A NEW GENERATION CHEMICAL FLOODING SIMULATOR

Semi-annual Report for the Period
Oct. 1, 2002 – March 30, 2003

by
Gary A. Pope, Kamy Sepehrnoori, and Mojdeh Delshad

April 2003

Work Performed under Contract No. DE–FC-26-00BC15314

Sue Mehlhoff, Project Manager
U.S. Dept of Energy
National Petroleum Technology Office
One West Third Street, Suite 1400
Tulsa, OK 74103-3159

Prepared by
Center for Petroleum and Geosystems Engineering
The University of Texas at Austin
Austin, TX 78712

DISCLAIMER

ABSTRACT

The premise of this research is that a general-purpose reservoir simulator for several improved oil recovery processes can and should be developed so that high-resolution simulations of a variety of very large and difficult problems can be achieved using state-of-the-art algorithms and computers. Such a simulator is not currently available to the industry. The goal of this proposed research is to develop a new-generation chemical flooding simulator that is capable of efficiently and accurately simulating oil reservoirs with at least a million gridblocks in less than one day on massively parallel computers. Task 1 is the formulation and development of solution scheme, Task 2 is the implementation of the chemical module, and Task 3 is validation and application. We have made significant progress on all three tasks and we are on schedule on both technical and budget. In this report, we will detail our progress on Tasks 1 through 3 for the first six months of the second year of the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

In this report, we will detail our progress on Tasks 1 through 3 for the first six months of the second year of the project. We have formulated the mass conservation equation and physical properties for polymer. We implemented and validated the polymer module. We have also conducted a series of benchmarks by running the General Purpose Adaptive (GPAS) simulator on a Linux cluster and studied the scalability while using different interconnects. The results were very encouraging and indicated that GPAS performance scales linearly from one to 64 single processor nodes using a low latency, high-bandwidth such as Myrinet.

EXECUTIVE SUMMARY

The premise of this research is that a general-purpose reservoir simulator for several improved oil recovery processes can and should be developed so that high-resolution simulations of a variety of very large and difficult problems can be achieved using state-of-the-art algorithms and computers. Such a simulator is not currently available to the industry. The goal of this proposed research is to develop a new-generation chemical flooding simulator that is capable of efficiently and accurately simulating oil reservoirs with at least a million gridblocks in less than one day on massively parallel computers. Task 1 is the formulation and development of solution scheme, Task 2 is the implementation of the chemical module, and Task 3 is validation and application. We have made significant progress on all three tasks and we are on schedule on both technical and budget. In this report, we will detail our progress on Tasks 1 through 3 for the first six months of the second year of the project.

We have formulated, implemented, and validated the addition of polymer species to the general purpose adaptive (GPAS) simulator. The mass balance equation is solved by for polymer concentration in the water phase and relevant physical properties are altered as a function of polymer concentration. We have currently implemented a non-Newtonian viscosity model that is a function of polymer concentration, shear rate, and salt concentration. Other polymer properties implemented in GPAS are (1) inaccessible pore volume, (2) aqueous phase permeability reduction, and (3) polymer adsorption using a Langmuir isotherm. A few test cases were run to validate the formulation and implementation of polymer component in GPAS.

We have also conducted a series of benchmarks by running GPAS on Xeon-bases Linux cluster and studied the scalability while using different interconnects for the cluster. The simulations were performed for a gas injection process in a reservoir with 197,120 gridblocks and total of 88 wells in a staggered line drive pattern. The speed up results indicated a very good performance. The GPAS showed the best performance as well as scalability (speedup) on the cluster with Myrinet.

EXPERIMENTAL

This project does not include an experimental component.

RESULTS AND DISCUSSION

Task 1: Formulation and Development of Solution Scheme

The effort on this task was directed towards the formulation of polymer, electrolytes, and surfactant species in GPAS.

Mass Conservation Equation

The differential form for the species mass conservation equation is expressed as (Lake, 1989):

$$\frac{\partial}{\partial t}\left(\varphi\sum_{j=1}^{Np}\rho_j S_j\omega_{ij}+(1-\varphi)\rho_s\omega_{is}\right)+\vec{\nabla}\bullet\left(\sum_{j=1}^{Np}(\rho_j\omega_{ij}\overrightarrow{u_j}-\varphi S_j\rho_s\overrightarrow{\overrightarrow{K_{ij}}}\bullet\vec{\nabla}\omega_{ij})\right) \quad \textbf{(1)}$$
$$=R_i, i=1,..,Nc$$

where ρ_j is the mass density of phase j, S_j is the saturation of phase j, ω_{ij} is the mass fraction of species i in j, ρ_s is the mass density of the stationary phase s, $\overrightarrow{\overrightarrow{K_{ij}}}$ is the dispersion tensor, ω_{is} is the mass fraction of species i in the stationary phase s, $\overrightarrow{u_j}$ is the flux of phase j, and R_i is the source/sink term.

The species mass conservation equation was then simplified to model the aqueous species such as polymer and surfactant, and electrolytes (salt) flow using the below assumptions:

- Aqueous species do not occupy any volume
- Physical dispersion is neglected.
- Aqueous species stay in the aqueous phase and do not partition to the oil or gas phase.

