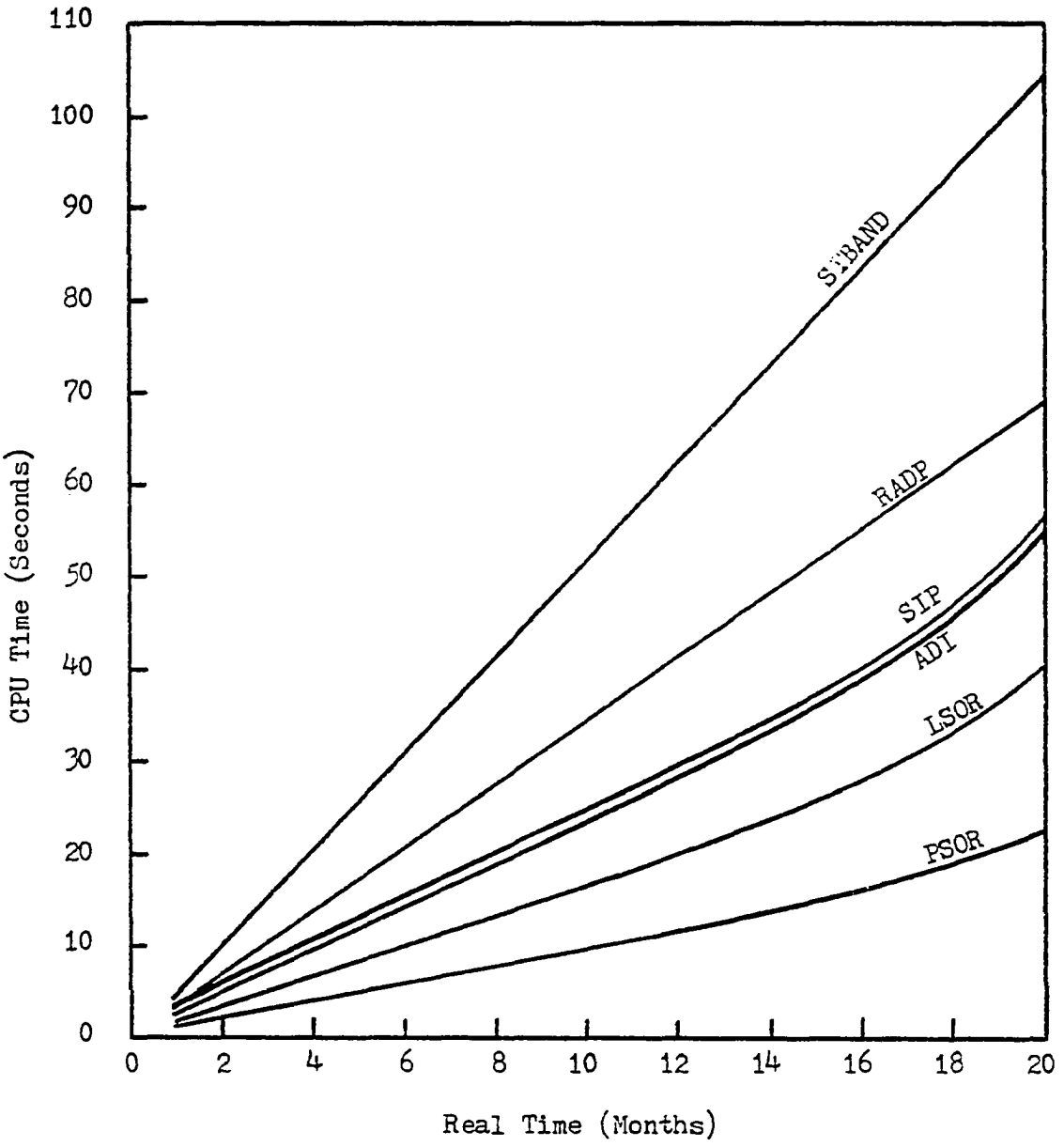


Figure 8.1 - CPU Time vs. Real Time (Homogeneous Case - 345 Cells)

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.94	1.58	2.52	2.11	5.44	3.39
60	1.90	3.17	5.10	4.22	10.80	6.79
90	2.85	4.78	7.76	6.34	16.17	10.23
120	3.79	6.33	10.33	8.32	21.55	13.65
150	4.71	7.85	12.76	10.54	26.92	17.05
180	5.60	9.42	15.25	12.59	32.25	20.44
210	6.55	11.01	17.74	14.72	37.57	23.89
240	7.49	12.65	20.17	16.72	42.83	27.38
270	8.42	14.38	22.72	19.01	48.09	30.88
300	9.41	16.17	25.14	21.00	53.52	34.28
330	10.43	18.06	27.70	23.22	58.82	37.69
360	11.45	20.04	30.27	25.29	64.00	41.13
390	12.63	22.08	33.00	27.42	69.12	44.54
420	13.83	24.09	35.65	29.50	74.22	47.96
450	15.04	26.12	38.27	31.70	79.58	51.37
480	16.39	28.24	40.92	33.75	84.84	54.78
510	17.74	30.38	43.83	35.80	90.05	58.20
540	19.06	32.69	47.37	38.02	95.18	61.64
570	20.58	35.10	51.29	40.10	100.42	65.10
600	22.19	37.62	57.56	43.01	105.59	68.49

TABLE 8.3 - CPU TIME FOR HETEROGENEOUS CASE IN SECONDS

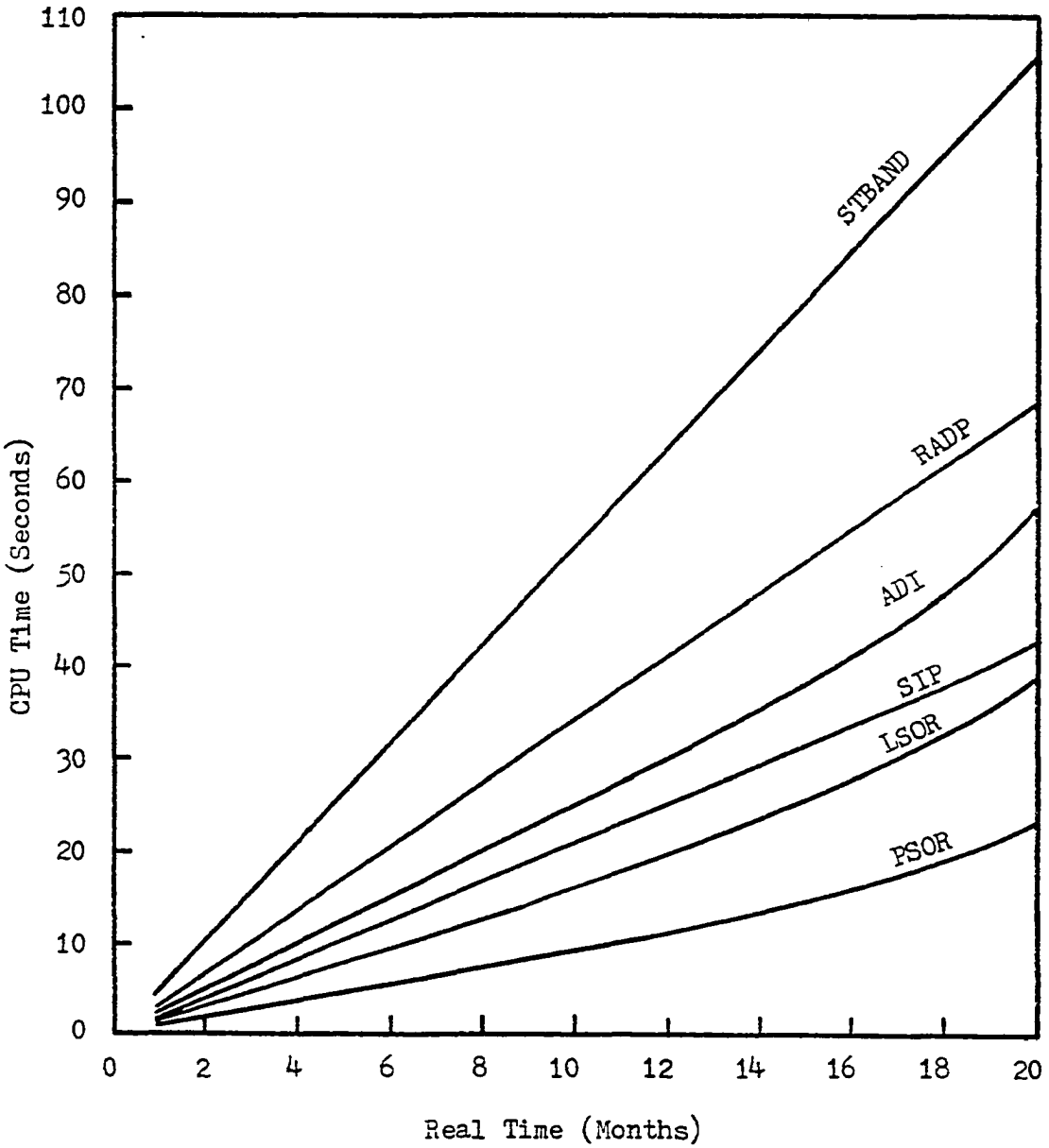


Figure 8.2 - CPU Time vs. Real Time (Heterogeneous Case - 345 Cells)

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0009	0.0010	0.0010	0.0008	0.0010	0.0010
60	0.0018	0.0018	0.0019	0.0026	0.0019	0.0019
90	0.0026	0.0027	0.0027	0.0033	0.0027	0.0027
120	0.0034	0.0035	0.0035	0.0041	0.0035	0.0035
150	0.0042	0.0042	0.0042	0.0047	0.0042	0.0042
180	0.0049	0.0049	0.0049	0.0045	0.0050	0.0049
210	0.0055	0.0055	0.0056	0.0051	0.0056	0.0056
240	0.0062	0.0062	0.0064	0.0061	0.0063	0.0062
270	0.0068	0.0068	0.0071	0.0068	0.0069	0.0068
300	0.0075	0.0075	0.0082	0.0074	0.0076	0.0075
330	0.0081	0.0081	0.0090	0.0078	0.0082	0.0081
360	0.0090	0.0090	0.0102	0.0084	0.0091	0.0090
390	0.0098	0.0098	0.0113	0.0086	0.0099	0.0098
420	0.0111	0.0111	0.0129	0.0092	0.0112	0.0111
450	0.0122	0.0122	0.0142	0.0099	0.0123	0.0122
480	0.0144	0.0143	0.0163	0.0105	0.0145	0.0144
510	0.0164	0.0163	0.0178	0.0109	0.0164	0.0164
540	0.0191	0.0182	0.0201	0.0111	0.0185	0.0177
570	0.0216	0.0201	0.0230	0.0115	0.0216	0.0213
600	0.0027	-0.0056	0.0267	0.0116	0.0083	0.0054

TABLE 8.4 - TOTAL MASS BALANCE RELATIVE ERROR FOR HOMOGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0010	0.0010	0.0010	0.0009	0.0010	0.0010
60	0.0019	0.0019	0.0019	0.0020	0.0019	0.0019
90	0.0027	0.0027	0.0027	0.0026	0.0028	0.0027
120	0.0035	0.0035	0.0035	0.0034	0.0036	0.0036
150	0.0043	0.0043	0.0043	0.0031	0.0043	0.0043
180	0.0050	0.0050	0.0051	0.0049	0.0051	0.0050
210	0.0057	0.0057	0.0058	0.0055	0.0058	0.0058
240	0.0065	0.0064	0.0067	0.0064	0.0066	0.0065
270	0.0072	0.0071	0.0075	0.0070	0.0072	0.0072
300	0.0080	0.0079	0.0087	0.0077	0.0081	0.0080
330	0.0086	0.0086	0.0097	0.0078	0.0087	0.0087
360	0.0096	0.0095	0.0112	0.0082	0.0097	0.0096
390	0.0103	0.0103	0.0124	0.0090	0.0104	0.0103
420	0.0114	0.0114	0.0144	0.0094	0.0116	0.0115
450	0.0126	0.0126	0.0158	0.0099	0.0128	0.0127
480	0.0146	0.0145	0.0178	0.0100	0.0147	0.0147
510	0.0163	0.0163	0.0199	0.0102	0.0167	0.0167
540	0.0188	0.0186	0.0231	0.0102	0.0191	0.0191
570	0.0216	0.0213	0.0274	0.0110	0.0219	0.0219
600	0.0184	0.0170	0.0293	0.0113	0.0178	0.0174

TABLE 8.5 - TOTAL MASS BALANCE RELATIVE ERROR FOR HETEROGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0015	0.0015	0.0015	0.0013	0.0015	0.0015
60	0.0030	0.0031	0.0031	0.0043	0.0031	0.0031
90	0.0045	0.0045	0.0045	0.0056	0.0046	0.0045
120	0.0058	0.0059	0.0059	0.0069	0.0060	0.0059
150	0.0071	0.0072	0.0072	0.0079	0.0073	0.0072
180	0.0083	0.0084	0.0083	0.0075	0.0085	0.0084
210	0.0094	0.0094	0.0094	0.0084	0.0096	0.0095
240	0.0105	0.0104	0.0101	0.0100	0.0106	0.0105
270	0.0113	0.0113	0.0107	0.0113	0.0115	0.0114
300	0.0121	0.0120	0.0107	0.0124	0.0122	0.0121
330	0.0126	0.0125	0.0109	0.0131	0.0128	0.0126
360	0.0128	0.0127	0.0100	0.0139	0.0130	0.0129
390	0.0126	0.0125	0.0095	0.0142	0.0128	0.0126
420	0.0121	0.0120	0.0076	0.0152	0.0123	0.0121
450	0.0107	0.0105	0.0061	0.0164	0.0108	0.0107
480	0.0094	0.0094	0.0031	0.0173	0.0094	0.0093
510	0.0053	0.0055	0.0012	0.0180	0.0055	0.0054
540	0.0017	0.0011	-0.0022	0.0187	0.0015	-0.0004
570	-0.0041	-0.0045	-0.0068	0.0197	-0.0035	-0.0037
600	-0.0477	-0.0615	-0.0127	0.0206	-0.0353	-0.0412

TABLE 8.6 - OIL MASS BALANCE RELATIVE ERROR FOR HOMOGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0015	0.0016	0.0016	0.0015	0.0016	0.0016
60	0.0031	0.0031	0.0032	0.0032	0.0032	0.0032
90	0.0046	0.0046	0.0046	0.0043	0.0047	0.0047
120	0.0060	0.0060	0.0060	0.0057	0.0061	0.0061
150	0.0073	0.0073	0.0073	0.0052	0.0074	0.0074
180	0.0085	0.0085	0.0085	0.0081	0.0086	0.0086
210	0.0097	0.0096	0.0095	0.0092	0.0098	0.0097
240	0.0105	0.0104	0.0100	0.0108	0.0106	0.0105
270	0.0113	0.0112	0.0106	0.0117	0.0114	0.0113
300	0.0118	0.0116	0.0102	0.0125	0.0119	0.0118
330	0.0122	0.0120	0.0102	0.0126	0.0124	0.0122
360	0.0124	0.0121	0.0087	0.0132	0.0125	0.0124
390	0.0124	0.0120	0.0078	0.0145	0.0125	0.0123
420	0.0120	0.0116	0.0051	0.0152	0.0120	0.0118
450	0.0107	0.0103	0.0037	0.0160	0.0107	0.0104
480	0.0095	0.0092	0.0006	0.0158	0.0094	0.0092
510	0.0058	0.0052	-0.0026	0.0158	0.0052	0.0049
540	0.0061	0.0061	-0.0081	0.0159	0.0056	0.0054
570	-0.0006	0.0001	-0.0159	0.0177	-0.0012	-0.0016
600	-0.0145	-0.0155	-0.0172	0.0174	-0.0166	-0.0175

TABLE 8.7 - OIL MASS BALANCE RELATIVE ERROR FOR HETEROGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
60	0.0002	0.0002	0.0002	0.0004	0.0002	0.0002
90	0.0002	0.0003	0.0002	0.0004	0.0003	0.0003
120	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
150	0.0003	0.0003	0.0003	0.0004	0.0003	0.0003
180	0.0003	0.0004	0.0005	0.0006	0.0004	0.0004
210	0.0004	0.0004	0.0007	0.0006	0.0004	0.0004
240	0.0006	0.0007	0.0016	0.0009	0.0006	0.0006
270	0.0008	0.0009	0.0024	0.0009	0.0009	0.0009
300	0.0015	0.0016	0.0048	0.0009	0.0015	0.0015
330	0.0022	0.0023	0.0065	0.0010	0.0022	0.0022
360	0.0040	0.0041	0.0105	0.0011	0.0040	0.0040
390	0.0061	0.0062	0.0136	0.0012	0.0061	0.0061
420	0.0100	0.0102	0.0198	0.0013	0.0100	0.0100
450	0.0142	0.0145	0.0250	0.0014	0.0143	0.0143
480	0.0220	0.0218	0.0336	0.0017	0.0220	0.0220
510	0.0314	0.0309	0.0397	0.0017	0.0310	0.0311
540	0.0441	0.0426	0.0497	0.0012	0.0423	0.0427
570	0.0571	0.0540	0.0625	0.0008	0.0563	0.0562
600	0.0644	0.0603	0.0792	-0.0002	0.0630	0.0634

