Multigrid Approach for Petroleum Reservoir Simulation on the Cray T3D

Kefei Wang, *David O. Ogbe*, and *Akanni S. Lawal*, Department of Petroleum Engineering, University of Alaska Fairbanks, Fairbanks, Alaska and *Donald J. Morton*, Arctic Region Supercomputing Center & Department of Mathematical Sciences, Cameron University, Lawton, Oklahoma

ABSTRACT: *An iterative multigrid solution is developed for solving large systems of fluid flow equations often encountered in petroleum reservoir engineering. The multigrid solver is coded into a parallelized version of a multi-phase, three-dimensional, black-oil reservoir simulator. Test problems from the petroleum engineering literature are studied to evaluate performance of the solver on the massively parallel Cray T3D.*

Results of simulations in the 4-, 8-, 16- ... 64-processor environment indicate substantial performance gains offered by the Cray T3D and the potential application of this approach in solving large reservoir engineering problems.

1 Introduction

Simulation of large reservoirs or entire fields containing several thousand grid blocks and as many as a thousand wells entails solution of very large sets of nonlinear equations. These equations which describe the chemical, physical and fluid flow processes taking place in the reservoir are solved to calculate several unknowns such as pressure, temperature and concentration of various hydrocarbon components at each time step per grid block. The computer resources required to solve the large set of equations governing multiphase flow in the reservoir can grow rapidly depending on the reservoir size, number of grid blocks, and the type of time stepping scheme used in the model [1].

The advances in parallel computing technology in the last decade hold great promise for engineers to conduct realistic large-scale reservoir simulations. Massively parallel computers such as the Cray T3D contain several hundreds of processors; hundreds of gigawords of memory, and the capability to perform billions of operations per second. The new generation of massively parallel processing systems can provide high performance at moderate cost required for large-scale reservoir simulation applications.

Although we have seen a 30-year history of petroleum reservoir simulation, the current versions of reservoir simulators were designed to run on single processor computers. There is the need to develop new software systems or modify existing software systems to take advantage of the sophisticated computing capability offered by the massively parallel processing systems. We need to develop some new solvers to take advantage of the new architectures of parallel machines.

In this work, we developed an efficient multigrid solver; and investigated the best way to integrate the solver into an existing petroleum reservoir simulator for large-scale simulations of fluid flow in oil reservoirs using a massively parallel processing computer. The investigation was conducted by porting an existing petroleum reservoir simulation model from a workstation to the Cray-T3D, developing a multigrid algorithm for solving the pressure equation on distributed memory parallel processors, and performing a simulation study of some test problems to validate the proposed approach.

2 The Multigrid Algorithm for Reservoir Simulation

Simulation of a multiphase flow in a petroleum reservoir often involves solving a system of nonlinear partial differential equations. The system of equations can be discretized by finite differencing to obtain:

$$\mathrm{L}u = b \tag{1}$$

where L is the differential operator, u is the unknown function, and b is the source term.

A number of iterative methods such as Gauss-Seidel, *LSOR* (Line Successive Over-relaxation), ADI (Alternating Direction Implicit), and *ORTHOMIN* have been successfully applied in reservoir simulation to solve equation 1.[3-5]. These iterative methods solve the discretized problem just by working on one grid block at a time, but the multigrid method solves the problem (equation 1) by working on a sequence of grids. The multigrid algorithm consists of three main operations: (1) transfer of information from the fine grid to the coarse grid using the restriction operator; (2) transfer of information from the coarse grid to the fine grid using the prolongation operator; and (3) attenuating the error using point relaxation. The restriction and prolongation operations are done by linear interpolation. Figure 1 is a schematic representation of the transfer of information between the fine grid and the coarse grid used in the multigrid method. The application of multigrid methods in petroleum simulation has been documented in the petroleum literature (See Refs. 1, 6-11). In this work, we present a brief overview of the multigrid methods to provide a better understanding of the parallelized multigrid algorithm developed in this work. Brandt [12], Hackbusch [13] and Wesseling[22] present detailed discussions on Multigrid methods. Here we will discuss the algorithm of the two- grid method.

2.1 Two-Grid Algorithm

The algorithm of the two-grid method consists of three steps: solving the linear system of equations to obtain an approximate solution on a fine-grid, correcting the residuals by translating information from the fine grid to the coarse-grid, and solving the linear system on the coarse-grid. The algorithm of one iterative step of the two- grid method is described as follows[13,14].