Thus, the surfactant and polymer component modeling option will include the capability of a surfactant and polymer to change the physical properties of the fluids, to adsorb on the rock surfaces and to undergo chemical reactions. But these surfactant and polymer specific features are added as adjunct subroutines and thus they do not change the aqueous species flow modeling equation.

Applying these assumptions, the mass conservation equation **(1)** reduces to:

$$\frac{\partial}{\partial t}\left(\varphi\rho_1 S_1\omega_{i1} + (1-\varphi)\rho_s\omega_{is}\right) + \vec{\nabla}\bullet\left(\rho_1\omega_{i1}\overrightarrow{u_1}\right) = R_i \qquad \textbf{(2)}$$

The one point upstream weighting method is used in this finite-difference formulation. In the absence of the physical dispersion, all the dispersion is contributed by the numerical dispersion.

Task 2: Formulation And Implementation Of Chemical Module

Last year we reported on the formulation, implementation, and validation of tracer components as the first set of chemical species added to GPAS. Although all the formulation in the aqueous species modeling applies to tracer, polymer, electrolytes, and surfactant, each of these species modeling can involve, in addition, its own assumptions, formulations and special properties. Henceforth, the aqueous species refers to a species present in trace quantities in the aqueous phase and this includes a tracer, polymer or surfactant, the main subroutine XAQCOMP solving for the aqueous species concentration as a function of space and time is referred as the chemical subroutine and the whole module encompassing all the separate subroutines modeling the tracer, polymer and surfactant features is referred as the chemical module. The final output of the chemical subroutine XAQCOMP is the dimensionless aqueous species concentration.

In GPAS simulator, the chemical module was linked to the equation-of-state compositional model EOSCOMP in an explicit manner. After EOSCOMP solves for the pressures, saturations and compositions of the non-aqueous species components for a particular time step and the convergence for the mass balance equations is attained, the chemical subroutine obtains the required input from the host EOSCOMP and solves for the aqueous species mass balance equation to find the concentration at a given point in space and time.

In GPAS, the parameters such as the phase flux, phase saturation, phase density and upstream locations are already being calculated in EOSCOMP. Because the aqueous species do not alter the phase behavior, the equation-of-state flash calculations are performed only for the non-aqueous species and subsequently the phase saturations, phase densities and phase fluxes are determined. In order to make the code efficient, these fluid flow and rock parameters are transferred from EOSCOMP at the last Newtonian iteration of each timestep to the chemical subroutine. The grid-block pressures are no longer required as the other parameters have already accounted for the pressure term.

The simplified method used in GPAS is as follows:

- Transfer of phase saturations, phase densities, phase flux, upstream locations and the well molar flow rates from EOSCOMP to the chemical subroutine.
- Solution of the mass conservation equation for the aqueous species incorporating the convection terms and the source/sink terms.
- Return to Step 1 for a new time step

A notable feature of this method is that the chemical module calculations are not performed for each Newtonian iteration in a timestep, since the equations for aqueous species flow are decoupled and can be solved explicitly. This saves considerable computing time for reservoir simulations with a large number of gridblocks.

In this report, we discuss our progress on formulation and implementation of polymer component and associated physical properties.

Polymer physical property relationships

The polymer physical properties are based on those used in UTCHEM. These relationships have been carefully tested against experimental and published data and are used in most commercial simulators as well. Here is the list of polymer properties implemented in GPAS.

- Viscosity: polymer affects the aqueous phase viscosity as a function of polymer concentration, shear rate, and electrolyte concentrations.
- Inaccessible pore volume: inaccessible pore volume is modeled as a function of permeability and porosity
- Permeability reduction: aqueous phase permeability reduction is modeled as a function of polymer concentration, salinity, and permeability.
- Polymer adsorption: polymer adsorption as a function of polymer concentration and permeability is modeled using a Langmuir-type isotherm

Aqueous-phase viscosity

Most dilute polymer solutions exhibit the non-Newtonian behavior–i.e., viscosity is a function of shear rate. The shear-rate dependence of the polymer viscosity is modeled by Meter's equation (Meter and Bird, 1964):

$$\mu_p = \mu_w + \frac{\mu_p^0 - \mu_w}{1 + \left(\frac{\dot{\gamma}_{eq}}{\dot{\gamma}_{1/2}} \right)^{P_\alpha - 1}} \qquad \textbf{(3)}$$

where $\dot{\gamma}_{1/2}$ is the shear rate at which viscosity is the average of μ_p^0 and μ_w and P_α is an empirical coefficient. When the above equation is applied to flow in permeable media, μ_p is usually called apparent viscosity and the shear rate is an equivalent shear rate $\dot{\gamma}_{eq}$.

The in-situ shear rate for aqueous phase (phase 1) is modeled by the modified Blake-Kozeny capillary bundle equation for multiphase flow as

$$\dot{\gamma}_{eq} = \frac{\dot{\gamma}_c \mid \vec{u}_1 \mid}{\sqrt{\bar{k}k_{rl}\varphi S_l}} \tag{4}$$

The viscosity at a fixed shear rate is a function of polymer concentration and salinity and this dependency is modeled using the Flory-Huggins type equation (Flory, 1953)

$$\mu_p^0 = \mu_w (1 + (A_{p1}C_{4l} + A_{p2}C_{4l}^2 + A_{p3}C_{4l}^3)C_{SEP}^{S_p}) \tag{5}$$

where C_{4l} is the polymer concentration in the aqueous phase or microemulsion phase in the presence of surfactant species, μ_w is the water viscosity, and A_{p1}, A_{p2} and A_{p3} are constants and input to the simulator. S_p is the slope of $\frac{\mu_p^0 - \mu_w}{\mu_w}$ vs. C_{SEP} on a log-log plot.