TABLE 8.8 - WATER MASS BALANCE RELATIVE ERROR FOR HOMOGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
60	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002
90	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002
120	0.0003	0.0003	0.0003	0.0004	0.0003	0.0003
150	0.0003	0.0003	0.0003	0.0005	0.0003	0.0003
180	0.0004	0.0004	0.0006	0.0006	0.0004	0.0004
210	0.0006	0.0006	0.0010	0.0008	0.0006	0.0006
240	0.0012	0.0013	0.0024	0.0008	0.0013	0.0013
270	0.0017	0.0018	0.0035	0.0008	0.0017	0.0017
300	0.0030	0.0030	0.0066	0.0013	0.0030	0.0030
330	0.0039	0.0041	0.0090	0.0016	0.0040	0.0040
360	0.0059	0.0062	0.0145	0.0017	0.0060	0.0060
390	0.0076	0.0079	0.0185	0.0018	0.0077	0.0077
420	0.0109	0.0113	0.0267	0.0019	0.0111	0.0112
450	0.0151	0.0156	0.0316	0.0020	0.0156	0.0157
480	0.0219	0.0221	0.0404	0.0024	0.0224	0.0226
510	0.0303	0.0310	0.0495	0.0029	0.0318	0.0322
540	0.0374	0.0369	0.0642	0.0027	0.0385	0.0390
570	0.0517	0.0504	0.0846	0.0023	0.0532	0.0540
600	0.0630	0.0607	0.0913	0.0032	0.0639	0.0645

TABLE 8.9 - WATER MASS BALANCE RELATIVE ERROR FOR HETEROGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0015	0.0015	0.0015	0.0013	0.0015	0.0015
60	0.0030	0.0031	0.0031	0.0043	0.0031	0.0031
90	0.0045	0.0045	0.0045	0.0056	0.0046	0.0045
120	0.0058	0.0059	0.0059	0.0069	0.0060	0.0059
150	0.0071	0.0072	0.0072	0.0079	0.0073	0.0072
180	0.0083	0.0084	0.0083	0.0070	0.0085	0.0084
210	0.0094	0.0094	0.0094	0.0084	0.0096	0.0095
240	0.0105	0.0105	0.0101	0.0100	0.0106	0.0105
270	0.0113	0.0113	0.0107	0.0113	0.0115	0.0114
300	0.0120	0.0120	0.0107	0.0124	0.0122	0.0121
330	0.0126	0.0125	0.0109	0.0131	0.0128	0.0126
360	0.0125	0.0123	0.0100	0.0139	0.0127	0.0125
390	0.0125	0.0125	0.0095	0.0142	0.0128	0.0126
420	0.0105	0.0103	0.0075	0.0152	0.0109	0.0106
450	0.0103	0.0102	0.0061	0.0164	0.0106	0.0104
480	0.0042	0.0041	0.0028	0.0173	0.0047	0.0043
510	0.0030	0.0032	0.0009	0.0180	0.0038	0.0033
540	-0.0100	-0.0090	-0.0035	0.0187	-0.0066	-0.0069
570	-0.0137	-0.0122	-0.0089	0.0197	-0.0124	-0.0145
600	-0.0247	-0.0231	-0.0173	0.0206	-0.0219	-0.0249

TABLE 8.10 - GAS MASS BALANCE RELATIVE ERROR FOR HOMOGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	0.0015	0.0016	0.0016	0.0015	0.0016	0.0016
60	0.0031	0.0031	0.0032	0.0032	0.0032	0.0032
90	0.0046	0.0046	0.0046	0.0043	0.0047	0.0047
120	0.0060	0.0060	0.0060	0.0057	0.0061	0.0061
150	0.0074	0.0073	0.0073	0.0052	0.0074	0.0074
180	0.0085	0.0085	0.0085	0.0081	0.0086	0.0086
210	0.0097	0.0096	0.0095	0.0092	0.0098	0.0097
240	0.0105	0.0104	0.0100	0.0103	0.0106	0.0105
270	0.0113	0.0112	0.0106	0.0117	0.0114	0.0113
300	0.0118	0.0116	0.0102	0.0125	0.0119	0.0118
330	0.0122	0.0120	0.0102	0.0125	0.0124	0.0122
360	0.0124	0.0121	0.0087	0.0132	0.0125	0.0123
390	0.0124	0.0120	0.0078	0.0145	0.0125	0.0123
420	0.0113	0.0108	0.0051	0.0152	0.0115	0.0112
450	0.0106	0.0102	0.0037	0.0160	0.0106	0.0104
480	0.0061	0.0055	0.0005	0.0155	0.0062	0.0058
510	0.0051	0.0044	-0.0028	0.0156	0.0047	0.0043
540	-0.0040	-0.0047	-0.0089	0.0159	-0.0041	-0.0048
570	-0.0062	-0.0069	-0.0177	0.0177	-0.0066	-0.0075
600	-0.0220	-0.0218	-0.0223	0.0174	-0.0222	-0.0239

TABLE 8.11 - GAS MASS BALANCE RELATIVE ERROR FOR HETEROGENEOUS CASE

8.5 - Average Reservoir Pressure

The average reservoir pressure is given in Tables 8.12 and 8.13. For a given time, the differences of pressures are relatively small to draw any conclusions.

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	15617	15624	15625	15577	15628	15626
60	15585	15594	15596	15807	15602	15597
90	15587	15593	15595	15762	15605	15598
120	15554	15561	15564	15739	15576	15567
150	15548	15554	15558	15678	15571	15560
180	15521	15526	15530	15420	15546	15534
210	15509	15514	15521	15374	15537	15522
240	15487	15488	15494	15441	15515	15498
270	15474	15475	15483	15481	15504	15485
300	15451	15450	15459	15465	15482	15462
330	15441	15439	15450	15416	15473	15452
360	15419	15414	15429	15413	15452	15429
390	15414	15409	15420	15336	15449	15424
420	15389	15382	15401	15353	15424	15398
450	15397	15389	15391	15419	15433	15406
480	15373	15362	15377	15448	15409	15380
510	15409	15397	15369	15456	15443	15415
540	15287	15271	15361	15424	15424	15394
570	15263	15203	15358	15447	15216	15093
600	14549	14302	15360	15437	14919	14755

TABLE 8.12 - AVERAGE RESERVOIR PRESSURE FOR HOMOGENEOUS CASE

REAL TIME DAYS	NUMERICAL PROCEDURE					
	PSOR	LSOR	ADI	SIP	STBAND	RADP
30	15624	15627	15628	15613	15631	15629
60	15595	15600	15604	15633	15609	15605
90	15604	15602	15607	15554	15616	15610
120	15573	15573	15579	15547	15591	15583
150	15577	15569	15578	15306	15591	15581
180	15552	15543	15553	15508	15570	15558
210	15550	15536	15549	15488	15566	15553
240	15528	15512	15525	15527	15547	15532
270	15522	15502	15521	15508	15542	15525
300	15502	15480	15500	15511	15525	15506
330	15498	15472	15497	15378	15521	15500
360	15477	15449	15477	15356	15502	15480
390	15477	15445	15476	15405	15502	15478
420	15454	15421	15458	15383	15483	15457
450	15464	15428	15459	15394	15492	15465
480	15441	15401	15442	15279	15469	15440
510	15482	15435	15445	15203	15505	15475
540	15468	15406	15434	15120	15480	15448
570	15526	15479	15443	15232	15537	15501
600	15378	15334	15445	15156	15311	15251

TABLE 8.13 - AVERAGE RESERVOIR PRESSURE FOR HETEROGENEOUS CASE

CHAPTER IX

CONCLUSIONS

Based on the numerical simulation results observed in this study, it is possible to draw the following conclusions:

1. Direct methods are more reliable than iterative ones. Some iterative methods fail to converge at certain times and under certain conditions. The error will propagate from this point on.
2. The search for optimum iteration parameters in iterative methods is a serious problem especially for SIP.
3. The ADI and SIP methods are sensitive to the type of reservoir models while direct methods are not. This observation is implied by the CPU time requirements. The ADI is slightly better for homogeneous case while SIP is considerably better for heterogeneous run.
4. Iterative methods require less computer storage than direct methods no matter how efficient the direct method is.
5. The iterative methods are faster than direct methods for the 345 cell model simulated in this study.
6. Among the iterative methods used, the PSOR is faster than LSOR, ADI, and SIP.

7. RADP requires about 65.5% of the CPU time consumed by STBAND. RADP requires about 91% of the computer storage required for STBAND. These improvements are due to the fact that RADP generates less new non-zero elements in the direct solution process. Therefore, RADP is more competitive to the iterative methods than STBAND direct method.

NOMENCLATURE

$\underline{\underline{A}}$	coefficient matrix
A	cross-sectional area to flow
A_i, B_i, C_i, F_i, E_i	SIP coefficient vectors
a_i, b_i, c_i, e_i, f_i	coefficients of the general pressure equation
ADI	Alternating Diagonal Implicit
B_1	PVT term
B_2	production term
B_3	capillary pressure term
B_4	gravity term
B_5	second degree derivative term
$\underline{\underline{C}}$	composite matrix (=L\U)
C	compressibility
D	subsea level, positive downward
DPLIM	maximum pressure change desired
DPMAX	maximum pressure change calculated
DSLIM	maximum saturation change desired
DSMAX	maximum saturation change calculated
DTMAX	maximum time step allowed
DTMIN	minimum time step allowed
d	day
$\underline{d}$	right-hand side vector of pressure equations

d_i	an element of $\underline{d}$
f	an arbitrary function
ft	foot
G_{ij}, H_{ij}	vectors in RADP
g	acceleration due to gravity
g_i, h_i	the new coefficients introduced for SIP
h	formation thickness
I	the largest of N_x and N_y
J	the smallest of N_x and N_y
K	Kelvin absolute temperature
k	absolute permeability
kg	kilogram
kPa	kilo-Pascal
k_{max}	maximum number of iterations on pressure
k_r	relative permeability
L	lower triangular matrix
LSOR	Line Successive Overrelaxation
L_d	length of the diagonal along which numbering is applied
l	an element of L
lb_m	pound mass
M	mass
MIP	mass in place
m	meter
m	semi-bandwidth
m	number of iteration parameters in the system
N	number of active cells

N_d	number of diagonals in the system
N_l	number of rows in the lower half of matrix $\mathbb{A}$
N_u	number of rows in the upper half of matrix $\mathbb{A}$
N_x	number of cells in the x-direction
N_y	number of cells in the y-direction
NTRMAX	maximum number of trials on material balance
OUTMOD	print out interval
$\underline{P}$	pressure vector
Pa	Pascal
Psi	pound per square inch
P_{BP}	bubble point pressure
P_{cg}	capillary pressure to gas
P_{cw}	capillary pressure to water
P_i	an element of $\underline{P}$
P_{pc}	pseudo critical pressure
P_{ref}	reference pressure
P_{wf}	flowing bottom hole pressure of the well
PSOR	Point Successive Overrelaxation
P*	pressure at standard condition
q	flow rate, positive rates denote injection, negative rates denote production
R	Rankine absolute temperature
RAD	Restricted Alternating Diagonal
RADP	RAD Procedure
R_{so}	solubility of gas into oil
R_{sw}	solubility of gas into water
r	constant multiplier

r_i, r_j	relative error
r_w	well radius
r_{max}	maximum absolute relative error
$(r_{mb})_{max}$	mass balance maximum absolute relative error
S	saturation
S	storage
S_{gc}	equilibrium gas saturation
S_{or}	irreducible oil saturation
St.	Standard
S_{wc}	connate water saturation
SCF	Standard Cubic Feet
SIP	Strongly Implicit Procedure
SOR	Successive Overrelaxation
STB	Stock tank barrel
STBAND	band matrix algorithm applied on standard ordering
s	second
T	temperature
T	transmissibility
T_{pc}	pseudo critical temperature
TM	total mass
TPRINT	print out time
T*	temperature at standard conditions
t	time
U	upper triangular matrix
u	an element of U
V_i	pore volume of cell i

W	work
z	gas compressibility factor
z^*	gas compressibility factor at standard conditions

Greek Letters

α	iteration parameter for SIP
β	iteration parameter for ADI
γ_i	normalizing factor for β
Δt	time increment
$\Delta x, \Delta y$	distance increments
ε	pressure tolerance
ε_m	mass balance tolerance
η	N_x
θ	Crank-Nicholson parameter
λ	mobility
μ	viscosity
μm	micro-meter
ρ	density
ρ^*	density at standard conditions
Φ	potential
ϕ	porosity
ω	iteration parameter for SOR

Subscripts and Superscripts

f	final
g	gas phase
i	cell number
i	initial
j	phase identifier (oil,water,or gas)

jc	cumulative of phase j
k	iteration counter
m	mass
n	time level
o	oil phase
r	rock
t	total
w	water phase
x	x-direction
y	y-direction

REFERENCES

1. Al-Marhoun, M.A., "Two-Dimensional Two-Phase Petroleum Reservoir Simulator (Development and System Study)," M.S. Thesis (1976), University of Petroleum and Minerals, Dhahran, Saudi Arabia.
2. Birkhoff, G.; Varga, R.S.; and Young, D., "Alternating Direction Implicit Methods," Advances in Computers (1962) 3, 189-273.
3. Brietenbach, E.A.; Thurnau, D.H.; and Van Poollen, H.K., "Solution of the Immiscible Fluid Flow Simulation Equation," Soc. Pet. Eng. J. (June 1969) 155-169.
4. Craft, B.C.; and Hawkins, M.F., "Applied Petroleum Reservoir Engineering" Prentice-Hall, Inc., Englewood Cliffs, N.J. (1959).
5. Craig, F.F., "Compositional Flow in Porous Media: A Reservoir Simulator," Ph.D. Dissertation (1976), University of Oklahoma, Norman, Oklahoma, U.S.A.
6. Crichlow, H.B., "Modern Reservoir Engineering - A Simulation Approach," Prentice-Hall, Inc., N.J. (1977).
7. Crout, P.D., "A Short Method for Evaluating Determinants and Solving Systems of Linear Equations with Real or Complex Coefficients," Trans. AIEE (1941), 60, 1235.
8. Gopal, V.N., "Gas Z-Factor Equations Developed for Computer," Oil & Gas J. (August 8, 1977) 58-60.
9. McDonald, A.E.; and Trimble, R.H., "Efficient Use of Mass Storage During Elimination for Sparse Sets of Simulation Equations," Soc. Pet. Eng. J. (Aug. 1977) 300-316.

10. Merchant, A.R.; Arnold, M.D.; and Harvey, A.H., "A Technique for Improving Material Balance Accuracy in Reservoir Simulation Models," SPE No. 4548.

11. Peaceman, D.W. and Rachford, H.H., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Indust. Appl. Math. (1955), 3, 28-41.

12. Price, H.S. and Coats, K.H., "Direct Methods in Reservoir Simulation," Transactions of SPE of AIME (1974, vol. 257, pp 295-308).

13. Stone, H.L., "Estimation of Three-Phase Relative Permeability and Residual Oil Data," JPT, Oct. - Dec., 1973, p. 53.

14. Stone, H.L., "Iterative Solution of Implicit Approximation of Multidimensional Partial Differential Equations," SIAM Numer. Anal. (1968), 5, 530.

15. Suarez, A. and Farouq Ali, S.M., "A Comparative Evaluation of Different Versions of the Strongly Implicit Procedure (SIP)," Fourth SPE Symposium on Numberical Simulation of Reservoir Performance, Los Angeles, California, Feb. 19-20, 1976, SPE No. 5732.

16. Todd,M.R.; O'Dell, P.M.; and Hirasaki, R.J., "Methods of Increased Accuracy in Numberical Reservoir Simulators," Trans. AIME (1972), 253.

17. Varga, R.S., "Matrix Iterative Analysis," Englewood Cliffs, N.J.: Prentice Hall, 1962.

18. Von Rosenberg, D.U., "Methods for the Numerical Solution of Partial Differential Equations," New York: American Elsevier Publishing Co., Inc., 1969.

19. Watts, J.W., "An Iterative Matrix Solution Method Suitable for Anisotropic Problems," SPE Reprint No. 11 (1973), p. 119.

20. Weinstein, H.G.; Stone, H.L.; and Kwan, T.V., "Simultaneous Solution of Multiphase Reservoir Flow Equations," SPE Reprint No. 11 (1973), p.124.

21. Woo, P.T.; Roberts, S.J.; and Gustavson, F.G., "Application of Sparse Matrix Techniques in Reservoir Simulation," SPE No. 4544.

APPENDICES

APPENDIX A

UNITS AND CONVERSION FACTORS

Symbol	Definition	SI Units	Practical Units	English Units	Conversion factor *
A	Area	m^2	m^2	ft^2	0.09290
C	Compressibility	Pa^{-1}	kPa^{-1}	Psi^{-1}	0.14504
D	Depth	m	m	ft	0.30480
g	Gravity	m/s^2	m/s^2	ft/s^2	0.30480
h	Thickness	m	m	ft	0.30480
k	Permeability	m^2	μm^2	darcy	0.98690
P	Pressure	Pa	kPa	Psi	6.89476
$q_{o,w}$	Liquid Flow Rate	$St.m^3/s$	$St.m^3/d$	STB/d	0.15899
q_g	Gas Flow Rate	$St.m^3/s$	$St.m^3/d$	SCF/d	0.02832
R_{sj}	Solubility	kg/kg	kg/kg	SCF/STB	0.17811 ρ_g^* / ρ_j^*
R_{sj}	Solubility	kg/kg	kg/kg	SCF/SCF	ρ_g^* / ρ_j^*
T	Temperature	K	K	R	5/9
t	Time	s	d	d	1
μ	Viscosity	Pa-s	mPa-s	cp	1
ρ	Density	kg/m^3	kg/m^3	lb_m/ft^3	16.0185

*To obtain the practical unit, multiply the English unit by the corresponding conversion factor.

APPENDIX B

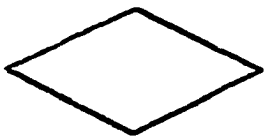
FLOW CHART CONVENTION

The following boxes, each of characteristic shape, are used for constructing the flow charts in this study.