The essence of the two-grid method is to translate the discretized problem between the coarse-grid and the fine- grid. First, we compute an approximate solution to equation 1 on the fine-grid using a fast iterative method. Usually, the number of iterations in this step is limited to 3 in order to gain speed. Consider the two-dimensional grid shown in Fig. 2. To translate the solution vector from the fine-grid to coarse-grid, we define the **restriction operator**, $\left(I_h^H\right)_R$. That is,

$$\left(I_h^H\right)_R = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2}$$

Using the restriction operator, the translated value (coarse-grid) for an interior node, for example node point 13, can be determined:

$$u_H^{13} = \frac{1}{16}(4u_h^{13} + 2u_h^{14} + 2u_h^{18} + 2u_h^{12} + 2u_h^{8} + u_h^{19} + u_h^{17})$$

$$+ \frac{1}{16}(u_h^7 + u_h^9) \tag{3}$$

To transfer information of the discretized problem from the coarse-grid to the fine-grid, we define a **prolongation operator**, $(I_H^h)_P$. That is

$$\left(I_H^h\right)_P = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

Now the values of unknowns, $u_h^7,\ u_h^{13},\ u_h^8,\ u_h^6,$, can be determined using

$$u_h^7 = \frac{1}{4}(u_H^{13} + u_H^3 + u_H^{11} + u_H^1) \tag{5}$$

$$u_h^{13} = u_H^{13} \tag{6}$$

$$u_h^8 = \frac{1}{2}(u_H^{13} + u_H^3) \tag{7}$$

$$u_h^6 = \frac{1}{2}(u_H^{11} + u_H^1) \tag{8}$$

The combination of restriction operator and prolongation operator is called a two-grid iteration because two levels H (coarse-grid) and h (fine-grid) are involved.

2.2 Multigrid Method

The multigrid method is formed by applying the two-grid method to a sequence of grids of neighboring levels, translating information from the finest grid to the coarse grid (Hackbusch [13]). It is a recursive process. The detailed algorithm is discussed in Reference 14.

3 Analysis and Parallelization of The Simulator

BOAST is a multi-phase, three dimensional black-oil simulator developed by the **US Department of Energy (US DOE)** in 1982. The modified version, *BOAST-VHS* (hereafter called *BOAST*) used in this study was developed by Chang et al.[15]. *BOAST* contains only an IMPES (Implicit Pressure Explicit Saturation) formulation with direct elimination and *LSOR* solution options. In order to locate the numerically intensive segments, we ported *BOAST* to Cray YMP-M98 system of the Arctic Region Supercomputing Center (ARSC), and analyzed the source code of the simulator using Cray's performance tools[16]. Analysis of CPU time consumed by *BOAST*'s subprograms showed that three segments of the *FORTRAN* code accounted for about 88% of the CPU time(Fig. 2). These computationally intensive segments (*LSOR*, *LTR1* AND *SOLMAT*), which are used for solving the linearized multiphase flow equations, were selected as the candidates for parallelization to improve code performance.

The Cray-T3D system at the Arctic Region Supercomputing Center of University of Alaska Fairbanks contains 128 processors. This machine supports three programming styles: message passing, data sharing, and work sharing as part of the CRAFT programming model[16,17]. The message passing primitives are based on the Parallel Virtual Machine (PVM) model. The main function of data sharing programming model is to distribute the arrays over the memory of the processor elements (PEs) that will execute the user's programs. Work sharing is an implicit MPP programming method that simplifies the task of distributing the iterations in the program loops across all available processors[17,18]. In this study, both data sharing and work sharing MPP programming methods were used to parallelize the simulator, *BOAST*. Figure 4 shows the CRAFT's data sharing compiler directives used to distribute the arrays that calculate transmissibility in the model.

The multigrid solver consists of five subroutines: **GS_ITERATIVE** for Gauss-seidel iterative solution of linear equations; **RESTRICTION** for translating vectors from fine-grid level to the coarse-grid level; **PROLONGATION** for translating vectors from coarse-grid to fine-grid; **COARSE_MATRIX** to handle the elements of matrices generated from the coarse grid; and **DIRECT_SOLVER** to solve the coarse-grid equations by direct elimination. We used the data sharing and work sharing MPP programming methods to parallelize the multigrid solver. Figure 5 shows the CRAFT's model work sharing compiler directives used to distribute the iterations in the loops of the prolongation subprogram. After developing an interface program, the multigrid solver was introduced as a subroutine callable from *BOAST*. The parallelized *BOAST* was validated using test problems taken from the literature.