C_{SEP} is the effective salinity for the polymer, calculated from

$$C_{SEP} = \frac{C_{51} + (\beta_p - 1)C_{61}}{C_{11}} \tag{6}$$

where C_{51}, C_{61}, and C_{11} are the anion, calcium, and water concentrations in the aqueous phase and β_p is an input parameter to the model.

The aqueous phase viscosity is modeled in the subroutine PROP. When the flag IFLAGP is turned on in the input file, the polymer section is activated. Polymer is treated as additional components in the aqueous phase. The aqueous species concentration, the variable CDTRA, in each grid block at a specific time is computed in

the subroutine XAQCOMP. The aqueous species concentration is stored as the polymer concentration, PLYC.

Inaccessible pore volume

High molecular weight polymer molecules usually flow faster than the solvent or smaller non-interacting components in the solution since the size of the polymer coil can approach the size of a pore throat in many reservoir rocks, which results in wall exclusion effects among other phenomena. This faster flow of polymer molecules can result in so-called inaccessible pore volume, which is commonly used to mean both pores that are inaccessible and also the higher velocity in accessible pores due to the large size of the polymer molecule lumped together.

Polymer inaccessible pore volume phenomenon depends on (1) characteristics of the porous media, especially its permeability, (2) polymer and electrolyte type, (3) polymer concentration, and (4) aqueous phase saturation.

The literature cites cases where as much as 30% of the pore volume may be inaccessible pore volume. The inaccessible pore volume is modeled by reducing the porosity in the component mass balance equation for polymer by multiplying with the input parameter E_{phi} defined below. The resulting effect is a faster polymer velocity than the velocity of a reference aqueous phase tracer.

Some of the possible reasons for the inaccessible pore volume effect as follows (Camilleri *et al*., 1987):

- Pore-wall exclusion: Pore-wall exclusion of the polymer molecules relative to the very small molecules such as water that make up the polymer solvent

- Rheological properties: the shear rate is considerably different in different parts of a given pore and also from pore to pore.
- Amount and type of retention: an adsorbed polymer coil effectively excludes a certain volume of the pore to further penetration by a mobile polymer coil.

A porosity correction factor is used to model the lumped effect. The effective porosity for the polymer is modeled as

$$\varphi_{polymer} = \varphi_{actual} E_{phi} \quad \textbf{(7)}$$

where E_{phi} is the effective porosity factor to be multiplied with the actual porosity to get the apparent porosity for the polymer. This modified porosity is used in the aqueous species conservation equation resulting in a faster velocity of polymer molecules.

Permeability reduction

Some water-soluble polymer solutions reduce both the mobility of the aqueous phase in low to moderate permeability rocks. This is because the polymer adsorb on the porous medium, or is trapped in small pore throats, or is entangled in small pores or all of these in some cases, and thus reduces the aqueous mobility further than that caused by the increase in aqueous viscosity alone. This effect is called permeability reduction and can be significant for polymers such as hydrolyzed polyacrylamide that is used in most polymer flood field applications, but typically is not significant for biopolymers such as xanthan gum. The permeability to oil is not reduced, so the effect on the fractional flow of oil is beneficial i.e. relatively more oil than water flows.

The permeability reduction is modeled by the permeability reduction factor, R_k defined as the ratio of the effective permeability of water to the effective permeability of polymer. The assumptions made for the model are:

- The polymer phase alone is affected by this reduction in permeability
- The permeability reduction is irreversible

The equation used for calculating the permeability reduction factor is

$$R_k = 1.0 + \frac{(R_{k,max} - 1.0)\, b_{rk} C_{41}}{1.0 + b_{rk}\, C_{41}} \qquad \textbf{(8)}$$

where

$$R_{k,max} = \min\left\{\left[1 - \frac{C_{rk}(A_{pl} C_{SEP}^{S_p})^{1/3}}{\left(\frac{\sqrt{k_x k_y}}{\varphi}\right)^{1/2}}\right]^{-4}, 10\right\} \qquad \textbf{(9)}$$

and b_{rk} and C_{rk} are the input parameters.

The permeability reduction is modeled in the subroutine PROP. The keyword IPERMRED in the input file will activate the permeability reduction physical model. The parameters b_{rk}, C_{rk} and A_{pl} are input parameters. The mobility of the aqueous phase containing the polymer is then reduced by the permeability reduction factor.