Commencement

Computation

Do loop



If statment

Input (read)

Output (print or punch)

Conclusion

APPENDIX C

CONSTANT VALUES USED IN THIS STUDY

DPLIM = 1000 kPa

DSLIM = 0.15

DTMAX = 10 d

DTMIN = 10 d (0.01 d for SIP)

g = 9.81 m/s^2

k_{max} = 55

m = 10

NTRMAX = 20

N_x = 15

N_y = 23

OUTMOD = 30 d

P^* = 100 kPa

P_{wf} = 4500 kPa

r_w = 0.1 m

T^* = 273.16 K

t_f = 600 d

t_i = 0 d

z^* = 1.0

Δx = 100 m

Δy = 100 m

ε = 0.0005

ε_m = 0.10

θ = 0.5

ρ_g^* = 0.9 kg/m^3

ρ_o^* = 825 kg/m^3

ρ_w^* = 1160 kg/m^3

ω = 1.6

APPENDIX D

Geometry and Properties of the Reservoir Model

E.1 Constant Properties of the Reservoir Model

E.2 Relative Permeabilities

E.3 Viscosities

E.4 Solubilities

E.5 Densities

E.6 Capillary Pressure Data

E.7 Reservoir Figures

E.1 Constant Properties of the Reservoir Model

C_o = 2.90×10^{-6} kPa^{-1}

C_w = 0.43×10^{-6} kPa^{-1}

C_r = 0.58×10^{-6} kPa^{-1}

k_x = 0.05 μm^2 (homogeneous case)

k_y = 0.05 μm^2 (homogeneous case)

P_{BP} = 14000 kPa

P_{oi} = 15500 kPa at 1450 m below sea level

P_{pc} = 5270 kPa

S_{gc} = 0.05

S_{or} = 0.3

S_{wc} = 0.3

T = 325 K

T_{pc} = 218 K

ϕ = 0.161 at 15500 kPa

E.2 Relative Permeabilities

$$k_{r_w} = (S_w - S_{wc})^2 / (1-S_{wc})^2 \qquad (E.1)$$

$$k_{r_o} = (S_o - S_{or})^2 / (1-S_{or} - S_{wc})^2 \qquad (E.2)$$

$$k_{r_g} = (S_g - S_{gc})^3 / (1 - S_{gc} - S_{wc} - S_{or})^3 \qquad (E.3)$$

$$k_{rg} \geq 0 \qquad (E.4)$$

E.3 Viscosities

$$\mu_w = 0.395 \tag{E.5}$$

$$\mu_g = 0.0096 + 6.43967 \times 10^{-7} * P \tag{E.6}$$

$$\mu_o = 0.459 - 6.6 \times 10^{-6} * P \quad , P < P_{BP} \tag{E.7}$$

$$\mu_o = 0.459 - 6.6 \times 10^{-6} * P_{BP} \quad , P \geq P_{BP} \tag{E.8}$$

E.4 Solubilities

$$R_{so} = 0.09457 + 5.5 \times 10^{-6} * P \quad , P < P_{BP} \tag{E.9}$$

$$R_{so} = 0.17157 \quad , P \geq P_{BP} \tag{E.10}$$

$$R_{sw} = 0 \tag{E.11}$$

E.5 Densities

$$\rho_o = \rho_o^* (1 - R_{so}) e^{C_o(P-100)} + R_{so}\rho_g - 85 \tag{E.12}$$

$$\rho_w = \rho_w^* e^{C_r(P-100)} \tag{E.13}$$

$$\rho_g = \frac{\rho_g^* T^* z^* P}{P^* T z} \tag{E.14}$$

The compressibility factor (z) is calculated from the following equations[8]:

$$z = P_r(0.1391T_r - 0.2988) + 0.0007T_r + 0.9969 \quad , P_r \leq 1.2 \tag{E.15}$$

$$z = P_r(0.0984T_r - 0.2053) + 0.0621T_r + 0.8580 \quad , 1.2 < P_r \leq 2.8 \tag{E.16}$$

$$z = P_r(-0.0284T_r + 0.0625) + 0.4714T_r - 0.0011 \quad , P_r > 2.8 \tag{E.17}$$

$$P_r = \frac{P}{P_{pc}} \tag{E.18}$$

$$T_r = \frac{T}{T_{pc}} \tag{E.19}$$

E.6 Capillary pressure data[10]

S_w	P_{cw}	S_o	P_{cg}
0.1	28.344	0.1	31.144
0.2	0.655	0.2	0.462
0.3	0.496	0.3	0.290
0.4	0.421	0.4	0.138
0.5	0.352	0.5	-0.007
0.6	0.283	0.6	-0.152
0.7	0.214	0.7	-0.296
0.8	0.145	0.8	-0.441
0.86	0.076	0.89	-0.586

E.7 Reservoir Figures

Figure

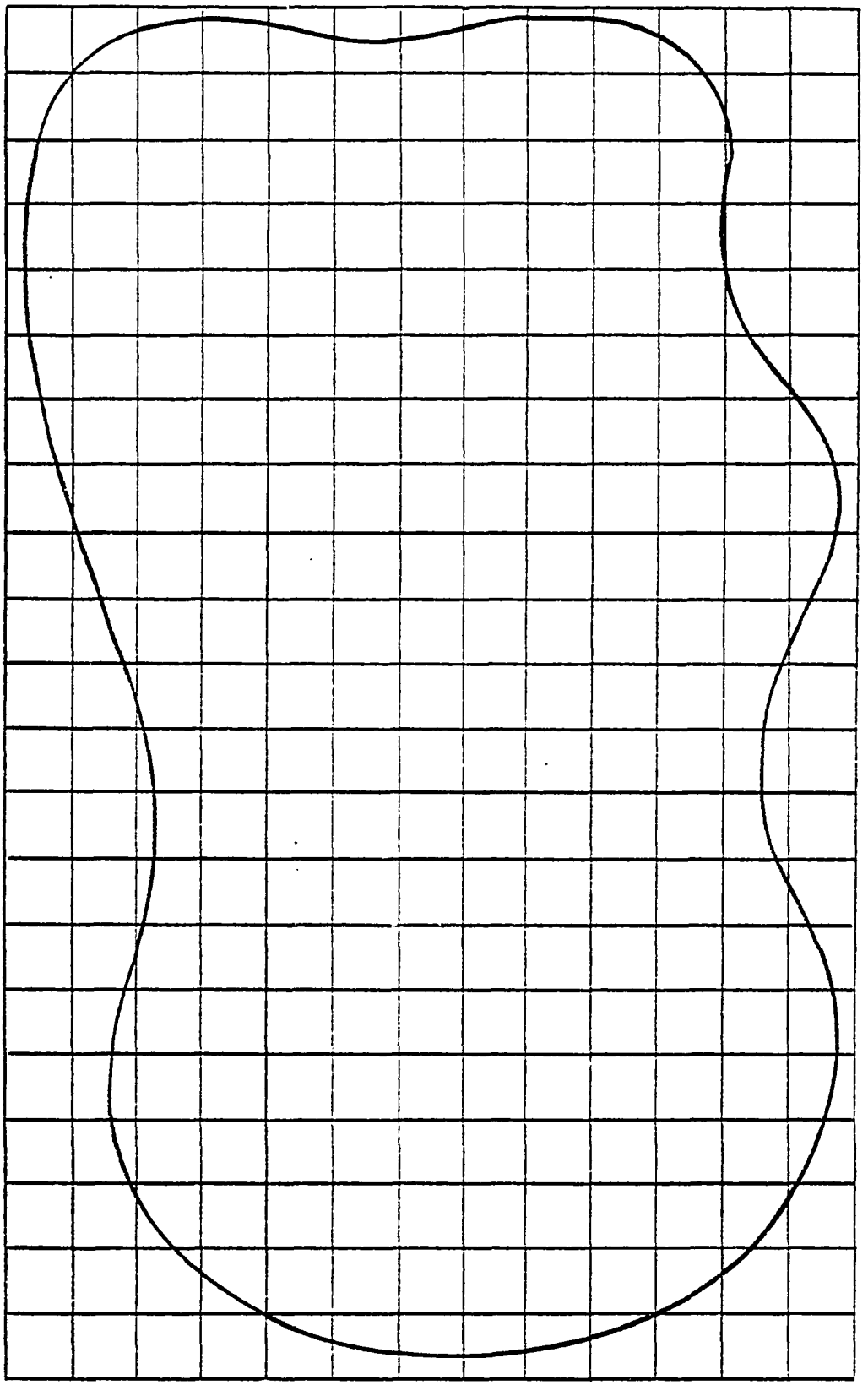


Figure E.1 - Discretized Reservoir System

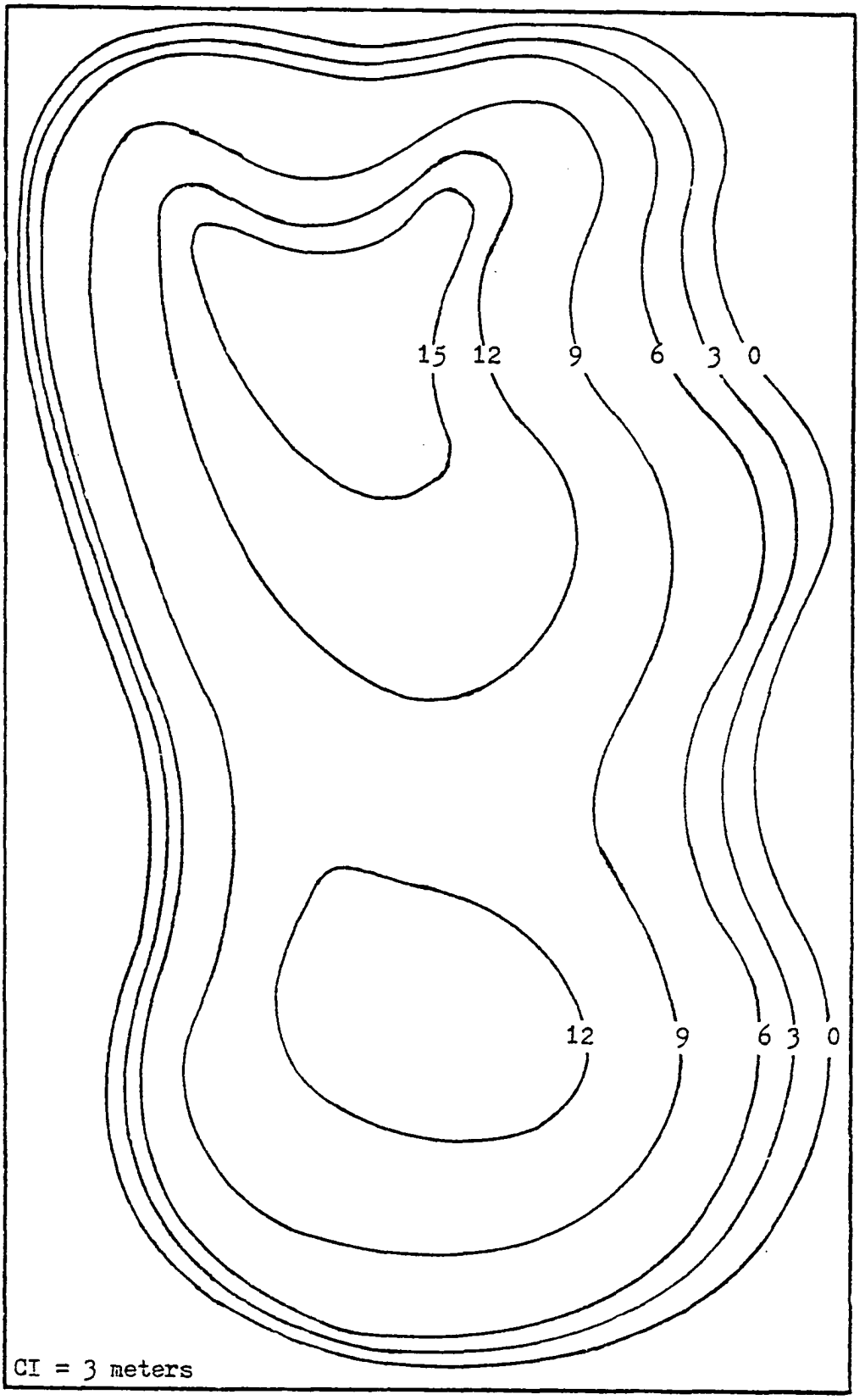


Figure E.2 - Reservoir Thickness(meters)

0	1	5	5	3	2	3	5	5	5	0	0	0
1	7	8	8	8	8	8	9	9	8	6	0	0
3	9	11	12	9	9	14	14	11	9	6	0	0
3	9	12	15	15	15	16	14	11	8	6	0	0
3	9	12	16	17	17	16	12	9	8	6	0	0
3	9	12	15	17	17	15	12	11	8	6	2	0
2	8	11	14	15	16	16	14	12	11	8	5	1
0	7	10	12	14	15	14	14	13	11	9	7	2
0	3	9	12	13	14	14	13	13	11	9	7	1
0	1	8	11	12	13	13	13	12	10	8	5	0
0	0	6	9	10	11	12	12	10	9	7	3	0
0	0	6	9	10	11	11	11	10	9	6	2	0
0	0	3	9	10	11	11	11	10	9	6	3	0
0	0	3	9	12	13	12	12	11	9	7	3	0
0	0	6	9	12	14	14	14	12	11	8	6	1
0	0	8	10	13	14	14	14	13	11	9	6	1
0	1	9	11	12	13	14	14	12	11	9	6	1
0	0	8	10	11	11	12	11	11	10	8	4	0
0	0	3	7	9	10	10	10	9	9	7	3	0
0	0	0	3	7	8	8	8	7	6	3	0	0
0	0	0	0	0	2	2	2	2	0	0	0	0

Figure E.3 - Digitized Reservoir Thickness (meters)

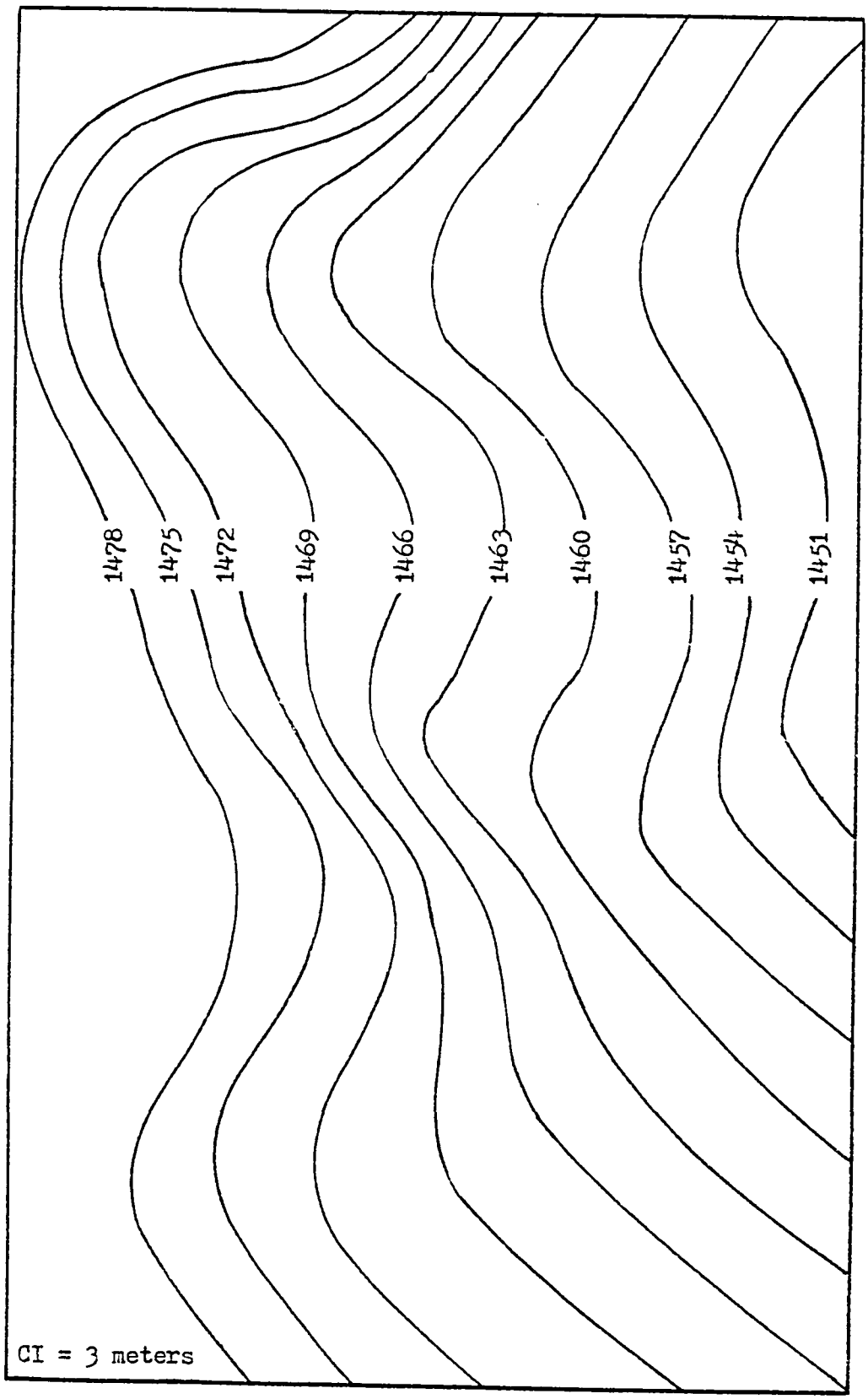


Figure E.4 - Top of the Reservoir Depth (meters below sea level)