4 Test Problems and Evaluation of Code Performance

To analyze the performance of the black-oil reservoir simulator in Cray-YMP and Cray-T3D computing environments, two test problems were derived from the petroleum engineering literature. The results obtained from the simulation study are presented in the following sections.

4.1 Test Problem 1: Two Dimensional Horizontal Well Problem

The reservoir is represented by the 15 X 15 grid system shown in Fig. 6. The well passes through the grid-block centers and its entire length is open to flow. The reservoir data, fluid properties, relative permeability and capillary pressure data are presented in Reference 14. Table 1 shows the a comparison of the code performance (CPU time) for simulations conducted on the Cray-YMP (one PE) and the VAX 8800 using the iterative *LSOR* solver. The problem was simulated for 10 years. Notice that the Cray-YMP outperforms the VAX 8800 by a factor of 10 for this small test problem. We also compared the performance of the *LSOR* and parallelized multigrid solvers by running a 65 X 65 grid-block model of this test problem on the Cray Y-MP and the Cray T3D. The results of the code performance from the simulation runs are shown in Table 2. In this case, the performance ratio (CPU time on Cray Y- MP/CPU Time on the T3D) with the multigrid solver improved by a factor of 2 to 4 depending on the number of processor elements used in the Cray T3D's runs.

4.2 Test Problem 2: Three Dimensional Horizontal Well Problem

The second test problem deals with oil recovery by bottom water drive in a thin reservoir (Nghiem et. al. [19]). For this test problem, we also used the fluid properties and relative permeability data from the second Society of Petroleum Engineers (SPE) Comparative Simulation Project[20] The capillary-pressure data, relative permeability curves are taken from Reference 19. The reservoir is represented by a 9 X 9 X 6 grid system. Table 3 lists the initial reservoir data for the 6-layer model. Detailed discussion is given in Reference 14. This problem was simulated for 10 years using the VAX 8800, the Cray YMP and the Cray T3D. The pressure and oil saturation profiles shown in Figures 7 and 8 indicated that all three computers gave comparable results. In Table 4, we present the CPU times used to run the model with *LSOR* and Multigrid solvers. The data in Table 4 shows degraded performance (large CPU times) for this three-dimensional horizontal well problem. However, an analysis of the CPU time consumed by *BOAST* for runs made on the Cray-YMP (one PE) indicates that the multigrid solve outperforms *LSOR* by a factor of 6 for this problem.

4.3 Discussion of Performance in Parallel Environment

This section discusses the performance of the algorithms on the Cray-T3D for test problem 2. In the code performance analysis, we have evaluated the performance of the new multigrid solver against an optimized *LSOR* solver developed on the Cray-T3D. We consider the simulations made in the 8-processor environment of the Cray T3D. From the results (Table 4), we observed that the multigrid out-performs the *LSOR* solver by a factor of 4. Therefore, multigrid method is always preferred --it is the method of choice for multiphase reservoir simulation. Based on the results of the parallelization of *BOAST*, we recommend that domain decomposition algorithm be introduced into the modified *BOAST* simulator[21]. If we divide the three-dimensional problem formulated on the original domain into sub-problems on the sub-domains, and take the advantage of massively parallel processors of Cray-T3D to solve the sub-problems, then we can get a much higher efficiency (speedup) from the parallelized multigrid algorithm. To obtain additional speedups required for large-scale simulation of three-dimensional problems, we plan to parallelize the code using PVM, the explicit MPP programming model available on the Cray T3D.

5 Summary and Conclusions

In this study, numerical algorithms based on multigrid method are developed for solving a large system of sparse linear equations that arise in reservoir simulation. These algorithms are designed for application in scaleable distributed memory parallel computers. The multigrid algorithms were parallelized and written as subprograms in an IMPES-type black-oil reservoir simulator. Analysis of performance of the modified black-oil simulator showed that the multigrid solver is superior compared to a widely used sequential solver (*LSOR*) for test problems considered in this study. This algorithm showed a good performance and runs at a speedup factor of 4 compared to the *LSOR*. The following conclusions are derived from the results of this study:

1. On the basis of the CPU times consumed by the three-dimensional test problem simulated in this work, it is concluded that the multigrid solver out-performs *LSOR* solver.
2. The efficiency and performance of the multigrid solver increase as the size of the problems increases. The convergence of multigrid method is very fast. The rate of convergence does not deteriorate when the discretisation is refined, whereas classical iterative methods (e.g. *LSOR*) slow down as the grid size is decreased. Usually, the computational work is proportional to the number of unknowns, which is a function of the number of equations in the linear system and the grid size.
3. The results of test problems investigated in this work showed that the multigrid method is more suitable for solving multi-phase flow problems found in petroleum reservoir simulations. Application of the proposed algorithms and methodology to solve other problems in complex reservoirs would require defining the appropriate restriction and prolongation operators.