Polymer adsorption

The retention of polymer molecules in permeable media is caused by the adsorption onto solid surfaces and trapping within small pores and is modeled in GPAS as a function of rock permeability, salinity, and polymer concentration as follows:

$$\hat{C}_4 = \min\left(\tilde{C}_4, \frac{a_4(\tilde{C}_4 - \hat{C}_4)}{1 + b_4(\tilde{C}_4 - \hat{C}_4)}\right) \qquad \textbf{(10)}$$

where

$$a_4 = (a_{41} + a_{42} C_{SEP}) \quad \textbf{(11)}$$

and the effective salinity for polymer (C_{SEP}) is

$$C_{SEP} = \frac{C_{51} + (\beta_p - 1)C_{61}}{C_{11}} \quad \textbf{(12)}$$

Task 3: Validation and Application

A few test cases were run to validate the formulation and implementation of polymer component in GPAS. One-dimensional single-phase and two-phase polymer flood cases were simulated with GPAS and the results were compared with analytical solutions.

One-dimensional single phase polymer flood

A one-dimensional polymer flood test case was simulated with GPAS to test the polymer species implementation. The specific purpose of this simulation was to validate the polymer viscosity model by comparing the polymer injection rates with the waterflood injection rates. The simulation domain consists of 2000 ft in the x direction, 1 ft in the y direction and 1 ft in the z direction. An injector is located at the grid index (1,1,1) operating at a constant bottomhole pressure of 2000 psi and a producer in located at (50,1,1) operating at a constant bottomhole pressure of 1000 psi. The initial reservoir pressure is 1500 psi. The summary of the input file is given in Table 1. Both simulations of a waterflood wherein the tracer option was used and a polymer flood wherein the polymer option was used were run. The polymer is added to the injected water throughout the simulation time of 10,000 days. The polymer viscosity parameters used were: AP1 = 5, AP2 = 0, AP3 = 0. A plot comparing the results of water injection rates to polymer phase injection rates is shown in **Figure 1**. The aqueous phase injection rate

decreases to a value inversely proportional to the increase in the aqueous phase viscosity. The polymer concentration profile at 0.5 pore volumes is compared with the tracer concentration profile in **Figure 2** and indicates good agreement, which validates the implementation of the mass conservation equation for polymer.

One-Dimensional two-phase polymer flood

Another one-dimensional polymer flood test case was simulated with GPAS. The specific objective was to compare the water saturation profile in a one-dimensional polymer flood obtained from GPAS with the analytical solution (Pope, 1980). The simulation domain consists of 1000 ft in the x direction, 1 ft in the y direction and 1 ft in the z direction. A 100x1x1 grid arrangement is used. The reservoir has an initial water saturation of 0.35 and an initial oil saturation of 0.65. The residual water saturation is 0.25. An injector with a constant bottomhole pressure constraint of 5000 psi is located at (1,1,1) and a producer with a constant bottomhole pressure constraint of 1500 psi is located at (100,1,1). The initial reservoir pressure is 1500 psi. The oil phase consists of one component only, Decane. A summary of the input file is provided in **Table 2**. The polymer phase is injected continuously at the injector and oil is produced at the producer. The polymer viscosity is 3.12 cp. The mobile water phase creates first the first saturation shock front and the polymer phase creates the second saturation shock front. The fractional flow of water and polymer phases is given in **Figure 3**. The relative permeability curves are plotted in **Figure 4** with the equations given in **Table 3**. Finally, the water saturation profile obtained from GPAS at various pore volumes injected is compared with the water saturation profile obtained from the analytical solution in **Figure 5.** Although the trend is very similar but there is a significant difference between

the results. The problem is thought to be a bug in the water derivatives used in the Jacobian matrix and we are in the process of debugging the code.

Benchmark simulations

We conducted a series of benchmarks by running GPAS on an Intel Xeon-bases Linux cluster and studied the scalability while using different interconnects for the cluster.

Our testing environment is based on a cluster consisting of 64 Dell PowerEdge 2650 servers interconnected with Fast Ethernet, Gigabit Ethernet, and Myrinet. Each PowerEdge 2650 has two Intel Xeon processors running at 2.4 GHz with 512KB L2 cache, 2GB of DDR-RAM (double data rate RAM) memory operating on a 400 MHz Front Side Bus. The chipset of PowerEdge 2650 is the ServerWorks GC-LE, which accommodates up to six registered DDR 200 (PC1600) DIMMs with a 2-way interleaved memory architecture. Each of the two PCI-X controllers on the 2650 has its own dedicated 1.6 GB/s full duplex connection to the North Bridge to accommodate the peak traffic generated by the PCI-X busses it controls.

The operating system installed for the cluster is RedHat Linux 7.3 with kernel version 2.4.18-4smp. The GPAS is compiled with Portland Group C/C++ and FORTRAN 77/90 compilers, and the PETSc library.

Simulation model

In this example, the reservoir description is layered with permeability different in each direction, and permeability in z-direction is 10 times smaller than the permeability in x-direction. In addition, the size of the reservoir simulated is 7.3 miles in x direction, 24.25 miles in y direction and 500 feet in z direction. The reservoir domain is divided

into 197,120 blocks with 77x256x10 scheme. This leads to 1.5 million unknowns solved at each time-step simultaneously. There are 34 production wells and 54 injection wells totaling 88 wells scattered across the reservoir in a staggered line drive pattern. 100 days of gas injection is simulated. To run the case, the GPAS requires a total of 1.7GB memory.