0	1481	1481	1481	1479	1475	1471	1466	1463	1461	0	0	0
1481	1481	1479	1478	1476	1473	1468	1466	1464	1461	1457	0	0
1478	1477	1476	1474	1471	1469	1468	1466	1463	1460	1456	0	0
1478	1477	1475	1475	1472	1470	1468	1465	1462	1459	1456	0	0
1478	1477	1476	1476	1474	1471	1468	1464	1461	1459	1456	0	0
1479	1479	1477	1477	1475	1474	1471	1466	1463	1460	1457	1453	0
1480	1479	1478	1478	1476	1474	1473	1470	1466	1463	1459	1458	1452
0	1482	1480	1477	1476	1475	1472	1470	1469	1463	1461	1458	1452
0	1480	1480	1478	1476	1474	1472	1470	1469	1465	1462	1458	1452
0	1481	1481	1479	1475	1473	1472	1469	1467	1464	1461	1457	0
0	0	1481	1480	1476	1472	1469	1468	1465	1463	1460	1456	0
0	0	1482	1482	1479	1474	1470	1467	1464	1463	1459	1454	0
0	0	1483	1483	1481	1478	1472	1469	1465	1463	1459	1456	0
0	0	1483	1483	1482	1481	1476	1472	1468	1464	1461	1458	0
0	0	1482	1482	1481	1480	1476	1473	1470	1468	1463	1461	1458
0	0	1482	1481	1481	1479	1476	1473	1471	1468	1466	1463	1459
0	1482	1482	1481	1480	1478	1476	1474	1471	1470	1467	1464	1461
0	0	1481	1481	1479	1477	1476	1474	1473	1470	1468	1464	0
0	0	1479	1479	1478	1476	1475	1474	1473	1472	1469	1466	0
0	0	0	1479	1479	1477	1475	1474	1473	1471	1469	0	0
0	0	0	0	0	1476	1474	1472	1471	0	0	0	0

Figure E.5 - Digitized Middle of the Reservoir Depth(meters)

0	15700	15700	15700	15687	15661	15635	15603	15584	15571	0	0	0
15700	15700	15687	15680	15667	15648	15616	15603	15590	15571	15545	0	0
15680	15674	15667	15654	15635	15622	15616	15603	15584	15564	15539	0	0
15680	15674	15661	15661	15642	15629	15616	15597	15577	15558	15539	0	0
15680	15674	15667	15667	15654	15635	15616	15590	15571	15558	15539	0	0
15687	15687	15674	15674	15661	15654	15635	15603	15584	15564	15545	15519	0
15693	15687	15680	15680	15667	15654	15648	15629	15603	15584	15558	15551	15513
0	15706	15693	15674	15667	15661	15642	15629	15622	15597	15571	15551	15513
0	15693	15693	15680	15667	15654	15642	15629	15622	15597	15577	15551	15513
0	15700	15700	15687	15661	15648	15642	15622	15609	15590	15571	15545	0
0	0	15700	15693	15667	15642	15622	15616	15597	15584	15564	15513	0
0	0	15706	15706	15687	15654	15629	15609	15590	15584	15558	15526	0
0	0	15712	15712	15700	15680	15642	15622	15597	15584	15558	15539	0
0	0	15712	15712	15706	15700	15667	15642	15616	15590	15571	15551	0
0	0	15706	15706	15700	15693	15667	15648	15629	15616	15584	15571	15551
0	0	15706	15700	15700	15687	15667	15648	15635	15616	15603	15584	15558
0	15706	15706	15700	15693	15680	15667	15654	15635	15629	15609	15590	15571
0	0	15700	15700	15687	15674	15667	15654	15648	15629	15616	15590	0
0	0	15687	15687	15680	15667	15661	15654	15648	15642	15622	15603	0
0	0	0	15687	15687	15674	15661	15654	15648	15635	15622	0	0
0	0	0	0	0	15667	15654	15642	15635	0	0	0	0

Figure E.6 - Digitized Reservoir Initial Pressure(kilo-Pascal)

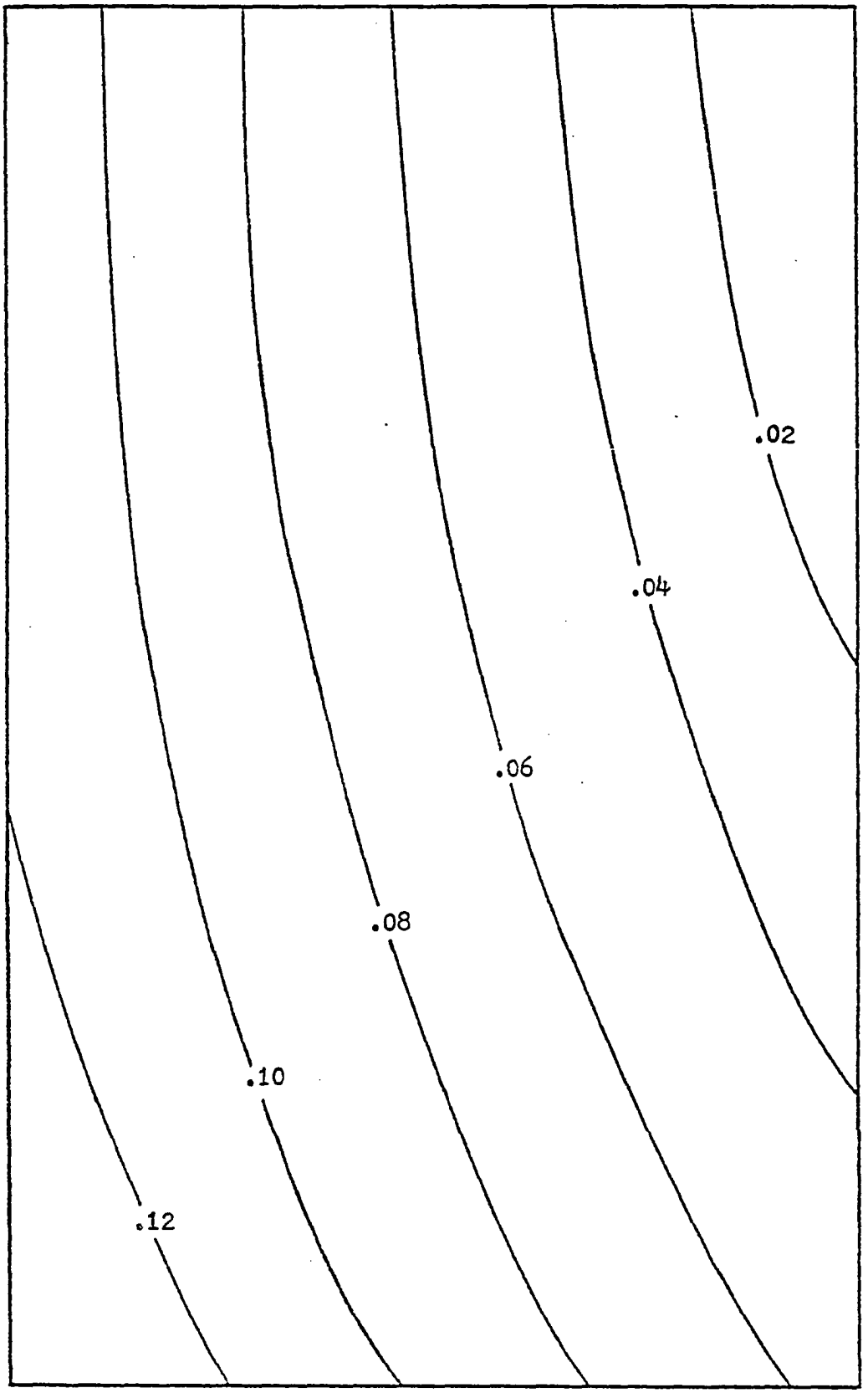


Figure E.7 - Reservoir Heterogeneous Permeability(micro-meter^2)

0	.10	.09	.08	.07	.06	.06	.05	.04	.03	0	0	0
.10	.10	.09	.08	.07	.06	.06	.05	.04	.03	.02	0	0
.10	.10	.09	.08	.07	.06	.06	.05	.04	.03	.02	0	0
.10	.10	.09	.08	.07	.06	.06	.05	.04	.03	.02	0	0
.10	.10	.09	.08	.08	.07	.06	.05	.04	.03	.02	0	0
.10	.10	.09	.08	.08	.07	.06	.05	.04	.04	.03	.02	0
.10	.10	.10	.09	.08	.07	.06	.05	.04	.04	.03	.02	.02
0	.10	.10	.09	.08	.07	.06	.06	.05	.04	.03	.02	.02
0	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04	.03	.02
0	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04	.03	0
0	0	.10	.09	.08	.08	.07	.06	.06	.05	.04	.03	0
0	0	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04	0
0	0	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04	0
0	0	.10	.10	.09	.08	.08	.07	.06	.05	.04	.04	0
0	0	.11	.10	.09	.08	.08	.07	.06	.06	.05	.04	.04
0	0	.11	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04
0	.12	.11	.10	.10	.09	.08	.07	.06	.06	.05	.04	.04
0	0	.11	.11	.10	.09	.08	.08	.07	.06	.06	.05	0
0	0	.11	.11	.10	.10	.09	.08	.07	.06	.06	.05	0
0	0	0	.11	.10	.10	.09	.08	.08	.07	.06	0	0
0	0	0	0	0	.10	.10	.09	.08	0	0	0	0

Figure E.8 - Digitized Reservoir Heterogeneous Permeability(micro-meter^2)

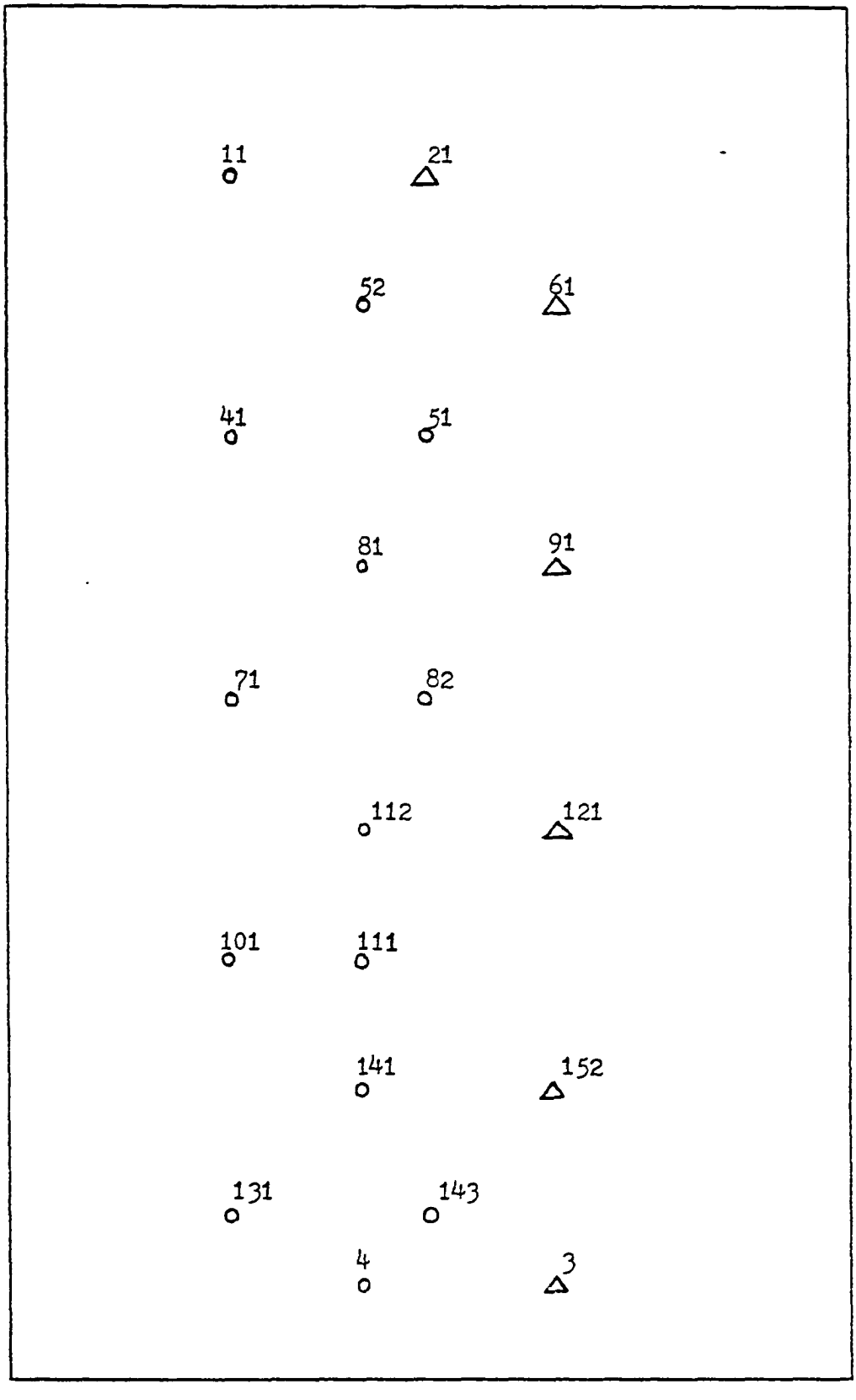


Figure E.9 - Configuration of Wells

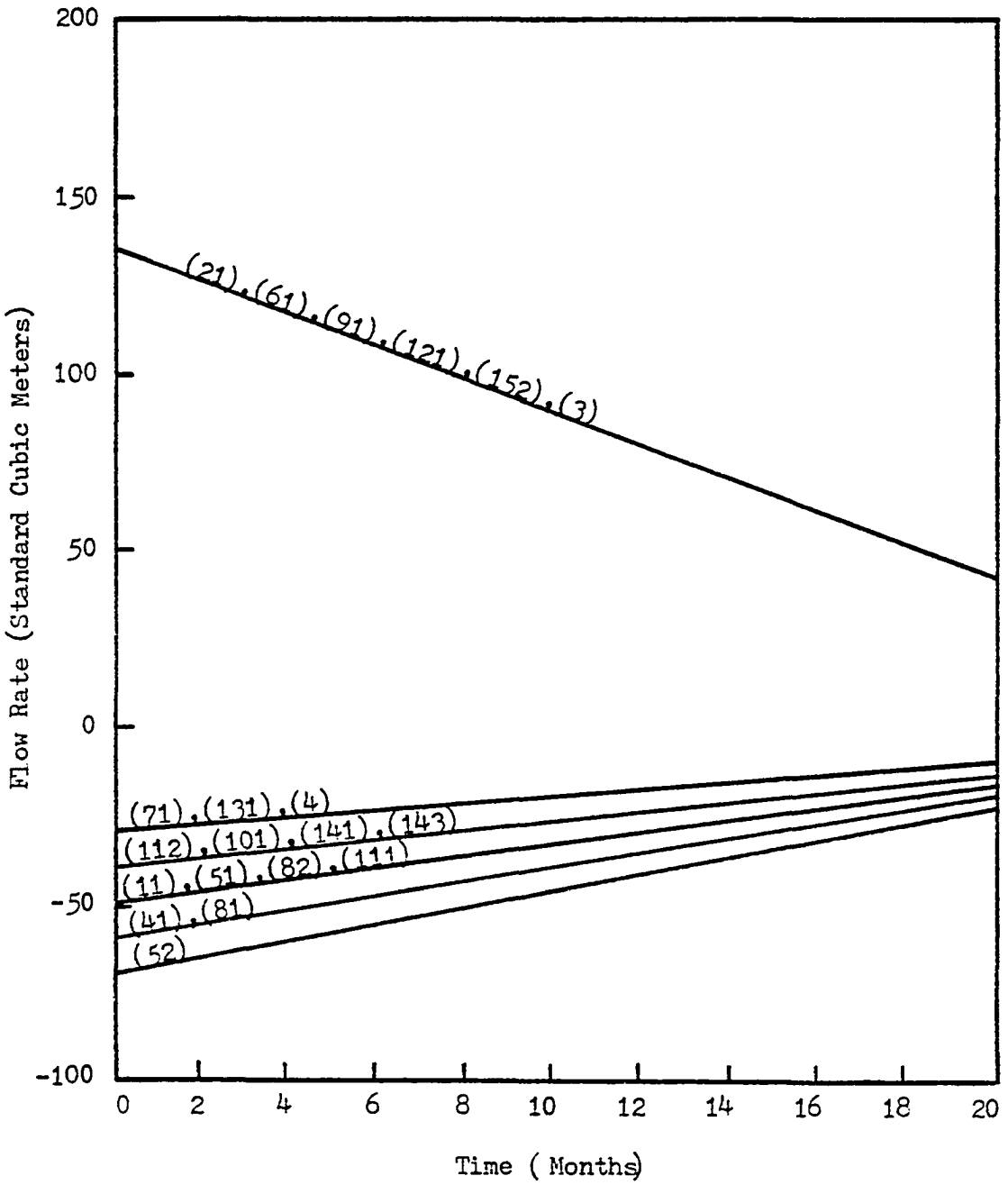


Figure E.10 - Production-Injection Flow Rates
(Well numbers are between parenthesis)