6 Acknowledgements

Financial support for this work was provided by Cray Research, Inc. as part of the University Research & Development Grant Program through the Arctic Region Supercomputing Center (ARSC). We appreciate the assistance on Cray T3D usage, code porting and parallelization received from Michael Ess and Ming Jiang of ARSC.

7 Nomenclature

b	=Right-hand side vector
BOAST	=Black-oil applied simulation tool
DOSHARED	= Work sharing compiler directive
$\left(I_h^H\right)_R$	=Restriction operator
$(I_H^h)_P$	=Prolongation operator
K	=Permeability
L	=Finite difference operator.
L_H	=Coarse-grid difference operator
L_h	=Fine-grid difference operator
P	=Pressure
Pc	=Capillary pressure
PE	=Processor element
PVM	=Parallel virtual machine
S	=Saturation
SHARED	=Data sharing compiler directive
u	=Vector of unknowns to be determined
$u_h^{(n)}$	=The initial iterative value on fine-grid
$\bar{u}_h^{(n)}$	=The approximate solution on fine-grid
v_h	=The residual value on fine-grid
v_H	=The residual value on coarse-grid
z	=Layer Thickness
Subscript	
g	=Gas index
h	=Fine-grid index
H	=Coarse-grid index
i, j, k	=Spatial indexes in the x-, y-, and z- directions
k	=Grid level index
o	=Oil index
P	=Prolongation operator index
R	=Restriction operator index
w	=Water index
Superscript	
h	=Fine-grid index
H	=Coarse-grid index
n	=The n-th iteration index
Greeks	
Δ	=incremental

8 References

1. Bhogeswara, R. and J.E. Killough (1993): "Parallel Linear Solvers for Reservoir Simulation: A Generic Approach for Existing and Emerging Computer Architecture," SPE 25240, paper presented at the 1993 Society of Petroleum Engineers Symposium on Reservoir Simulation, New Orleans, LA.
2. Li, K-M.G. (1993): A Linear Equation Solver for Massively Parallel Computers," SPE 25241, paper presented at the 1993 Society of Petroleum Engineers Symposium on Reservoir Simulation, New Orleans, Louisiana.
3. Aziz, K. and A. Settari (1979): *Petroleum Reservoir Simulation*, Applied Science Publishers, London.
4. Vinsome, P.K.W. (1976): "ORTHOMIN, an Iterative Method for Solving Sparse Banded Sets of Simultaneous Linear Equations," SPE 5729, paper presented at the SPE-AIME Fourth Symposium on Numerical Simulation of Reservoir Performance, Los Angeles.
5. Mattax, C.C. and R.L. Dalton (editors, 1990): *Reservoir Simulation*, Monograph Volume 13, Society of Petroleum Engineers, Richardson, TX.
6. Behie, A. and P.A. Forsyth, Jr. (1983): "Multigrid Solution of Pressure Equation in Reservoir Simulation," *Soc. Pet. Engr. J.*, Vol 23, 623-32.