Performance metric

The efficiency in parallel applications is usually measured by speedup. In this study, the term "*speedup*" of a cluster with *N* processors is defined as

$$\text{Speedup} = \frac{t_1}{t_N}$$

where t_1 is the amount of execution time by running on one processor and t_N is the amount of execution time spent with N processors. The ideal speedup of parallel simulation with N processors is N, that is, the program runs N times faster. However, in reality, as the number of processors becomes larger, we usually observe a speedup less than N. The performance reduction is due to increasing inter-processor communication or the memory contention arising from a cluster whose nodes are Symmetric Multi-Processors (SMP) machines. Sometimes, it can be due to an unfavorable programming style, in which the inefficient program does not decompose the application evenly. These are known as the overheads that are not encountered if there is only one processor.

Results and analysis

We have configured three different interconnects, Fast Ethernet, Gigabit Ethernet, and Myrinet for our study. For this part of the study, only one processor per compute node is used. **Figure 6** shows the execution times (bars) and speedups (lines) of the

simulation case from one processor to 64 processors. The left y axis in the figure is the execution time in terms of seconds and the right y axis is the performance speedup of the GPAS. The GPAS has the best performance as well as the scalability (speedup) on the cluster with Myrinet. In fact, the speedup at 16 processors is almost 20, a super linear speedup. The performance difference becomes significant only after 16 processors.

The simulation was constituted of several sections such as I/O, Initialization, Linear Solver, Viscosity and Relative Permeability, Residuals, Ghost Region, Jacobian, and Dependent Variables. These sections are timed individually during the simulation and reported at the end. To identify which section of the GPAS is affected by which part of the cluster configuration, we use the time spent on each section at a single processor as the base to generate a speedup curve for each section of the code. **Figure 7** shows the speedup of each GPAS's section for the large case (77x256x10) on the Myrinet cluster. The plot shows that each section of the code has its own speedup rate from one processor to 32 processors. The speedups of "Compute Residuals" and the "Update Viscosity and Relperm" have a slope 2.5 that indicates these sections benefit considerably from the aggregation of the memory bandwidth and cache sizes. The "Update Dependent Variables" and the "Update Jacobian" sections have the speedup rate around 1.0. That indicates the aggregated memory and cache effects on these sections are offset by the communication overheads of having more processors. Finally, the speedup rates for the two sections, "Total Linear Solver Time" and the "Total Initialization Time" are less than 1.0. That is because these sections are communication sensitive and the performances are affected more by the interconnect capacity than others.

Furthermore, we calculate the ratio of the time spent in each section for Fast Ethernet to that of Myrinet, and plot the results in **Figure 8**. The ratios clearly show that "Total Linear Solver Time" and the "Total Initialization Time" are the most communication dependent sections and they are the only two sections show relatively large difference from Fast Ethernet configuration to Myrinet configuration. On the other hand, all other sections have ratio of 1, which indicates the communication that required by those sections is minimal. Therefore, the differences between the two interconnect performance is not shown in those sections.

Single-processor vs. dual-processor

We continued our study in comparing the performances of GPAS for single-processor and dual-processor per node configurations. Two processes running in a system will compete for the memory resource. Particularly, the shared memory bus will be the performance bottleneck, when the memory is accessed at the same time by the two processes. In addition, the communication traffics generated by the two processes could create another potential bottleneck on the I/O resource, such as the PCI bus or the network interface card. Based on the general knowledge, we calculate the ratio of the time spent in each section for single-processor to that of dual-processor runs performed on the Myrinet cluster, and plot them in **Figure 9**. Almost all the ratios are decreasing as the number of processor increases. That shows the memory contention problem becomes less as more processors are used. The reason is the data set per processor getting smaller, which reduces the memory contention relatively. On the other hand, with the fact that as the processor count increases, the communication among processes also increases, for the

communication sensitive sections such as the “Total Linear Solver Time”, the ratio is not just considerably large, but also increases with the processor count.

CONCLUSIONS

In this report, we gave our progress on Tasks 1 through 3 for the first six months of the second year of the project. We have formulated the mass conservation equation and physical properties for polymer. We implemented and validated the polymer module. We have also conducted a series of benchmarks by running the General Purpose Adaptive (GPAS) simulator on a Linux cluster and studied the scalability while using different interconnects. The results were very encouraging and indicated that GPAS performance scales linearly from one to 64 single processor nodes using a low latency, high-bandwidth such as Myrinet.

REFERENCES

Camilleri, D., Engelsen, S., Lake, L.W., Lin, E.C., Ohno, T., Pope, G.A. and Sepehrnoori, K.:" Description of an Improved Compositional Micellar/Polymer Simulator," *SPE Reservoir Engineering*, November 1987.

Flory, P.J.: *Principles of Polymer Chemistry,* Ithaca, New York, Cornell University Press, 1953.

Lake. L.W : *Enhanced Oil Recovery*, Prentice Hall , New Jersey, 1989.

Table 1: Input data for the single-phase one-dimensional polymer flood case.