APPENDIX E

COMPUTER LISTING OF THE NUMERICAL PROCEDURES

```
      C-------------------------------------------------------------------MAR   10
      C     SUBJECT: TWO-DIMENSIONAL THREE-PHASE PETROLEUM RESERVOIR SIMULATORMAR   20
      C              A COMPUTER STUDY TO SUPPLEMENT A PH.D. DISSERTATION.    MAR   30
      C     LOCATION:    UNIVERSITY OF OKLAHOMA, NORMAN, OKLAHOMA, U.S.A.    MAR   40
      C     DEPARTMENT:  PETROLEUM & GEOLOGICAL ENGINEERING                  MAR   50
      C     COMPUTER:    IBM 360                                             MAR   60
      C     AUTHOR:      MUHAMMAD ALI AL-MARHOUN                             MAR   70
      C     ADVISOR:     H. B. CRICHLOW                                      MAR   80
      C     DATE:        SPRING 1978                                         MAR   90
      C-------------------------------------------------------------------MAR  100
0001        BLOCK DATA                                                      MAR  110
      C-------------------------------------------------------------------MAR  120
      C     PURPOSE OF SUBROUTINE                                           MAR  130
      C        INITIALIZATION                                               MAR  140
      C-------------------------------------------------------------------MAR  150
0002        COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)   MAR  160
0003        COMMON/C07/  IQ(345),QO(345),QW(345),QG(345),NW,INDW(100),PWF,RW MAR  170
0004        COMMON/C08/  PRMX(345),PRMY(345)                                MAR  180
0005        COMMON/C10/  HKX(345),HKY(345)                                  MAR  190
0006        COMMON/C11/  QOC(345),QWC(345),QGC(345),RT,ROC,RWC,RGC,NWN(100) MAR  200
0007        COMMON/C12/  POR(345),VOL(345),CO,CW,CR,TVOL                    MAR  210
0008        COMMON/C13/  RPRMO(345),RPRMW(345),RPRMG(345)                   MAR  220
0009        COMMON/C14/  SO(345),SW(345),SG(345),SWC,SOR,SGC,PBP            MAR  230
0010        COMMON/C15/  VISO(345),VISW(345),VISG(345)                      MAR  240
0011        COMMON/C16/  HLAMTY(345),HLAMTX(345),HLAMOX(345),HLAMOY(345),   MAR  250
           H             HLAMWX(345),HLAMWY(345),HLAMGX(345),HLAMGY(345)   MAR  260
0012        COMMON/C19/  DENO(345),DENW(345),DENG(345),DENOSC,DENWSC,DENGSC MAR  270
           D             ,DPH(345)                                         MAR  280
0013        COMMON/C20/  PCW(345),PCG(345),PCWS(10),SWS(10),PCGS(10),SOS(10) MAR 290
0014        COMMON/C21/  PSAVE(345),SOSAVE(345),SWSAVE(345),SGSAVE(345)    MAR  300
           O             ,DT(345),BO(345),BW(345)                          MAR  310
0015        COMMON/C22/  THETA,DT,AMAXDT,AMINDT                             MAR  320
0016        COMMON/C24/  QOCS(345),QWCS(345),QGCS(345)                      MAR  330
0017        COMMON/C27/  ISA(345),IAS(345),ML,NE,NB1,AA(16,175),CC(16,175)  MAR  340
0018        COMMON/C28/  IPS(390),M1,M2,ISS,AR(410),CB(410),EE(410),FF(410) MAR  350
0019        COMMON/C29/  G(17,195),HH(16,195),Y(410)                        MAR  360
0020        DATA         AR,CB,EE,FF,Y,G,HH,AA,CC/14085*0.0/                MAR  370
0021        DATA         E,A,B,C,F,D,QO,QW,QG,HKX,HKY,VOL,QOCS,QWCS,QGCS,   MAR  380
           1            HLAMTY,HLAMTX,HLAMOX,HLAMOY,HLAMWX,HLAMWY,HLAMGX,HLAMGYMAR  390
           2            ,PSAVE,DENO,DENW,DENG,PCW,PCG,SOSAVE,SWSAVE,SGSAVE, MAR  400
           3            QOC,QWC,QGC,RPRMO,RPRMW,RPRMG,VISO,VISW,VISG,SG,IQ  MAR  410
           4            /14490*0.0,345*0/                                  MAR  420
0022        DATA         DPH,POR,PRMX,PRMY,P,SO,SW/2415*0.0/                MAR  430
0023        DATA         AMAXDT,AMINDT,DT/0.0,1.0E 10,1.0/                  MAR  440
0024        END                                                             MAR  450
```

```
                C-------------------------------------------------------------------MAR  460
                C     SIMULATOR MAINLINE                                            MAR  470
                C-------------------------------------------------------------------MAR  480
0001                  COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)  MAR  490
0002                  COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                   MAR  500
0003                  COMMON/C03/  EPS,K,KMAX,RMAX,M,W,ICNVRG,BETA(90),GAMA(345),NNR(30)MAR 510
0004                  COMMON/C04/  TIM,TFINAL,DTMIN,DTMAX,DPLIM,DSLIM,OUTMOD,TPRINT  MAR  520
0005                  COMMON/C05/  EPSM,RMBMAX,NTRIAL,NTRMAX,AMIPO,AMIPW,AMIPG,TIMCPU,TMMAR 530
0006                  COMMON/C06/  DX(20),DY(25),DXM(20),DYM(25)                    MAR  540
0007                  COMMON/C07/  IO(345),QO(345),QW(345),QG(345),NW,INDW(100),PWF,RW MAR 550
0008                  COMMON/C08/  PRMX(345),PRMY(345)                              MAR  560
0009                  COMMON/C09/  H(345)                                           MAR  570
0010                  COMMON/C10/  HKX(345),HKY(345)                                MAR  580
0011                  COMMON/C11/  QOC(345),QWC(345),QGC(345),RT,ROC,RWC,RGC,NWN(100) MAR 590
0012                  COMMON/C12/  POR(345),VOL(345),CO,CW,CR,TVOL                  MAR  600
0013                  COMMON/C13/  RPRMO(345),RPRMW(345),RPRMG(345)                 MAR  610
0014                  COMMON/C14/  SO(345),SW(345),SG(345),SWC,SOR,SGC,PBP          MAR  620
0015                  COMMON/C15/  VISO(345),VISW(345),VISG(345)                    MAR  630
0016                  COMMON/C16/  HLAMTY(345),HLAMTX(345),HLAMOX(345),HLAMOY(345),  MAR  640
                     H             HLAMWX(345),HLAMWY(345),HLAMGX(345),HLAMGY(345)  MAR  650
0017                  COMMON/C17/  DXSUM ,DYSUM ,DXDX(20),DYDY(25)                  MAR  660
0018                  COMMON/C18/  ODENO(345),ODENW(345),OPOR(345)                  MAR  670
0019                  COMMON/C19/  DENO(345),DENW(345),DENG(345),DENOSC,DENWSC,DENGSC MAR 680
                     D             ,DPH(345)                                        MAR  690
0020                  COMMON/C20/  PCW(345),PCG(345),PCWS(10),SWS(10),PCGS(10),SOS(10) MAR 700
0021                  COMMON/C21/  PSAVE(345),SOSAVE(345),SWSAVE(345),SGSAVE(345)   MAR  710
                     O             ,OT(345),BO(345),BW(345)                         MAR  720
0022                  COMMON/C22/  THETA,DT,AMAXDT,AMINDT                           MAR  730
0023                  COMMON/C23/  NUD,NUDX(175),NLD,NLDX(175),NU,NUDP1,NLDP1,MM    MAR  740
0024                  COMMON/C24/  QOCS(345),QWCS(345),QGCS(345)                    MAR  750
0025                  COMMON/C25/  RSO(345),RSW(345)                                MAR  760
0026                  COMMON/C27/  ISA(345),IAS(345),ML,NE,NB1,AA(16,175),CC(16,175) MAR 770
0027                  COMMON/C28/  IRS(390),M1,M2,ISS,AR(410),CB(410),EE(410),FF(410) MAR 780
0028                  COMMON/C29/  G(17,195),HH(16,195),Y(410)                      MAR  790
0029                  COMMON/C30/  ALPHA(10)                                        MAR  795
0030                  CALL         INPUT                                            MAR  800
0031                  CALL         ONCE                                             MAR  810
                C-------------------------------------------------------------------MAR  820
                C     ATTENTION : CALL RS FOR RADP ONLY                             MAR  830
                C-------------------------------------------------------------------MAR  840
0032                  CALL         RS                                               MAR  860
0033                  PRINT  300                                                    MAR  870
0034                     NLEVEL    = 0                                            MAR  880
0035                     TIMCPU    = 0.0                                          MAR  890
0036               10    NTRIAL    = 0                                            MAR  900
0037                     NLEVEL    = NLEVEL + 1                                   MAR  910
0038                  PRINT  100,  NLEVEL                                           MAR  920
0039               20    NTRIAL    = NTRIAL + 1                                   MAR  930
0040                  CALL         TIME                                             MAR  940
0041                  CALL         COEFF                                            MAR  950
0042                  CALL         CPUTIM                                           MAR  960
                C-------------------------------------------------------------------MAR  970
```

```
      C     CALL ONE OF THE FOLLOWING SIX DIFFERENT METHODS                    MAR  980
      C------------------------------------------------------------------------MAR  990
0043        CALL          PSOR                                                 MAR 1000
0044        CALL          LSOR                                                 MAR 1010
0045        CALL          ADIP                                                 MAR 1020
0046        CALL          SIP                                                  MAR 1030
0047        CALL          STBAND(IND(1),IND(N))                                MAR 1040
0048        CALL          RADP                                                 MAR 1060
0049        CALL          CPUTIM(TINC)                                         MAR 1070
0050           TIMCPU     = TIMCPU + TINC                                      MAR 1080
0051        CALL          SATRN                                                MAR 1090
0052        CALL          MASERR                                               MAR 1100
0053        PRINT  200,  NTRIAL,DT,TIM,RT,RGC,RWC,RGC,K,RMAX,ICNVRG            MAR 1110
0054        IF(RMDMAX  .LE. EPSM ) GO TO 30                                    MAR 1120
0055        IF( NTRIAL .LT. NTRMAX) GO TO 20                                   MAR 1130
0056      30 IF((TPRINT-TIM) .GT. 0.0001*DTMIN ) GO TO 40                      MAR 1140
0057        CALL          OUTPUT                                               MAR 1150
0058      40 IF((TFINAL-TIM) .GT. 0.0001*DTMIN ) GO TO 10                      MAR 1160
0059        STOP                                                               MAR 1170
0060     100 FORMAT(10X,'NLEVEL =',I5       )                                 MAR 1180
0061     200 FORMAT(    28X,I10,2F10.2,4F10.4,I10, F10.4, I10)                MAR 1190
0062     300 FORMAT(1H1)                                                       MAR 1200
0063        END                                                                MAR 1210
```

```
                C-------------------------------------------------------------------MAR 1220
0001                  SUBROUTINE   PSOR                                             MAR 1230
                C-------------------------------------------------------------------MAR 1240
                C     PURPOSE OF SUBROUTINE                                         MAR 1250
                C        TO PERFORM POINT SUCCESSIVE OVERRELAXATION METHOD           MAR 1260
                C       .TO SOLVE THE PRESSURE EQUATIONS                            MAR 1270
                C-------------------------------------------------------------------MAR 1280
0002                  COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)  MAR 1290
0003                  COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                   MAR 1300
0004                  COMMON/C03/  EPS,K,KMAX,RMAX,M,W,ICNVRG,BETA(90),GAMA(345),NNR(30)MAR 1310
0005                     K         = 0                                             MAR 1320
0006                     ICNVRG    = 0                                             MAR 1330
0007                10   K         = K + 1                                         MAR 1340
0008                     RMAX      = 0.0                                           MAR 1350
                C-------------------------------------------------------------------MAR 1360
                C        COMPUTATION OF OVER-RELAXED PRESSURES                      MAR 1370
                C-------------------------------------------------------------------MAR 1380
0009                  DO 20       J = 1, N                                        MAR 1390
0010                     I         = IND(J)                                        MAR 1400
0011                     PS        = P(I)                                          MAR 1410
0012                     P(I)      =(D(I)-  E(I)*P(I-NX  )-  A(I)*P(I-1   )        MAR 1420
                      *                    -  C(I)*P(I+1   )-  F(I)*P(I+NX  ))/B(I)  MAR 1430
0013                     P(I)      = PS + W * (P(I) - PS)                          MAR 1440
0014                     R         = ABS((P(I)-PS)/P(I))                           MAR 1450
0015                  IF(  R .GT.   RMAX)   RMAX =  R                               MAR 1460
0016                20 CONTINUE                                                     MAR 1470
                C-------------------------------------------------------------------MAR 1480
                C        CHECK FOR CONVERGENCE                                      MAR 1490
                C-------------------------------------------------------------------MAR 1500
0017                  IF(  RMAX .LE. EPS) GO TO 30                                  MAR 1510
0018                  IF(K    .GE. KMAX ) GO TO 40                                  MAR 1520
0019                  GO TO 10                                                      MAR 1530
0020                30   ICNVRG    = 1                                             MAR 1540
0021                40 RETURN                                                       MAR 1550
0022                  END                                                           MAR 1560
```

```
0001            SUBROUTINE   LSOR                                                  MAR 1570
                C-------------------------------------------------------------------MAR 1580
                C      PURPOSE OF SUBROUTINE                                       MAR 1590
                C         TO PERFORM LINE  SUCCESSIVE OVERRELAXATION METHOD        MAR 1600
                C         TO SOLVE THE PRESSURE EQUATIONS                          MAR 1610
                C-------------------------------------------------------------------MAR 1620
0002            COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)      MAR 1630
0003            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                       MAR 1640
0004            COMMON/C03/  EPS,K,KMAX,RMAX,M,W,ICNVRG,BETA(9),GAMA(345),NNR(30)MAR 1650
0005            DIMENSION    PSAVE(345),DX(345)                                   MAR 1660
0006            DATA         DX/345*0.0/                                          MAR 1670
0007               ICNVRG    = 0                                                  MAR 1680
0008               K         = 0                                                  MAR 1690
0009         10    K         = K + 1                                              MAR 1700
0010               RMAX      = 0.0                                                MAR 1710
                C-------------------------------------------------------------------MAR 1720
                C         COMPUTATION OF OVER-RELAXED PRESSURES                    MAR 1730
                C-------------------------------------------------------------------MAR 1740
0011               JE        = 0                                                  MAR 1750
0012            DO 30     KK = 2, NYM1                                           MAR 1760
0013               JS        = JE + 1                                             MAR 1770
0014               JE        = JE + NNR(KK)                                       MAR 1780
0015            DO 20      J = JS , JE                                           MAR 1790
0016               I         = IND(J)                                             MAR 1800
0017               PSAVE(I)  = P(I)                                               MAR 1810
0018         20 DX(I)        = D(I)-E(I)*P(I-NX  )-F(I)*P(I+NX  )                 MAR 1820
0019            CALL         THOMSH(JS,JE,DX)                                     MAR 1830
0020            DO 30      J = JS , JE                                           MAR 1840
0021               I         = IND(J)                                             MAR 1850
0022               P(I)      = PSAVE(I)+W*(P(I)-PSAVE(I))                         MAR 1860
0023               R         = ABS((P(I)-PSAVE(I))/P(I))                          MAR 1870
0024            IF(  R .GT.   RMAX)   RMAX =  R                                  MAR 1880
0025         30 CONTINUE                                                         MAR 1890
                C-------------------------------------------------------------------MAR 1900
                C         CHECK FOR CONVERGENCE                                    MAR 1910
                C-------------------------------------------------------------------MAR 1920
0026            IF(  RMAX .LE. EPS) GO TO 40                                     MAR 1930
0027            IF(K     .GE. KMAX ) GO TO 50                                    MAR 1940
0028            GO TO 10                                                         MAR 1950
0029         40    ICNVRG    = 1                                                  MAR 1960
0030         50 RETURN                                                           MAR 1970
0031            END                                                              MAR 1980
```

```
0001            SUBROUTINE   THOMSH (JS , JE,R)                                  MAR 1990
                C-------------------------------------------------------------------MAR 2000
                C     PURPOSE OF SUBROUTINE                                       MAR 2010
                C        TO PERFORM THE THOMAS ALGORITHM ON A CLOSED BAND MATRIX   MAR 2020
                C-------------------------------------------------------------------MAR 2030
0002            COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)     MAR 2040
0003            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                       MAR 2050
0004            DIMENSION    R(NXY),BETA(345)                                     MAR 2060
0005            DATA         BETA/345*1.0/                                        MAR 2070
0006            JEP1         = JE+JS                                              MAR 2080
0007            I            = IND(JS)                                            MAR 2090
0008               BETA(I)   = B(I)                                               MAR 2100
0009            P(I)         = R(I)/B(I)                                          MAR 2110
0010            JSP1         = JS + 1                                             MAR 2120
0011            DO      20 J = JSP1,JE                                            MAR 2130
0012            I            = IND(J)                                             MAR 2140
0013            IM1          = I-1                                                MAR 2150
0014               BETA(I)   = B(I)-A(I)*C(IM1)/BETA(IM1)                         MAR 2160
0015         20 R(I)          = (R(I)-A(I)*R(IM1))/BETA(I)                       MAR 2170
0016            I            = IND(JE)                                            MAR 2180
0017            P(I)         = R(I)                                               MAR 2190
0018            DO      30 J = JSP1,JE                                            MAR 2200
0019            I            = IND(JEP1 - J)                                      MAR 2210
0020         30 P(I)          = R(I) - C(I)*P(I+1)/BETA(I)                       MAR 2220
0021            RETURN                                                           MAR 2230
0022            END                                                              MAR 2240
```

```
0001                  SUBROUTINE   ADIP                                                MAP 2250
                    C-------------------------------------------------------------------MAP 2260
                    C     PURPOSE OF SUBROUTINE                                         MAP 2270
                    C        TO PERFORM THE ALTERNATING DIRECTION IMPICIT PROCEDURE      MAP 2280
                    C        TO SOLVE THE PRESSURE EQUATIONS                            MAP 2290
                    C-------------------------------------------------------------------MAP 2300
0002                      COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)  MAP 2310
0003                      COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                   MAP 2320
0004                      COMMON/C03/  EPS,K,KMAX,RMAX,M,W,ICNVRG,BETA(90),GAMA(345),NNR(30)MAP 2330
0005                      COMMON/C06/  DX(20),DY(25),DXM(20),DYM(25)                    MAP 2340
0006                      COMMON/C16/  HLAMTY(345),HLAMTX(345),HLAMOX(345),HLAMCY(345), MAP 2350
                         H             HLAMWX(345),HLAMWY(345),HLAMGX(345),HLAMGY(345) MAP 2360
0007                      DIMENSION    BB(345),DD(345),PSAVE(345)                       MAP 2370
0008                      DATA         DD/345*0.0/                                      MAP 2380
0009                      DO 10        I = 1,NXY                                        MAP 2390
0010                         BB(I)     = 0.0                                            MAP 2400
0011                 10      DD(I)     = 0.0                                            MAP 2410
0012                         K         = 0                                              MAP 2420
0013                         KB        = 0                                              MAP 2430
0014                         ICNVRG    = 0                                              MAP 2440
0015                 30      K         = K + 1                                          MAP 2450
0016                         KB        = KB + 1                                         MAP 2460
0017                         RMAX      = 0.0                                            MAP 2470
0018                      DO 35        J = 1, N                                         MAP 2480
0019                         I         = IND(J)                                         MAP 2490
0020                 35      PSAVE(I)  = P(I)                                           MAP 2500
                    C-------------------------------------------------------------------MAP 2510
                    C     SELECTION    OF ACCELERATION PARAMETER                       MAP 2520
                    C-------------------------------------------------------------------MAP 2530
0021                      IF(KB .EQ. (M+1) ) KB = 1                                     MAP 2540
0022                         BET       = BETA(KB)                                       MAP 2550
                    C-------------------------------------------------------------------MAP 2560
                    C     X-DIRECTION SWEEP                                             MAP 2570
                    C-------------------------------------------------------------------MAP 2580
0023                      DO 40        J = 1,N                                          MAP 2590
0024                         I         = IND(J)                                         MAP 2600
0025                         BB(I)     = B(I) + E(I) + F(I) - GAMA(I)*BET               MAP 2610
0026                 40      DD(I)     = D(I)-E(I)*P(I-NX)+(E(I)+F(I)-GAMA(I)*BET)*P(I) MAP 2620
                        &                -F(I)*P(I+NX)                                  MAP 2630
0027                      CALL         TRIHOR(A,BB,C,DD,P)                              MAP 2640
                    C-------------------------------------------------------------------MAP 2650
                    C     Y-DIRECTION SWEEP                                             MAP 2660
                    C-------------------------------------------------------------------MAP 2670
0028                      DO 50        J = 1,N                                          MAP 2680
0029                         I         = IND(J)                                         MAP 2690
0030                         BB(I)     = B(I)+A(I) + C(I) - GAMA(I)*BET                 MAP 2700
0031                 50      DD(I)     = D(I)-A(I)*P(I-1)+(A(I)+C(I)-GAMA(I)*BET)*P(I)  MAP 2710
                        &                -C(I)*P(I+1)                                   MAP 2720
0032                      CALL         TRIVER(E,BB,F,DD,P)                              MAP 2730
                    C-------------------------------------------------------------------MAP 2740
                    C     CALCULATION OF RMAX                                           MAP 2750
                    C-------------------------------------------------------------------MAP 2760
```

```
0033              DO 60       J=1,N                                          MAR 2770
0034                 I         = IND(J)                                      MAR 2780
0035                 DIFF      = P(I) - PSAVE(I)                             MAR 2790
0036                 R         = ABS(DIFF /P(I))                             MAR 2800
0037              IF(  R .GT.   RMAX)   RMAX =  R                            MAR 2810
0038           60 CONTINUE                                                   MAR 2820
                C-------------------------------------------------------------MAR 2830
                C         CHECK FOR CONVERGENCE                               MAR 2840
                C-------------------------------------------------------------MAR 2850
0039              IF(  RMAX .LE. EPS) GO TO 70                               MAR 2860
0040              IF(K    .GE. KMAX ) GO TO 80                               MAR 2870
0041              GO TO  30                                                  MAR 2880
0042           70    ICNVRG    = 1                                           MAR 2890
0043           80 RETURN                                                     MAR 2900
0044              END                                                        MAR 2910
```

```
0001            SUBROUTINE   TRIHOR(A,B,C,R,P)                                  MAR 2920
              C-------------------------------------------------------------------MAR 2930
              C     PURPOSE OF SUBROUTINE                                        MAR 2940
              C        TO PERFORM THE THOMAS ALGORITHM ON A CLOSED BAND MATRIX   MAR 2950
              C-------------------------------------------------------------------MAR 2960
0002            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                     MAR 2970
0003            DIMENSION    A(NXY),B(NXY),C(NXY),R(NXY),P(NXY),BETA(345)       MAR 2980
0004            DATA         BETA/345*1.0/                                      MAR 2990
0005            NP1          = N + 1                                            MAR 3000
0006            I            = IND( 1)                                          MAR 3010
0007               BETA(I)   = B(I)                                             MAR 3020
0008            R(I)         = R(I)/B(I)                                        MAR 3030
0009            DO      20 J = 2, N                                             MAR 3040
0010            I            = IND(J)                                           MAR 3050
0011            IM1          = I-1                                              MAR 3060
0012               BETA(I)   = B(I)-A(I)*C(IM1)/BETA(IM1)                       MAR 3070
0013         20    R(I)      = (R(I)-A(I)*R(IM1))/BETA(I)                       MAR 3080
0014            I            = IND(N)                                           MAR 3090
0015            P(I)         = R(I)                                             MAR 3100
0016            DO      30 J = 2, N                                             MAR 3110