7. Dendy, Jr., J.E., McCormick, S.F., Ruge, J.W., Russel, T.F. and S. Schaffer (1989): "Multigrid Methods for Three-Dimensional Petroleum Reservoir Simulation," SPE 18409, paper presented at the Tenth Society of Petroleum Engineers Reservoir Simulation Symposium, Houston, Texas.
8. Scott, T. (1985): "Multi-Grid Methods for Oil Reservoir Simulation," *J. Comp. Phys*. (June 1985) Vol. 59, No 2. 290-307.
9. Alcouffe, R.E. (1981): "The Multigrid Method for the Diffusion Equation with Strongly Discontinuous Coefficients," *SIAM J. Sci. Stat. Computing*, Vol 2, 430-454.
10. Meijerink, J.A., van Daalen, D.T., Hoogerbrugge, P.J., and R.J.A. Zeestraten: (1991): "Towards a more Effective Parallel Reservoir Simulator," SPE 21212, paper presented at the Eleventh Society of Petroleum Engineers Reservoir Simulation Symposium, Anaheim, California.
11. Killough, J.E., and Rao Bhogeswara (1991): "Simulation of Compositional Reservoir Phenomena on a Distributed-Memory Parallel Computer," *J. Pet. Tech*., Nov. 1991, 1386-94.
12. Brandt, A. (1977): "Multi-level Adaptive Solution to Boundary-value Problems," *Math. Comp*., Vol 31, 333-90.
13. Hackbusch, W. (1985): *Multigrid Methods and Applications*, Springer-Verlag, Berlin.
14. Wang, K (1995): "Multi-Grid Solution for Petroleum Reservoir Simulation on Distributed Memory Parallel Processors," M.S. Thesis, University of Alaska Fairbanks.
15. Chang, M.M., Sarathi, P., Heemstra, R.J., Cheng, A.M., and J.F. Pautz (1992): User's Guide and Documentation Manual for "*BOAST*-VHS for the PC", Topical Report NIPER-542.
16. Cray Research, Inc (1992): "Programming the Cray- T3D with Fortran SG-2509 DRAFT," The Manual for Cray-T3D Applications Programming.
17. Pase, D.M., MacDonald, T., and Meltzer, A.: M. (1994): "MPP Fortran Programming Model SN- 2513 2/94," Cray Research Inc., in M. Jiang MPP course notes, August 1994, Arctic Region Supercomputing Center, University of Alaska Fairbanks.
18. Koeninger, R. Kent, Furtney, M. and M. Walker (1994): "A Shared Memory MPP from Cray Research", *Digital Tech*. J. Vol. 6. No. 2. Spring 1994. 8-21.
19. Nghiem, L., Collins, D.A. and R. Sharma (1991): "Modeling Horizontal Wells in Reservoir Simulation, Seventh SPE Comparative Solution Project," SPE 21221, paper presented at the Eleventh Society of Petroleum Engineers Reservoir Simulation Symposium, Anaheim, California.
20. Weinstein, H.G., Chapplelear, J.E. and J.S. Nolen (1986): "Second Comparative solution Project:A Three-phase Coning Study," *J. Pet. Tech*., March 1986, 345-353.
21. Killough, J.E. and M.F. Wheeler (1987): "Parallel Iterative Linear Equation Solver: An Investigation of Domain Decomposition Algorithm for Reservoir Simulation," SPE 16201, paper presented at the 1987 SPE Symposium on Reservoir Simulation, San Antonio, Texas.
22. Wesseling, P. (1991): *An Introduction to Multi-Grid Methods*, John Wiley & Sons Ltd., England.

Table 1. Performance of BOAST Simulator on Various Computers (A 2-D Horizontal well Test Problem 1: Size = 15 X 15 X 1)

Computer System	Solver	CPU Time (Sec)	Wall-Clock Time(Sec)	Performance Ratio (CPU Time) VAX/Cray-YMP
VAX 8800	LSOR	150.02	302.2	1.0
Cray YMP (1-PE)	LSOR	15.16	60.14	9.9

Table 2. CPU Time Consumed by Modified BOAST Simulator for Test Problem 1 (Modified Test Problem 1: Size = 65

Computer System	No of CPUs	CPU Time (Sec)	Wall-Clock Time(Sec)	Performance Ratio (CPU Time)	Solver
				LSOR/Multigrid	
Cray YMP.	1.0	25.5	110.8	1.0	LSOR
Cray YMP.	1.0	11.23	45.67	2.3	Multigrid
			YMP-Multigrid/T3D-Multigrid		
Cray T3D.	8.0	6.23	7.1	1.8	Multigrid
Cray T3D.	32.0	4.56	5.26	2.5	Multigrid
Cray T3D.	64.0	3.14	4.01	3.6	Multigrid

Table 3. Reservoir Data of Test Problem 2: a 3-D Horizontal Well Problem (Source: Nghiem et al [19])

Layer	Thickness, Δz (ft)	Depth (ft)	Pressure p (psi)	Oil Sat, So	Water Sat., Sw
1	20.0	3600	3600	0.711	0.289
2	20.0	3620	3608	0.652	0.348
3	20.0	3640	3616	0.527	0.473
4	20.0	3660	3623	0.351	0.649
5	30.0	3685	3633	0.131	0.869
6	50.0	3725	3650	0	1