Dimensions (ft):	Length	2000
	Width	1
	Thickness	1
Number of gridblocks in X direction	NX	50
Number of gridblocks in Y direction	NY	1
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Absolute Permeability (md)		50
Initial Water Saturation (fraction)		0.99
Reservoir Temperature (^{o}F)		130
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H_2O	1.0
Water viscosity (cp)		0.79
Water Compressibility (1/psi)		6.7×10^{-5}
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		10,000
Polymer Injection Time (days)		10,000
Constant Time Step Size (day)		0.1
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Injection Polymer Concentration (mass fraction)		1.0
Injection Fluid Concentration (mole fraction)	H_2O	1.0
Producer Well Location (I,J,K)		(50,1,1)
Production Bottom Hole Pressure (psi)		1000
Polymer Viscosity Parameter	AP1	5.0
	AP2, AP3	0.0

Component Critical Properties:

Component	T_c (^{0}R)	P_c (PSI)	V_c (ft^3/lb-mole)	ω_i (dyne$^{1/4}$ cm$^{11/4}$/gm-mole)	MW_i (lbm/lb-mole)	ψ_i
H_2O	1165.14	3207.4	1.0215	0.34400	18.02	100.0

Table 2: Input data for the one-dimensional two-phase polymer flood case.

Dimensions (ft):	Length	1000
	Width	1
	Thickness	1
Number of gridblocks in X direction	NX	100
Number of gridblocks in Y direction	NY	1
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Permeability (md)		500
Initial Water Saturation		0.35
Residual Water Saturation		0.25
Reservoir Temperature (^{o}F)		130
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	$C_{10}H_{22}$	1.0
Water viscosity (cp)		0.52
Water Compressibility (1/psi)		3.0e-5
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		5,000
Polymer Injection Time (days)		5,000
Constant Time Step Size (days)		0.0001
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		5000
Injection Polymer Concentration		1.0
Injected Fluid Concentration (mole fraction)	H_2O	1.0
Producer Well Location (I,J,K)		(100,1,1)
Production Bottom Hole Pressure (psi)		1500
Polymer Viscosity Parameter	AP1	5.0
	AP2, AP3	0.0

Component Critical Properties:

Component	T_c (^{0}R)	P_c (PSI)	V_c (ft^3/lb-mole)	ω_i (dyne$^{1/4}$ cm$^{11/4}$/gm-mole)	MW_i (lbm/lb-mole)	ψ_i
$C_{10}H_{22}$	1111.8	304.0	12.087	0.488	142.3	431.0

Table 3: Relative permeability parameters used in two-phase polymerflood case

$K_{rw}^{0} = 0.2$ $m = 3$ $S_{wr} = 0.25$	$K_{ro}^{o} = 1.0$ $n = 4$ $S_{or} = 0.3$
$S = \frac{S_w - S_{wr}}{1 - S_{or} - S_{wr}}$ $K_{r1} = K_{r1}^{o}(S)^{m}$	$S = \frac{S_w - S_{wr}}{1 - S_{or} - S_{wr}}$ $K_{r2} = K_{r2}^{o}(1 - S)^{n}$

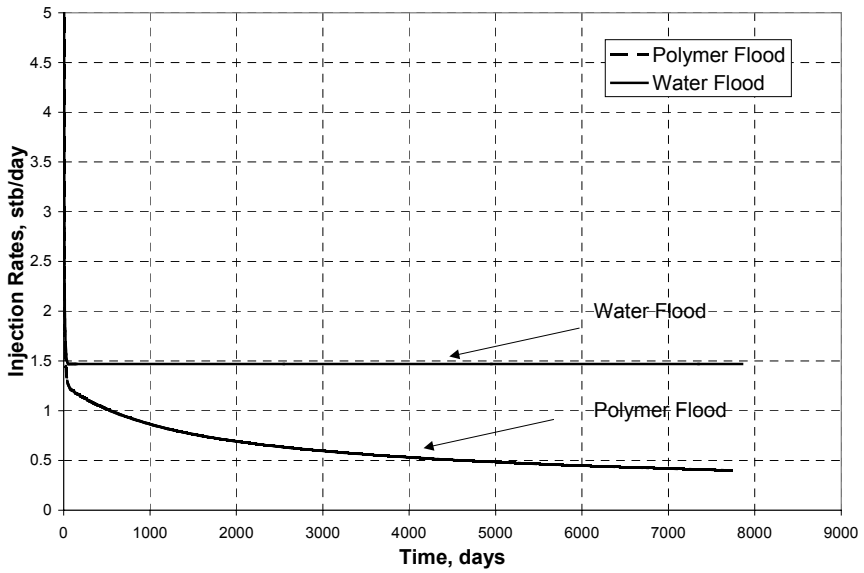

Figure 1: Comparison of the water phase with the polymer phase injection rates for the one-dimensional homogenous polymer flood test case.