0017            I            = IND(NP1 - J)                                     MAR 3120
0018         30    P(I)      = R(I) - C(I)*P(I+1)/BETA(I)                       MAR 3130
0019            RETURN                                                          MAR 3140
0020            END                                                             MAR 3150
```

```
0001            SUBROUTINE   TRIVER(E,B,F,R,P)                                  MAR 3160
                C-------------------------------------------------------------------MAR 3170
                C     PURPOSE OF SUBROUTINE                                       MAR 3180
                C        TO PERFORM THE THOMAS ALGORITHM ON A WIDE   BAND MATRIX  MAR 3190
                C-------------------------------------------------------------------MAR 3200
0002            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                      MAR 3210
0003            DIMENSION    E(NXY),B(NXY),F(NXY),R(NXY),P(NXY),BETA(345)        MAR 3220
0004            DATA         BETA/345*1.0/                                       MAR 3230
0005            NP1          = N + 1                                             MAR 3240
0006            DO 20       J = 1,N                                              MAR 3250
0007            I            =IND(J)                                             MAR 3260
0008            IMNX         = I-NX                                              MAR 3270
0009               BETA(I)   =B(I)-E(I)*F(IMNX  )/BETA(IMNX)                     MAR 3280
0010         20    R(I)      =(R(I)-E(I)*R(IMNX  ))/ BETA(I)                     MAR 3290
0011            DO 40       J = 1,N                                             MAR 3300
0012            I            = IND(NP1 - J)                                      MAR 3310
0013         40    P(I)      =R(I)-F(I)*P(I+NX  )/BETA(I)                        MAR 3320
0014            RETURN                                                           MAR 3330
0015            END                                                              MAR 3340
```

```
0001            SUBROUTINE   SIP                                                    MAR 3350
              C-------------------------------------------------------------------MAR 3360
              C     PURPOSE OF SUBROUTINE                                          MAR 3370
              C        TO PERFORM STRONGLY IMPLICIT PROCEDURE                      MAR 3380
              C        TO SOLVE THE PRESSURE EQUATIONS                             MAR 3390
              C-------------------------------------------------------------------MAR 3400
0002            COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)     MAR 3410
0003            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                       MAR 3420
0004            COMMON/C03/  EPS,K,KMAX,RMAX,M,W,ICNVRG,BETA(90),GAMA(345),NNR(30)MAR 3430
0005            COMMON/C21/  PSAVE(345),SOSAVE(345),SWSAVE(345),SGSAVE(345)       MAR 3450
               0             ,OT(345),BO(345),BW(345)                           MAR 3460
0006            COMMON/C30/  ALPHA(10)                                           MAR 3465
0007            DIMENSION    EE(345),AA(345),CC(345),FF(345),Y(345),DPSAVE(345)  MAR 3470
0008            DATA         EE,AA,CC,FF,Y/1725*0.0/                             MAR 3490
0009            DO  5      J = 1,N                                               MAR 3760
0010               I         = IND(J)                                            MAR 3770
0011          5    D(I)      =D(I)-E(I)*PSAVE(I-NX)-A(I)*PSAVE(I-1 )-B(I)*PSAVE(I)MAR 3780
               1                    -C(I)*PSAVE(I+1 )-F(I)*PSAVE(I+NX)          MAR 3790
0012               ICNVRG    = 0                                                 MAR 3800
0013               K         = 0                                                 MAR 3810
0014               KZETA     = 0                                                 MAR 3820
0015         30    K         = K + 2                                             MAR 3830
0016               RMAX      = 0.0                                               MAR 3840
              C-------------------------------------------------------------------MAR 3850
              C     CALCULATION OF CURRENT ITERATION PARAMETER ALFA                MAR 3860
              C-------------------------------------------------------------------MAR 3870
0017            IF( MOD(KZETA ,M) .EQ. 0) KZETA = 0                              MAR 3880
0018               KZETA     = KZETA + 1                                         MAR 3890
0019               ALFA      = ALPHA(KZETA)                                      MAR 3900
              C-------------------------------------------------------------------MAR 3910
              C     ODD-NUMBERED ITERATION                                         MAR 3920
              C-------------------------------------------------------------------MAR 3930
0020            DO 40      J = 1, N                                              MAR 3940
0021               I         = IND(J)                                            MAR 3950
0022               IMNX      = I-NX                                              MAR 3960
0023               DPSAVE(I) = P(I)                                              MAR 3970
0024               IM1       = I-1                                               MAR 3980
0025               ALFAG     = ALFA*CC(IMNX)*E(I)/(CC(IMNX)*ALFA + 1.0)          MAR 3990
0026               ALFAH     = ALFA*FF(IM1 )*A(I)/(FF(IM1 )*ALFA + 1.0)          MAR 4000
0027               EE(I)     = E(I)-ALFAG                                        MAR 4010
0028               AA(I)     = A(I)-ALFAH                                        MAR 4020
0029               BB        = B(I)+ALFAG+ALFAH-EE(I)*FF(IMNX)-AA(I)*CC(IM1)     MAR 4030
0030               CC(I)     = (C(I)-ALFAG)/BB                                   MAR 4040
0031               FF(I)     = (F(I)-ALFAH)/BB                                   MAR 4050
0032         40    Y(I)      = (D(I)-EE(I)*Y(IMNX)-AA(I)*Y(IM1))/BB              MAR 4060
0033               NP1       = N + 1                                             MAR 4070
0034            DO 50      J = 1,N                                               MAR 4080
0035               I         = IND(NP1-J)                                        MAR 4090
0036         50    P(I)      = Y(I)-CC(I)*P(I+1)-FF(I)*P(I+NX)                   MAR 4100
              C-------------------------------------------------------------------MAR 4110
              C     EVEN-NUMBERED ITERATION                                        MAR 4120
              C-------------------------------------------------------------------MAR 4130
```

```
0037              DO 60       J = 1, N                                                MAR 4140
0038                 I          = IND(NP1-J)                                         MAR 4150
0039                 IPNX       = I + NX                                             MAR 4160
0040                 IP1        = I + 1                                              MAR 4170
0041                 ALFAG      = ALFA*EE(IP1 )*C(I)/(EE(IP1 )*ALFA + 1.0)           MAR 4180
0042                 ALFAH      = ALFA*AA(IPNX)*F(I)/(AA(IPNX)*ALFA + 1.0)           MAR 4190
0043                 FF(I)      = F(I)-ALFAH                                         MAR 4200
0044                 CC(I)      = C(I)-ALFAG                                         MAR 4210
0045                 BB         = B(I)+ALFAG+ALFAH-CC(I)*AA(IP1)-FF(I)*EE(IPNX)      MAR 4220
0046                 AA(I)      = (A(I)-ALFAH)/BB                                    MAR 4230
0047                 EE(I)      = (E(I)-ALFAG)/BB                                    MAR 4240
0048           60    Y(I)       = (D(I)-CC(I)*Y(IP1)-FF(I)*Y(IPNX))/BB               MAR 4250
0049              DO 70       J=1,N                                                   MAR 4260
0050                 I          = IND(J)                                             MAR 4270
0051           70    P(I)       = Y(I)-EE(I)*P(I-NX)-AA(I)*P(I-1)                    MAR 4280
                C-------------------------------------------------------------------MAR 4290
                C     CALCULATION OF RMAX                                           MAR 4300
                C-------------------------------------------------------------------MAR 4310
0052              DO 80       J=1,N                                                   MAR 4320
0053                 I          = IND(J)                                             MAR 4330
0054                 DIFF       = P(I) -DPSAVE(I)                                    MAR 4340
0055                 R          = ABS(DIFF /(P(I)+PSAVE(I)))                         MAR 4350
0056              IF(  R .GT.   RMAX)   RMAX =  R                                    MAR 4360
0057           80 CONTINUE                                                           MAR 4370
                C-------------------------------------------------------------------MAR 4380
                C        CHECK FOR CONVERGENCE                                      MAR 4390
                C-------------------------------------------------------------------MAR 4400
0058              IF(  RMAX .LE. EPS) GO TO 90                                       MAR 4410
0059              IF(K    .GE. KMAX ) GO TO 100                                      MAR 4420
0060              GO TO  30                                                          MAR 4430
0061           90    ICNVRG     = 1                                                  MAR 4440
0062          100 DO 110       J = 1,N                                              MAR 4450
0063                 I          = IND(J)                                             MAR 4460
0064          110    P(I)       = P(I) + PSAVE(I)                                    MAR 4470
0065              RETURN                                                             MAR 4480
0066              END                                                                MAR 4490
```

```
0001            SUBROUTINE   STBAND(IS,IE)                                    MAP 4500
                C-------------------------------------------------------------MAP 4510
                C     PURPOSE OF SUBROUTINE                                    MAP 4520
                C        TO PERFORM THE STBAND PROCEDURE                       MAP 4530
                C        TO SOLVE THE PRESSURE EQUATIONS.                      MAP 4540
                C-------------------------------------------------------------MAP 4550
0002            COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345) MAP 4560
0003            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                   MAP 4570
0004            DIMENSION    GAMA(345),ALFA(15),G(15,345),H(15,345)           MAP 4580
0005            DATA         GAMA/345*0.0/                                    MAP 4590
0006            DO 10      J = 1,NXY                                          MAP 4600
0007               G(NX,J)   = E(J)                                            MAP 4610
0008               G(1 ,J)   = A(J)                                            MAP 4620
0009               H(1 ,J)   = C(J)                                            MAP 4630
0010               H(NX,J)   = F(J)                                            MAP 4640
0011               P(J)      = 0.0                                             MAP 4650
0012            DO 10      I = 2,NXM1                                         MAP 4660
0013               G(I,J)    = 0.0                                           MAP 4670
0014         10    H(I,J)    = 0.0                                           MAP 4680
0015            DO 70      J = IS,IE                                          MAP 4690
0016            IF(ABS(B(J)) .LE. 1.0E-10 ) GO TO 70                          MAP 4700
0017               ALFA(NX)  = G(NX,J)                                         MAP 4710
0018            DO 20      II = 1,NXM1                                        MAP 4720
0019               I         = NX - II                                        MAP 4730
0020               ALFA(I)   = G(I,J)                                          MAP 4740
0021               IP1       = I+1                                            MAP 4750
0022            DO 20      K = IP1,NX                                         MAP 4760
0023         20    ALFA(I)   = ALFA(I)-ALFA(K)*H(K-I,J-K)                      MAP 4770
0024               BETA      = B(J)                                           MAP 4780
0025            DO 30      K = 1,NX                                           MAP 4790
0026         30    BETA      = BETA-ALFA(K)*H(K,J-K)                         MAP 4800
0027            DO 50      I = 1,NXM1                                         MAP 4810
0028               JPI       = J+I                                            MAP 4820
0029               IP1       = I+1                                            MAP 4830
0030            DO 40      K = IP1,NX                                         MAP 4840
0031         40    H(I,J)    = H(I,J)-ALFA(K-I)*H(K,JPI-K)                   MAP 4850
0032         50    H(I,J)    = H(I,J)/BETA                                   MAP 4860
0033               H(NX,J)   = H(NX,J)/BETA                                    MAP 4870
0034            DO 60      I = 1,NX                                           MAP 4880
0035               GAMA(J)   = D(J)                                            MAP 4890
0036            DO 60      K = 1,NX                                           MAP 4900
0037         60    GAMA(J)   = GAMA(J)-ALFA(K)*GAMA(J-K)                       MAP 4910
0038               GAMA(J)   = GAMA(J)/BETA                                    MAP 4920
0039            70 CONTINUE                                                   MAP 4930
0040            DO 80      JJ = IS,IE                                         MAP 4940
0041               J         = IE+IS-JJ                                       MAP 4950
0042               P(J)      = GAMA(J)                                        MAP 4960
0043            DO 80      K = 1,NX                                           MAP 4970
0044         80    P(J)      = P(J)-H(K,J)*P(J+K)                            MAP 4980
0045            RETURN                                                        MAP 4990
0046            END                                                           MAP 5000
```

```
0001            SUBROUTINE   RADP                                                  MAP 7250
                C-------------------------------------------------------------------MAP 7260
                C     PURPOSE OF SUBROUTINE                                         MAP 7270
                C        TO PERFORM THE RAD PROCEDURE                               MAP 7280
                C        TO SOLVE THE PRESSURE EQUATIONS.                           MAP 7290
                C-------------------------------------------------------------------MAP 7300
0002            COMMON/C01/  E(345),A(345),B(345),C(345),F(345),P(345),D(345)       MAP 7310
0003            COMMON/C23/  NUD,NUDX(175),NLD,NLDX(175),NU,NUDP1,NLDP1,M          MAR 7320
0004            COMMON/C28/  IRS(390),M1,M2,ISS,AA(410),CC(410),EE(410),FF(410)     MAR 7330
0005            COMMON/C29/  G(17,195), H(16,195),Y(410)                            MAP 7340
                C-------------------------------------------------------------------MAR 7350
                C        TRANSFER COEFFICIENT MATRIX FROM STANDARD ORDERING TO RAD   MAR 7360
                C        UPPER RIGHT QUARTER OF THE COMPOSITE MATRIX                MAP 7370
                C-------------------------------------------------------------------MAP 7380
0006            DO 10       K = 1,NUD                                               MAP 7390
0007               J          = NUDX(K)                                             MAR 7400
0008               I          = IRS(J)                                              MAP 7410
0009               AA(J)      = A(I)/B(I)                                           MAR 7420
0010               CC(J)      = C(I)/B(I)                                           MAR 7430
0011               EE(J)      = E(I)/B(I)                                           MAR 7440
0012               FF(J)      = F(I)/B(I)                                           MAR 7450
0013         10    Y(J)       = D(I)/B(I)                                           MAP 7460
0014            DO 20       K = 1,NLD                                               MAP 7470
0015               J          = NLDX(K)                                             MAR 7480
0016               I          = IRS(J)                                              MAR 7490
0017               AA(J)      = A(I)                                                MAR 7500
0018               G(M,J-NU)  = B(I)                                                MAR 7510
0019               CC(J)      = C(I)                                                MAP 7520
0020               EE(J)      = E(I)                                                MAP 7530
0021               FF(J)      = F(I)                                                MAR 7540
0022         20    Y(J)       = D(I)                                                MAR 7550
                C-------------------------------------------------------------------MAP 7560
                C        CONTRIBUTION OF AA,CC,EE,FF VECTORS                        MAP 7570
                C-------------------------------------------------------------------MAP 7580
0023            DO 30      JJ = 1, NLD                                             MAR 7590
0024               K          = NLDX(JJ)                                            MAR 7600
0025               J          = K - NU                                              MAP 7610
0026               J1         = J + 1                                               MAP 7620
0027               K1         = J + M1                                              MAR 7630
0028               K2         = J + M2                                              MAR 7640
0029               G( 1,J)    = -EE(K )*EE(J )                                      MAP 7650
0030               G( 2,J)    = -EE(K )*CC(J )-CC(K )*EE(J1)                        MAR 7660
0031               G( 3,J)    = -CC(K )*CC(J1)                                      MAR 7670
0032               G(M1,J)    = -EE(K )*AA(J )-AA(K )*EE(K2)                        MAR 7680
0033               G(M ,J)    = G(M,J)-EE(K )*FF(J )-CC(K )*AA(J1)                  MAP 7690
                1                        -AA(K )*CC(K2)-FF(K )*EE(K1)              MAP 7700
0034               H( 1,J)    = -FF(K1)*FF(K )                                      MAR 7710
0035               H( 2,J)    = -FF(K2)*AA(K )-AA(K1)*FF(K )                        MAP 7720
0036               H( 3,J)    = -AA(K2)*AA(K )                                      MAP 7730
0037               H(M1,J)    = -FF(J1)*CC(K )-CC(K1)*FF(K )                        MAR 7740
0038            DO 30       I = 4, M2                                               MAR 7750
0039               G(I ,J)    = 0.0                                                 MAR 7760
```

```
0040          30    H(I ,J)   = 0.0                                                  MAR 7770
              C-------------------------------------------------------------------MAR 7780
              C         GROUP NO. 1                                                 MAR 7790
              C-------------------------------------------------------------------MAR 7800
0041               DO 40      I = 1,3                                               MAR 7810
0042          40    H(I,ISS)  = H(I,ISS)/G(M,ISS)                                   MAR 7820
0043                H(M1,ISS) = H(M1,ISS)/G(M,ISS)                                  MAR 7830
0044               DO 60      JJ = 2,3                                              MAR 7840
0045                  J          = NLDX(JJ) - NU                                   MAR 7850
0046                  J0         = J - 1                                           MAR 7860
0047                  J2         = JJ + 2                                          MAR 7870
0048                  G(M,J)     = G(M,J)-G(M1,J)*H(M1,JJ)                         MAR 7880
0049                  H( 1,J)    = H( 1,J)/G( M,J)                                 MAR 7890
0050               DO 50      I = 2, J2                                             MAR 7900
0051          50    H(I,J)     = (H(I,J)-H(I-1,J0)*G(M1,J))/G(M,J)                MAR 7910
0052          60    H(M1,J)    = H(M1,J)/G(M ,J)                                   MAR 7920
              C-------------------------------------------------------------------MAR 7930
              C         GROUP NO. 2                                                 MAR 7940
              C-------------------------------------------------------------------MAR 7950
0053               DO 90      JJ = 4,NLD                                            MAR 7960
0054                  J          = NLDX(JJ)-NU                                     MAR 7970
0055                  JM         = J - M                                           MAR 7980
0056               DO 70      I = 2,M                                               MAR 7990
0057                  JMI        = JM + I                                          MAR 8000
0058                  KE         = I - 1                                           MAR 8010
0059               DO 70      K = 1,KE                                              MAR 8020
0060          70    G(I,J  )   = G(I,J  )-G(I-K,J  )*H(M-K,JMI-K)                  MAR 8030
0061                H( 1,J)    = H( 1,J)/G( M,J)                                 MAR 8040
0062               DO 90      I = 2 , M1                                            MAR 8050
0063                  KE         = I - 1                                           MAR 8060
0064               DO 80      K = 1, KE                                             MAR 8070
0065          80    H(I,J)     = H(I,J)-H(I-K,J-K)*G(M-K,J)                        MAR 8080
0066          90    H(I,J)     = H(I,J)/G(M,J)                                     MAR 8090
              C-------------------------------------------------------------------MAR 8100
              C         BACK SUBSTITUTION                                           MAR 8110
              C-------------------------------------------------------------------MAR 8120
0067               DO 110     K = 1,NLD                                             MAR 8130
0068                  J          = NLDX(K)                                         MAR 8140
0069                  JNU        = J - NU                                          MAR 8150
0070                  JM         = J - M                                           MAR 8160
0071                  Y(J)       = Y(J)-EE(J)*Y(JNU)-CC(J)*Y(JNU+1)                MAR 8170
                 1                 -AA(J)*Y(JNU+M2)-FF(J)*Y(JNU+M1)                MAR 8180
0072               DO 100     KK = 1, M1                                            MAR 8190
0073         100   Y(J)       = Y(J)-G(KK,JNU)*Y(JM+KK)                           MAR 8200
0074         110   Y(J)       = Y(J)/G(M,JNU)                                     MAR 8210
0075               DO 120     JJ = 2,NLD                                            MAR 8220
0076                  J          = NLDX(NLDP1-JJ)                                  MAR 8230
0077                  JM         = J + M                                           MAR 8240
0078                  JNU        = J - NU                                          MAR 8250
0079               DO 120     K = 1 , M1                                            MAR 8260
0080         120   Y(J)       = Y(J) - H(K,JNU) * Y(JM - K)                       MAR 8270
0081               DO 130     K = 1,NUD                                             MAR 8280
```

```
0082                J          = NUDX(NUDP1-K)                                  MAR 8290
0083                JNU        = J + NU                                         MAR 8300
0084          130   Y(J)       = Y(J) - FF(J)* Y(JNU)-AA(J)* Y(JNU-1)           MAR 8310
                 1                     - CC(J)* Y(JNU-M2)-EE(J)* Y(JNU-M1)     MAR 8320
                C--------------------------------------------------------------MAR 8330
                C         TRANSFORM PRESSURE TO STANDARD ORDERING              MAR 8340
                C--------------------------------------------------------------MAR 8350
0085                DO 140     K = 1,NUD                                        MAR 8360
0086                   J          = NUDX(K)                                     MAR 8370
0087          140   P(IRS(J)) = Y(J)                                            MAR 8380
0088                DO 150     K = 1,NLD                                        MAR 8390
0089                   J          = NLDX(K)                                     MAR 8400
0090          150   P(IRS(J)) = Y(J)                                            MAR 8410
0091                RETURN                                                      MAR 8420
0092                END                                                         MAR 8430
```

```
0001            SUBROUTINE   RS                                                    MAR 8440
              C-------------------------------------------------------------------MAR 8450
              C     PURPOSE OF SUBROUTINE                                         MAR 8460
              C        TO REORDER THE STANDARD ORDERING INTO RAD                  MAR 8470
              C-------------------------------------------------------------------MAR 8480
0002            COMMON/C02/  NX,NY,NXY,N,IND(345),NXM1,NYM1                       MAR 8490
0003            COMMON/C23/  NUD,NUDX(175),NLD,NLDX(175),NU,NUDP1,NLDP1,M         MAR 8500
0004            COMMON/C28/  IRS(390),M1,M2,ISS,AA(410),CC(410),EE(410),FF(410)   MAR 8510
0005            READ 300,    NUD,NLD,LD,ND                                      MAR 8520
0006               NR        = LD*ND                                            MAR 8530
0007               M         = LD + 2                                           MAR 8540
0008               NU        = NR /2                                            MAR 8550
0009            IF( MOD(ND, 2) .NE. 0 ) NU=NM/2+M                                MAR 8560
0010               M1        = M - 1                                            MAR 8570
0011               M2        = M - 2                                            MAR 8580
0012            READ 300,    (NUDX(J),IRS(NUDX(J)),J=1,NUD)                     MAR 8590
0013            READ 300,    (NLDX(J),IRS(NLDX(J)),J=1,NLD)                     MAR 8600
0014             30 PRINT 200                                                    MAR 8610
0015               DO 40      J = 1,NU                                           MAR 8620
0016               JNU        = J + NU                                           MAR 8630
0017             40 PRINT 100,  J,IRS(J),JNU,IRS(JNU)                           MAR 8640
0018               ISS        = NLDX(1) - NU                                     MAR 8650
0019               NLDP1      = NLD + 1                                          MAR 8660
0020               NUDP1      = NUD + 1                                          MAR 8670
0021            RETURN                                                          MAR 8680
0022           100 FORMAT(20X,2I10,10X,2I10)                                    MAR 8690
0023           200 FORMAT(1H1,27X,'RAD', 5X,'GAUSS',16X,'RAD', 5X,'GAUSS'/,     MAR 8700
                  1         28X,'===', 5X,'=====',16X,'===', 5X,'=====' )      MAR 8710
0024           300 FORMAT(20I4)                                                 MAR 8720
0025               END                                                          MAR 8730
```