Horizontal Permeability, Kh = 300 (md) for all layers
Vertical Permeability, Kv = 30 (md) for all layers
Initial bubble-point pressure = gridblock initial oil pressure

Table 4. CPU Time Consumed by Modified BOAST Simulator for Test Problem 2 (Test Problem 2: A 3-D horizontal Well

Computer System	No of CPUs	CPU Time (Sec)	Wall-Clock Time(Sec)	Performance Ratio (CPU Time)		Solver
				LSOR/Multigrid	YMP-Multigrid/T3D -Multigrid	
Cray YMP.	1.0	6300	9650	1.0		LSOR
Cray YMP.	1.0	1023	1200	6.2		Multigrid
Cray T3D.	8.0	3300	3450	1	1.9	LSOR
Cray T3D.	8.0	867	920	3.8	1.2	Multigrid

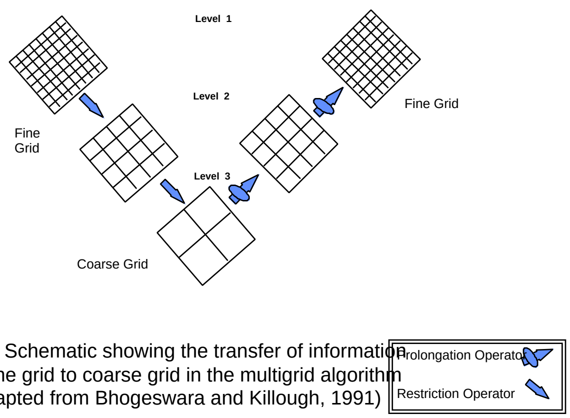

Fig 1 - Schematic showing the transfer of information from fine grid to coarse grid in the multigrid algorithm (Adapted from Bhogeswara and Killough, 1991)

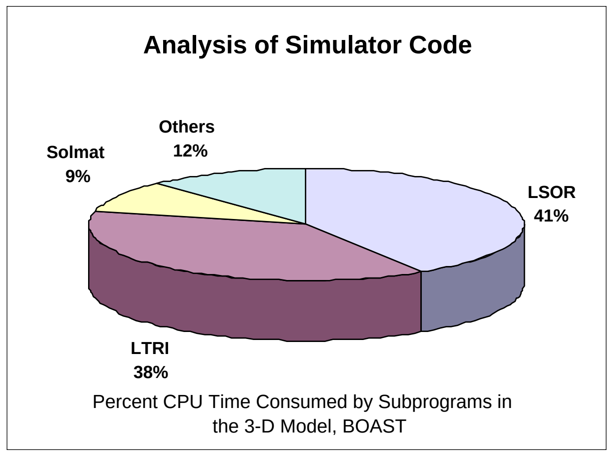

Fig 3 - Pre-parallelization Analysis of Simulator Co

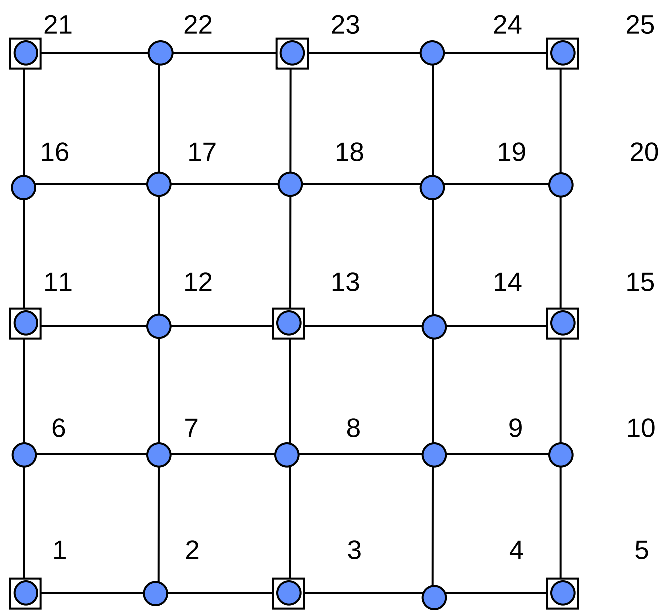

Fig 2 - Two Dimensional Grid with Coarse-grid overlaying the Fine-grid

```
          DIMENSION  AW(NX, NY, NZ), AE(NX, NY, NZ), AS(NX, NY, NZ),
C              AN(NX, NY, NZ), AT(NX, NY, NZ), AB(NX, NY, NZ),    E(NX, NY, NZ),
C                B(NX, NY, NZ)
CDIR$SHARED AW(:BLOCK, :, :), AE(:BLOCK, :, :), AS(:BLOCK, :, :),
C              AN(:BLOCK, :, :),AT(:BLOCK, :, :), AB(:BLOCK, :, :),    E(:BLOCK, :,
C               B(:BLOCK, :, :)
```