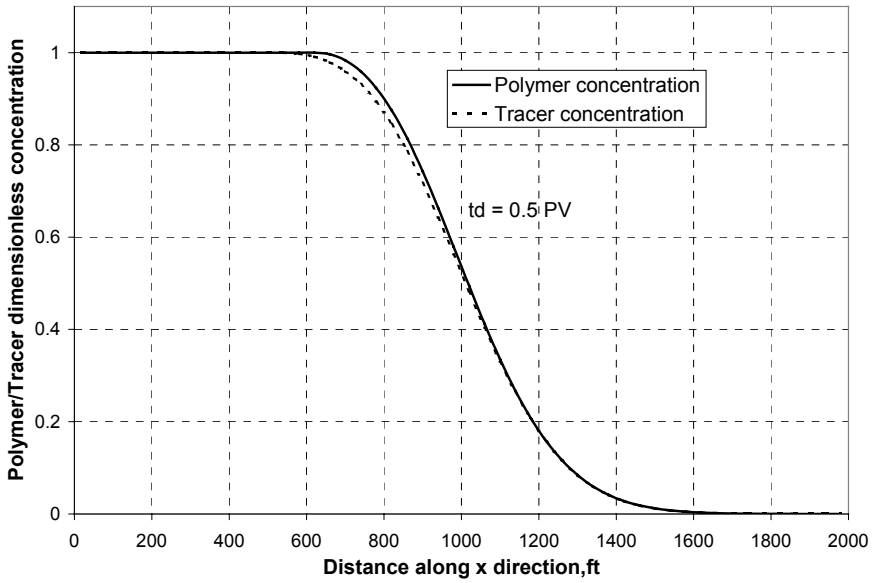

Figure 2: Comparison of the polymer and tracer concentration profiles for the one-dimensional polymer and waterflood test cases at 0.5 PV.

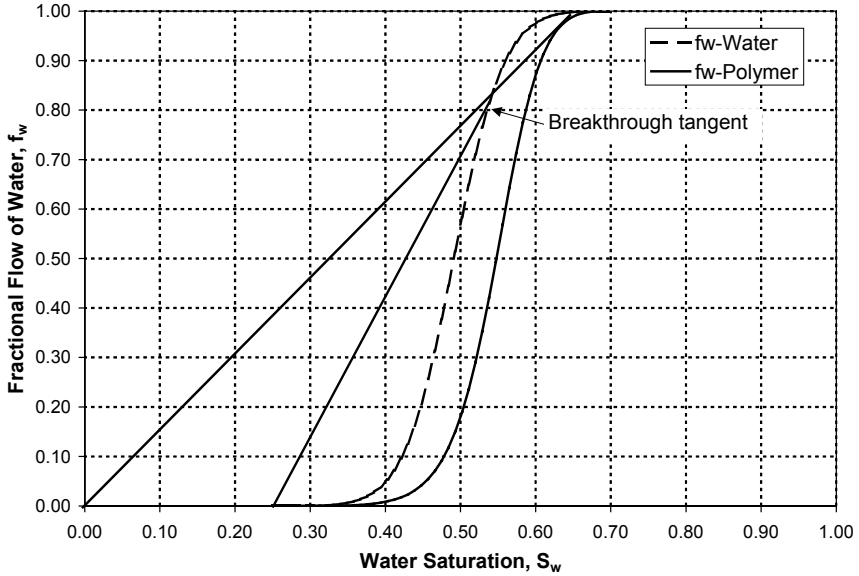

Figure 3: Fractional flow of water and polymer phases for the one-dimensional polymer flood.

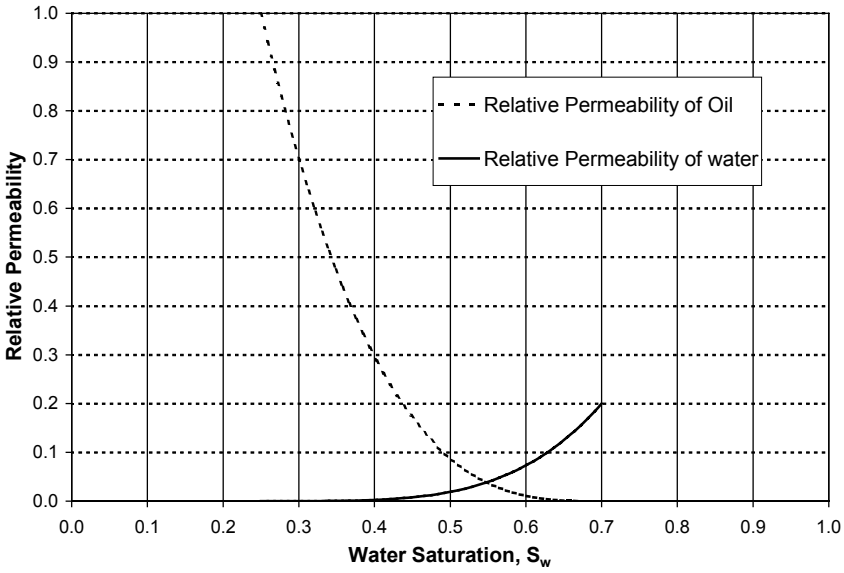

Figure 4: Relative Permeability curves used in the one-dimensional polymer flood test case.