APPENDIX F

A SAMPLE OF COMPUTER SIMULATION OUTPUT

	KILOGRAM	ST. CUBIC METER
OIL IN PLACE	0.1260E 10	0.1528E 07
WATER IN PLACE	0.1125E 10	0.9701E 06
GAS IN PLACE(FREE & DISSOLVED)	0.2162E 09	0.2402E 09
TOTAL FLUIDS IN PLACE	0.2602E 10	0.2427E 07

OUTPUT NO. 1

TIME	33.0000
CPU TIME	10.3230
MAXIMUM TIME STEP	13.0504
MINIMUM TIME STEP	0.0636
MAX. ABS. MASS BALANCE RELATIVE ERROR	0.1126E-02
TOTAL MASS BALANCE RELATIVE ERROR	-0.6004E-03
OIL MASS BALANCE RELATIVE ERROR	-0.1126E-02
WATER MASS BALANCE RELATIVE ERROR	0.8827E-04
GAS MASS BALANCE RELATIVE ERROR	-0.1123E-02
AVERAGE RESERVOIR PRESSURE	14814.

WELLS REPORT

NEGATIVE QUANTITY INDICATES PRODUCTION
POSITIVE QUANTITY INDICATES INJECTION
ALL UNITS ARE IN STANDARD CUBIC METERS

SER. NO.	WELL NO.	X-AXIS	Y-AXIS	OIL DAILY	OIL CUMULATIVE	GAS DAILY	GAS CUMULATIVE	WATER DAILY	WATER CUMULATIVE	GAS-OIL RATIO
1	11	5	4	-51.87	-1556.20	-8158.23	-0.24E 06	-0.00	-0.00	157.27
2	52	7	6	-73.41	-2202.17	-11544.68	-0.35E 06	-0.00	-0.00	157.27
3	41	5	8	-60.68	-1820.46	-9543.60	-0.29E 06	-0.00	-0.00	157.27
4	51	8	8	-51.87	-1556.20	-8158.23	-0.24E 06	-0.00	-0.00	157.27
5	81	7	10	-60.68	-1820.46	-9543.60	-0.29E 06	-0.00	-0.00	157.27
6	71	5	12	-33.28	-998.32	-5233.58	-0.16E 06	-0.00	-0.00	157.27
7	82	8	12	-51.87	-1556.20	-8158.23	-0.24E 06	-0.00	-0.00	157.27
8	112	7	14	-46.98	-1409.39	-7388.59	-0.22E 06	-0.00	-0.00	157.27
9	101	5	16	-39.15	-1174.49	-6157.16	-0.18E 06	-0.00	-0.00	157.27
10	111	7	16	-51.87	-1556.20	-8158.23	-0.24E 06	-0.00	-0.00	157.27
11	141	7	18	-46.98	-1409.39	-7388.59	-0.22E 06	-0.00	-0.00	157.27
12	131	5	20	-30.34	-910.23	-4771.79	-0.14E 06	-0.00	-0.00	157.27
13	143	8	20	-43.06	-1291.94	-6772.88	-0.20E 06	-0.00	-0.00	157.27
14	4	7	21	-29.36	-880.87	-4617.86	-0.14E 06	-0.00	-0.00	157.27
15	21	8	4	0.0	0.0	0.0	0.0	86.62	2598.70	
16	61	10	6	0.0	0.0	0.0	0.0	86.87	2606.22	
17	91	10	10	0.0	0.0	0.0	0.0	110.00	3300.00	
18	121	10	14	0.0	0.0	0.0	0.0	110.00	3300.00	
19	152	10	18	0.0	0.0	0.0	0.0	110.00	3300.00	
20	3	10	21	0.0	0.0	0.0	0.0	103.73	3112.03	
					-0.20E 05		-0.32E 07		0.18E 05	

OIL PRESSURE AT THE MIDDLE OF THE CELL

I / J	1	2	3	4	5	6	7	8	9	10
1	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0.	0.	14510.	14528.	14560.	14661.	14835.	15017.	15128.	15190.
3	0.	14490.	14495.	14515.	14534.	14669.	14842.	15058.	15126.	15181.
4	0.	14502.	14506.	14492.	14420.	14645.	14848.	15204.	15132.	15189.
5	0.	14489.	14495.	14508.	14511.	14615.	14725.	14920.	15059.	15210.
6	0.	14477.	14481.	14489.	14498.	14543.	14538.	14799.	15015.	15513.
7	0.	14452.	14445.	14451.	14442.	14518.	14580.	14720.	14925.	15117.
8	0.	14428.	14422.	14394.	14316.	14462.	14555.	14593.	14846.	15043.
9	0.	0.	14386.	14389.	14397.	14462.	14525.	14674.	14873.	15065.
10	0.	0.	14393.	14385.	14401.	14432.	14411.	14665.	14900.	15375.
11	0.	0.	14395.	14386.	14369.	14452.	14521.	14658.	14879.	15094.
12	0.	0.	0.	14349.	14282.	14427.	14529.	14574.	14859.	15072.
13	0.	0.	0.	14366.	14366.	14425.	14527.	14693.	14920.	15157.
14	0.	0.	0.	14368.	14381.	14401.	14404.	14710.	14968.	15631.
15	0.	0.	0.	14354.	14361.	14413.	14484.	14717.	14946.	15187.
16	0.	0.	0.	14330.	14254.	14419.	14427.	14722.	14948.	15144.
17	0.	0.	0.	14372.	14389.	14439.	14534.	14759.	14985.	15235.
18	0.	0.	14361.	14361.	14390.	14447.	14482.	14766.	15027.	15587.
19	0.	0.	0.	14386.	14370.	14470.	14585.	14774.	15038.	15306.
20	0.	0.	0.	14354.	14284.	14454.	14574.	14681.	15049.	15361.
21	0.	0.	0.	0.	14391.	14449.	14496.	14791.	15136.	16230.
22	0.	0.	0.	0.	0.	0.	14615.	14855.	15182.	15645.
23	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

I / J	11	12	13	14	15
1	0.	0.	0.	0.	0.
2	15223.	0.	0.	0.	0.
3	15226.	15266.	0.	0.	0.
4	15236.	15277.	0.	0.	0.
5	15250.	15280.	0.	0.	0.
6	15282.	15279.	0.	0.	0.
7	15195.	15245.	15283.	0.	0.
8	15152.	15222.	15242.	15294.	0.
9	15158.	15219.	15250.	15301.	0.
10	15222.	15230.	15261.	15309.	0.
11	15182.	15234.	15274.	0.	0.
12	15185.	15249.	15326.	0.	0.
13	15228.	15287.	15337.	0.	0.
14	15327.	15333.	15358.	0.	0.
15	15279.	15334.	15371.	0.	0.
16	15256.	15343.	15376.	15409.	0.
17	15338.	15360.	15394.	15429.	0.
18	15398.	15407.	15426.	15443.	0.
19	15401.	15439.	15469.	0.	0.
20	15445.	15482.	15491.	0.	0.
21	15612.	15552.	0.	0.	0.
22	0.	0.	0.	0.	0.
23	0.	0.	0.	0.	0.