Fig. 4 - Data Sharing MPP Model for Distributing Arrays of Transmissibilit

```
  CDIR$DOSHARED (I) ON X(I, J, K)
        DO k = 2, kk1+1
        DO j = 2,  jj1+1
        DO I = 2,  ii1+1
          P1(2*i-2, 2*j-2, 2*k-2) = X(i, j, k)
        ENDDO
        ENDDO
        ENDDO

  CDIR$DOSHARED (I) ON P(I, J, K)
        DO k = 2, kk1+1
        DO j = 2,  jj1+1
        DO I = 2,  ii1+1
          P(i,  j,  k) = P(i, j, k) + P1(i, j, k)
        ENDDO
        ENDDO
       ENDDO
```

Fig. 5 - Work Sharing MPP Model for th
Prolongation Subprogram

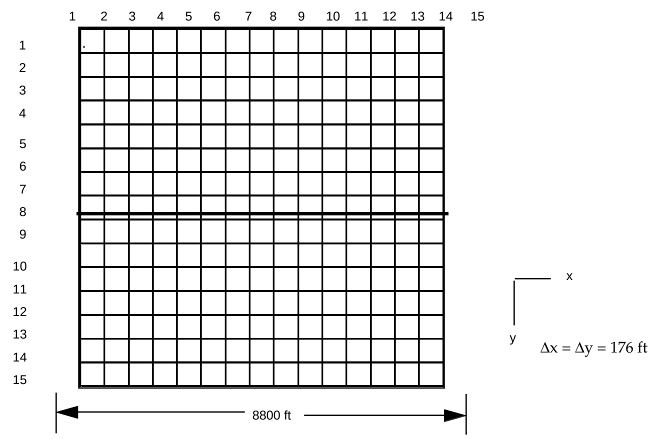

Fig 6 - Reservoir and Grid System of Test Problem 1
(Adapted from Data of Chang et al., 1992)

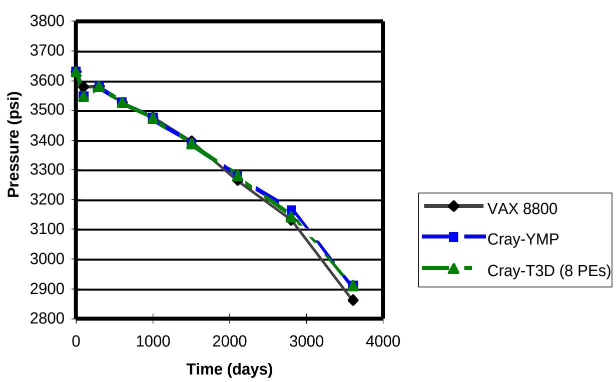

Fig 7 - Pressure - Time Profile for Test Problem 2

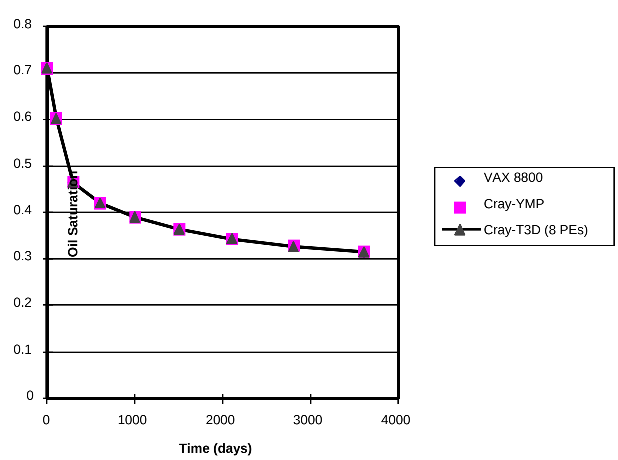

Fig 8 - Oil saturation - Time Profile for Test Problem 2