OIL SATURATION

I / J	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.6996	0.6996	0.6997	0.6997	0.6998	0.6998	0.6998	0.6998
3	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6996	0.6998	0.6998
4	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6994	0.6999	0.6997	0.6998
5	0.0	0.6996	0.6996	0.6997	0.6997	0.6997	0.6997	0.6995	0.6998	0.6993
6	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6989	0.5244
7	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6997	0.6997	0.6998	0.6991
8	0.0	0.6996	0.6996	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6998
9	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6993
10	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6996	0.6997	0.6990	0.5451
11	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6994
12	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6998
13	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6986
14	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6982	0.5013
15	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6998	0.6988
16	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6997	0.6998	0.6998
17	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6992
18	0.0	0.0	0.6996	0.6996	0.6996	0.6996	0.6996	0.6997	0.6989	0.5329
19	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6993
20	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6997	0.6998	0.6963
21	0.0	0.0	0.0	0.0	0.6996	0.6996	0.6996	0.6997	0.6951	0.4432
22	0.0	0.0	0.0	0.0	0.0	0.0	0.6997	0.6998	0.6998	0.6940
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

I / J	11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0
2	0.6998	0.0	0.0	0.0	0.0
3	0.6998	0.6998	0.0	0.0	0.0
4	0.6998	0.6998	0.0	0.0	0.0
5	0.6998	0.6998	0.0	0.0	0.0
6	0.6998	0.6998	0.0	0.0	0.0
7	0.6998	0.6998	0.6998	0.0	0.0
8	0.6998	0.6998	0.6998	0.6998	0.0
9	0.6998	0.6998	0.6998	0.6998	0.0
10	0.6996	0.6998	0.6998	0.6998	0.0
11	0.6998	0.6998	0.6998	0.0	0.0
12	0.6998	0.6998	0.6998	0.0	0.0
13	0.6998	0.6998	0.6998	0.0	0.0
14	0.6990	0.6998	0.6998	0.0	0.0
15	0.6998	0.6998	0.6998	0.0	0.0
16	0.6998	0.6998	0.6998	0.6998	0.0
17	0.6998	0.6998	0.6998	0.6998	0.0
18	0.6995	0.6998	0.6998	0.6998	0.0
19	0.6998	0.6998	0.6998	0.0	0.0
20	0.6998	0.6998	0.6998	0.0	0.0
21	0.6968	0.6998	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0

WATER SATURATION

I / J	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.3004	0.3004	0.3003	0.3003	0.3002	0.3002	0.3002	0.3002
3	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3004	0.3002	0.3002
4	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3006	0.4141	0.3003	0.3002
5	0.0	0.3004	0.3004	0.3003	0.3003	0.3003	0.3003	0.3005	0.3002	0.3001
6	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3011	0.4756
7	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3003	0.3003	0.3002	0.3009
8	0.0	0.3004	0.3004	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3002
9	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3001
10	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3004	0.3003	0.3010	0.4549
11	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3006
12	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3002
13	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3014
14	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3018	0.4987
15	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3002	0.3012
16	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3003	0.3002	0.3002
17	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3008
18	0.0	0.0	0.3004	0.3004	0.3004	0.3004	0.3004	0.3003	0.3011	0.4671
19	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3007
20	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3003	0.3002	0.3007
21	0.0	0.0	0.0	0.0	0.3004	0.3004	0.3004	0.3003	0.3049	0.5598
22	0.0	0.0	0.0	0.0	0.0	0.0	0.3003	0.3002	0.3002	0.3060
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

I / J	11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0
2	0.3002	0.0	0.0	0.0	0.0
3	0.3002	0.3002	0.0	0.0	0.0
4	0.3002	0.3002	0.0	0.0	0.0
5	0.3002	0.3002	0.0	0.0	0.0
6	0.3007	0.3002	0.0	0.0	0.0
7	0.3002	0.3002	0.3002	0.0	0.0
8	0.3002	0.3002	0.3002	0.3002	0.0
9	0.3002	0.3002	0.3002	0.3002	0.0
10	0.3004	0.3002	0.3002	0.3002	0.0
11	0.3002	0.3002	0.3002	0.0	0.0
12	0.3002	0.3002	0.3002	0.0	0.0
13	0.3002	0.3002	0.3002	0.0	0.0
14	0.3010	0.3002	0.3002	0.0	0.0
15	0.3002	0.3002	0.3002	0.0	0.0
16	0.3002	0.3002	0.3002	0.3002	0.0
17	0.3002	0.3002	0.3002	0.3002	0.0
18	0.3005	0.3002	0.3002	0.3002	0.0
19	0.3002	0.3002	0.3002	0.0	0.0
20	0.3002	0.3002	0.3002	0.0	0.0
21	0.3032	0.3002	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0

GAS SATURATION

J	I = 1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

J	I = 11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0

OUTPUT NO. 2

TIME	60.0000
CPU TIME	13.7285
MAXIMUM TIME STEP	18.4000
MINIMUM TIME STEP	11.6000
MAX. ABS. MASS BALANCE RELATIVE ERROR	0.4186E-02
TOTAL MASS BALANCE RELATIVE ERROR	-0.2003E-02
OIL MASS BALANCE RELATIVE ERROR	-0.3502E-02
WATER MASS BALANCE RELATIVE ERROR	0.9373E-04
GAS MASS BALANCE RELATIVE ERROR	-0.4186E-02
AVERAGE RESERVOIR PRESSURE	13761.

WELLS REPORT

NEGATIVE QUANTITY INDICATES PRODUCTION
POSITIVE QUANTITY INDICATES INJECTION
ALL UNITS ARE IN STANDARD CUBIC METERS

SER. NO.	WELL NO.	X-AXIS	Y-AXIS	OIL DAILY	OIL CUMULATIVE	GAS DAILY	GAS CUMULATIVE	WATER DAILY	WATER CUMULATIVE	GAS-OIL RATIO
1	11	5	4	-50.77	-3079.32	-7984.80	-0.48E 06	-0.00	-0.00	157.27
2	52	7	6	-71.85	-4357.52	-11299.25	-0.69E 06	-0.00	-0.00	157.27
3	41	5	8	-59.39	-3602.22	-9340.71	-0.57E 06	-0.00	-0.00	157.27
4	51	8	8	-50.77	-3079.32	-7984.80	-0.48E 06	-0.00	-0.00	157.27
5	81	7	10	-59.39	-3602.22	-9340.71	-0.57E 06	-0.00	-0.00	157.27
6	71	5	12	-32.57	-1975.41	-5122.32	-0.31E 06	-0.00	-0.00	157.27
7	82	8	12	-50.77	-3079.32	-7984.80	-0.48E 06	-0.00	-0.00	157.27
8	112	7	14	-45.98	-2788.81	-7231.52	-0.44E 06	-0.00	-0.00	157.27
9	101	5	16	-38.32	-2324.01	-6026.26	-0.37E 06	-0.00	-0.00	157.27
10	111	7	16	-50.77	-3079.32	-7984.80	-0.48E 06	-0.00	-0.00	157.27
11	141	7	18	-45.98	-2788.81	-7231.52	-0.44E 06	-0.00	-0.00	157.27
12	131	5	20	-29.70	-1801.11	-4670.35	-0.28E 06	-0.00	-0.00	157.27
13	143	8	20	-42.15	-2556.41	-6628.89	-0.40E 06	-0.00	-0.00	157.27
14	4	7	21	-28.74	-1743.01	-4519.73	-0.27E 06	-0.00	-0.00	157.27
15	21	8	4	0.0	0.0	0.0	0.0	76.92	4906.17	
16	61	10	6	0.0	0.0	0.0	0.0	75.60	4874.20	
17	91	10	10	0.0	0.0	0.0	0.0	96.59	6197.56	
18	121	10	14	0.0	0.0	0.0	0.0	95.00	6150.13	
19	152	10	18	0.0	0.0	0.0	0.0	95.27	6158.12	
20	3	10	21	0.0	0.0	0.0	0.0	86.29	5700.63	
					-0.40E 05		-0.63E 07		0.34E 05	

OIL PRESSURE AT THE MIDDLE OF THE CELL

I	1	2	3	4	5	6	7	8	9	10
J										
1	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0.	0.	13487.	13504.	13532.	13626.	13820.	14002.	14110.	14168.
3	0.	13469.	13472.	13491.	13504.	13631.	13822.	14049.	14111.	14163.
4	0.	13481.	13485.	13467.	13381.	13604.	13814.	14375.	14121.	14168.
5	0.	13468.	13472.	13483.	13478.	13574.	13672.	13885.	14028.	14188.
6	0.	13455.	13458.	13462.	13465.	13501.	13479.	13769.	13971.	14930.
7	0.	13431.	13421.	13424.	13408.	13478.	13530.	13662.	13883.	14072.
8	0.	13406.	13398.	13365.	13280.	13420.	13505.	13530.	13804.	13997.
9	0.	0.	13358.	13357.	13360.	13418.	13472.	13611.	13825.	14017.
10	0.	0.	13361.	13349.	13359.	13383.	13343.	13595.	13835.	14721.
11	0.	0.	13361.	13347.	13321.	13399.	13459.	13585.	13822.	14337.
12	0.	0.	0.	13300.	13225.	13365.	13460.	13488.	13800.	14009.
13	0.	0.	0.	13307.	13304.	13354.	13450.	13605.	13846.	14096.
14	0.	0.	0.	13299.	13309.	13319.	13307.	13610.	13874.	15135.
15	0.	0.	0.	13276.	13281.	13326.	13388.	13612.	13853.	14097.
16	0.	0.	0.	13245.	13157.	13329.	13317.	13612.	13851.	14342.
17	0.	0.	0.	13286.	13299.	13339.	13423.	13665.	13873.	14095.
18	0.	0.	13269.	13268.	13292.	13339.	13356.	13659.	13892.	14909.
19	0.	0.	0.	13294.	13266.	13357.	13469.	13662.	13912.	14195.
20	0.	0.	0.	13255.	13172.	13333.	13442.	13528.	13912.	14248.
21	0.	0.	0.	0.	13276.	13320.	13349.	13661.	14002.	15716.
22	0.	0.	0.	0.	0.	0.	13471.	13729.	14052.	14578.
23	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

I	11	12	13	14	15
J					
1	0.	0.	0.	0.	0.
2	14198.	0.	0.	0.	0.
3	14200.	14239.	0.	0.	0.
4	14211.	14250.	0.	0.	0.
5	14224.	14251.	0.	0.	0.
6	14250.	14248.	0.	0.	0.
7	14154.	14204.	14238.	0.	0.
8	14137.	14177.	14193.	14244.	0.
9	14113.	14171.	14200.	14250.	0.
10	14187.	14181.	14238.	14257.	0.
11	14128.	14177.	14216.	0.	0.
12	14121.	14184.	14262.	0.	0.
13	14157.	14214.	14263.	0.	0.
14	14261.	14254.	14274.	0.	0.
15	14191.	14242.	14276.	0.	0.
16	14157.	14242.	14273.	14304.	0.
17	14204.	14255.	14286.	14321.	0.
18	14306.	14301.	14316.	14332.	0.
19	14292.	14328.	14355.	0.	0.
20	14332.	14368.	14375.	0.	0.
21	14494.	14417.	0.	0.	0.
22	0.	0.	0.	0.	0.
23	0.	0.	0.	0.	0.

OIL SATURATION

I	1	2	3	4	5	6	7	8	9	10
J										
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.6993	0.6993	0.6993	0.6994	0.6992	0.6995	0.6995	0.6995
3	0.0	0.6993	0.6993	0.6985	0.6985	0.6994	0.6994	0.6967	0.6995	0.6995
4	0.0	0.6993	0.6988	0.6985	0.6973	0.6994	0.6895	0.4932	0.6981	0.6995
5	0.0	0.6993	0.6993	0.6987	0.6993	0.6994	0.6994	0.6962	0.6995	0.6900
6	0.0	0.6993	0.6993	0.6993	0.6993	0.6993	0.6980	0.6946	0.6865	0.4169
7	0.0	0.6993	0.6993	0.6984	0.6993	0.6978	0.6988	0.6994	0.6978	0.6893
8	0.0	0.6993	0.6979	0.6993	0.6974	0.6993	0.6969	0.6994	0.6976	0.6995
9	0.0	0.0	0.6993	0.6990	0.6993	0.6984	0.6990	0.6994	0.6985	0.6913
10	0.0	0.0	0.6993	0.6993	0.6987	0.6993	0.6964	0.6994	0.6883	0.4376
11	0.0	0.0	0.6942	0.6935	0.6979	0.6977	0.6979	0.6994	0.6986	0.6911
12	0.0	0.0	0.0	0.6975	0.6992	0.6993	0.6970	0.6986	0.6974	0.6995
13	0.0	0.0	0.0	0.6985	0.6968	0.6993	0.6974	0.6994	0.6594	0.6798
14	0.0	0.0	0.0	0.6993	0.6963	0.6993	0.6965	0.6994	0.6761	0.3993
15	0.0	0.0	0.0	0.6983	0.6958	0.6993	0.6984	0.6994	0.6994	0.6817
16	0.0	0.0	0.0	0.6977	0.6958	0.6988	0.6967	0.6994	0.6979	0.6995
17	0.0	0.0	0.0	0.6966	0.6929	0.6990	0.6993	0.6939	0.6994	0.6890
18	0.0	0.0	0.6993	0.6971	0.6993	0.6993	0.6981	0.6978	0.6860	0.4254
19	0.0	0.0	0.0	0.6933	0.6975	0.6989	0.6993	0.6940	0.6995	0.6900
20	0.0	0.0	0.0	0.6964	0.6992	0.6993	0.6962	0.6994	0.6989	0.6583
21	0.0	0.0	0.0	0.0	0.6921	0.6993	0.6989	0.6932	0.6464	0.3664
22	0.0	0.0	0.0	0.0	0.0	0.0	0.6983	0.6940	0.6994	0.6232
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

I	11	12	13	14	15
J					
1	0.0	0.0	0.0	0.0	0.0
2	0.6995	0.0	0.0	0.0	0.0
3	0.6995	0.6995	0.0	0.0	0.0
4	0.6995	0.6995	0.0	0.0	0.0
5	0.6995	0.6995	0.0	0.0	0.0
6	0.6897	0.6995	0.0	0.0	0.0
7	0.6995	0.6995	0.6995	0.0	0.0
8	0.6995	0.6995	0.6995	0.6995	0.0
9	0.6995	0.6995	0.6995	0.6995	0.0
10	0.6932	0.6995	0.6995	0.6995	0.0
11	0.6995	0.6995	0.6995	0.0	0.0
12	0.6995	0.6995	0.6995	0.0	0.0
13	0.6995	0.6995	0.6995	0.0	0.0
14	0.6835	0.6995	0.6995	0.0	0.0
15	0.6995	0.6995	0.6995	0.0	0.0
16	0.6995	0.6995	0.6995	0.6995	0.0
17	0.6995	0.6995	0.6995	0.6995	0.0
18	0.6919	0.6995	0.6995	0.6995	0.0
19	0.6995	0.6995	0.6995	0.0	0.0
20	0.6995	0.6995	0.6995	0.0	0.0
21	0.6602	0.6995	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0

WATER SATURATION

I / J	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.3007	0.3007	0.3007	0.3006	0.3006	0.3005	0.3005	0.3005
3	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3003	0.3005	0.3005
4	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3058	0.3068	0.3019	0.3005
5	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3038	0.3005	0.3100
6	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3135	0.5831
7	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3005	0.3120
8	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3005
9	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3005	0.3087
10	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3005	0.3117	0.5624
11	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3005	0.3089
12	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3005	0.3005
13	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3202
14	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3239	0.6007
15	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3183
16	0.0	0.0	0.0	0.3007	0.3008	0.3007	0.3007	0.3005	0.3005	0.3005
17	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3006	0.3110
18	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3007	0.3006	0.3140	0.5746
19	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3007	0.3006	0.3005	0.3100
20	0.0	0.0	0.0	0.3007	0.3008	0.3007	0.3007	0.3006	0.3006	0.3417
21	0.0	0.0	0.0	0.0	0.3007	0.3007	0.3007	0.3006	0.3536	0.6336
22	0.0	0.0	0.0	0.0	0.0	0.0	0.3007	0.3006	0.3006	0.3768
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

I / J	11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0
2	0.3005	0.0	0.0	0.0	0.0
3	0.3005	0.3005	0.0	0.0	0.0
4	0.3005	0.3005	0.0	0.0	0.0
5	0.3005	0.3005	0.0	0.0	0.0
6	0.3103	0.3005	0.0	0.0	0.0
7	0.3005	0.3005	0.3005	0.0	0.0
8	0.3005	0.3005	0.3005	0.3005	0.0
9	0.3005	0.3005	0.3005	0.3005	0.0
10	0.3068	0.3005	0.3005	0.3005	0.0
11	0.3005	0.3005	0.3005	0.0	0.0
12	0.3005	0.3005	0.3005	0.0	0.0
13	0.3005	0.3005	0.3005	0.0	0.0
14	0.3165	0.3005	0.3005	0.0	0.0
15	0.3005	0.3005	0.3005	0.0	0.0
16	0.3005	0.3005	0.3005	0.3005	0.0
17	0.3005	0.3005	0.3005	0.3005	0.0
18	0.3081	0.3005	0.3005	0.3005	0.0
19	0.3005	0.3005	0.3005	0.0	0.0
20	0.3005	0.3005	0.3005	0.0	0.0
21	0.3398	0.3005	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0

GAS SATURATION

I / J	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0000	0.0	0.0012	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0008	0.0008	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0005	0.0008	0.0020	0.0	0.0047	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0006	0.0	0.0010	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0000	0.0013	0.0048	0.0	0.0
7	0.0	0.0	0.0	0.0009	0.0	0.0015	0.0035	0.0	0.0016	0.0
8	0.0	0.0	0.0014	0.0	0.0019	0.0	0.0024	0.0	0.0019	0.0
9	0.0	0.0	0.0	0.0002	0.0	0.0010	0.0003	0.0	0.0009	0.0
10	0.0	0.0	0.0	0.0003	0.0006	0.0	0.0029	0.0	0.0	0.0
11	0.0	0.0	0.0051	0.0058	0.0014	0.0016	0.0014	0.0	0.0009	0.0
12	0.0	0.0	0.0	0.0017	0.0000	0.0	0.0024	0.0007	0.0021	0.0
13	0.0	0.0	0.0	0.0008	0.0024	0.0	0.0019	0.0	0.0000	0.0
14	0.0	0.0	0.0	0.0	0.0030	0.0	0.0028	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0009	0.0035	0.0	0.0039	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0016	0.0034	0.0105	0.0026	0.0	0.0015	0.0
17	0.0	0.0	0.0	0.0027	0.0064	0.0003	0.0	0.0055	0.0	0.0
18	0.0	0.0	0.0	0.0022	0.0	0.0	0.0012	0.0016	0.0	0.0
19	0.0	0.0	0.0	0.0059	0.0017	0.0004	0.0	0.0054	0.0	0.0
20	0.0	0.0	0.0	0.0028	0.0	0.0	0.0032	0.0	0.0005	0.0
21	0.0	0.0	0.0	0.0	0.0072	0.0	0.0004	0.0061	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0011	0.0055	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

I / J	11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0