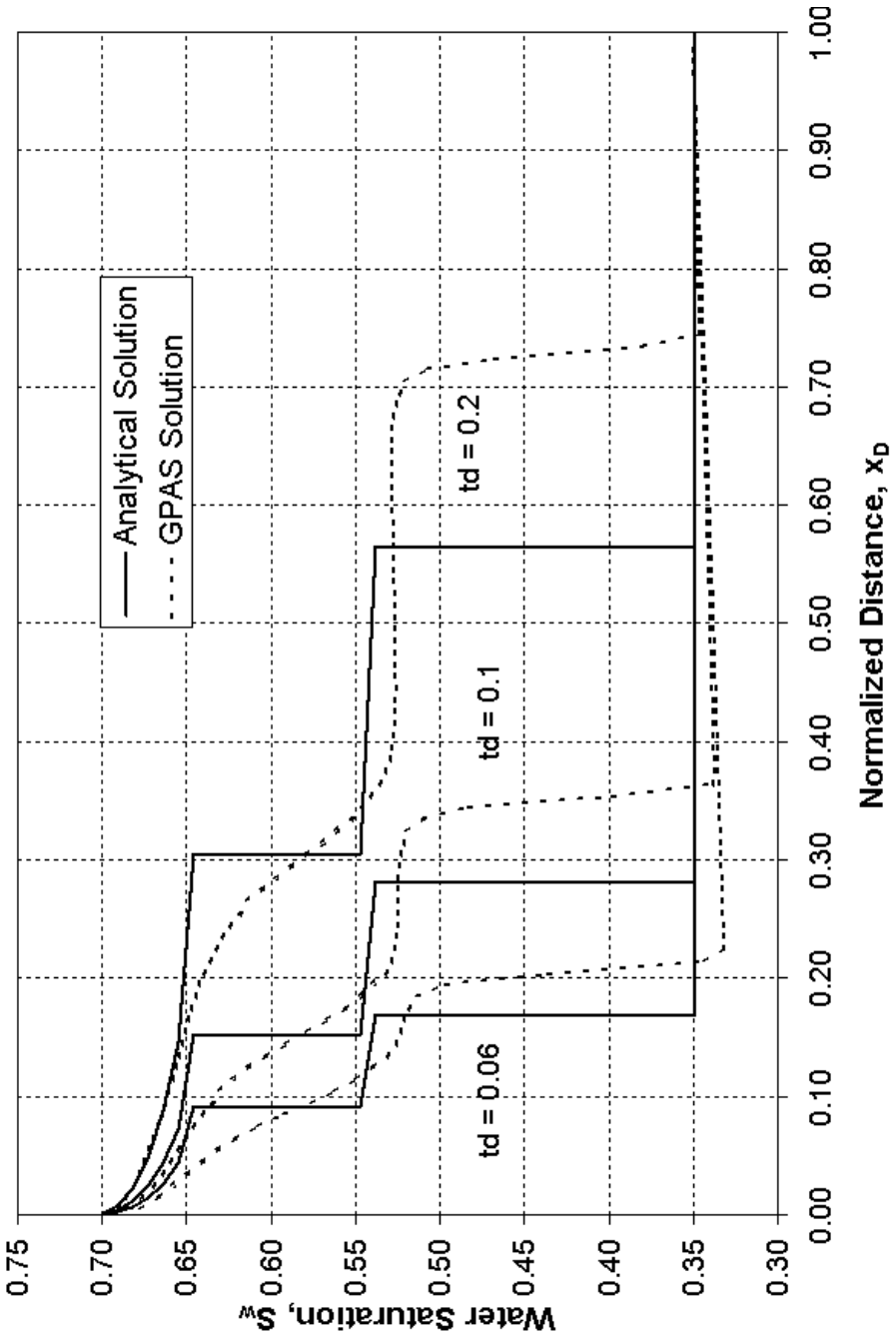

Figure 5: Comparison of the water saturation profile from GPAS with the analytical solution at 0.06, 0.1 and 0.2 pore volumes injected for the one-dimensional polymer flood case.

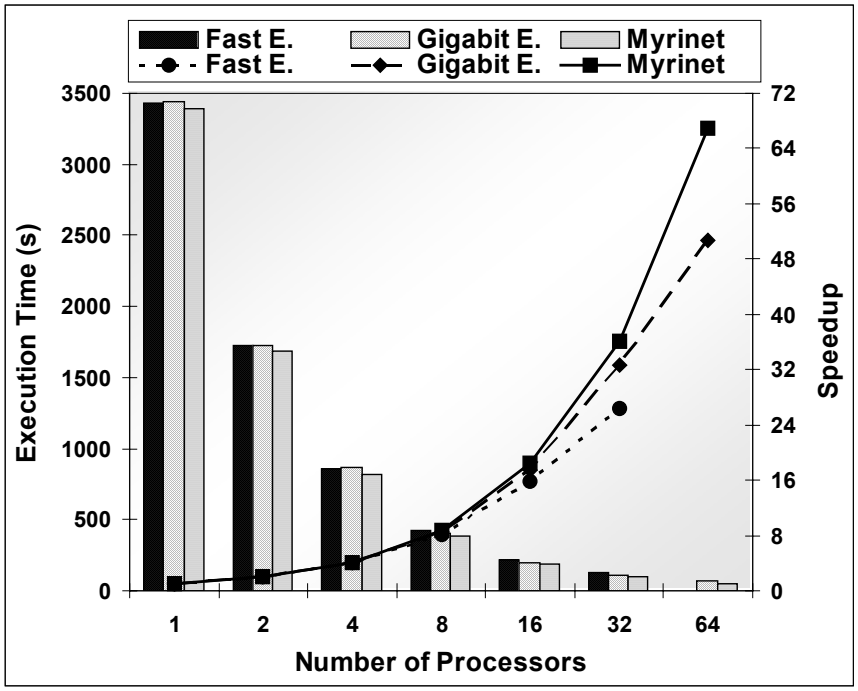

Figure 6: GPAS execution time and speedup plots for the case of 77x256x10 using single processor per node.

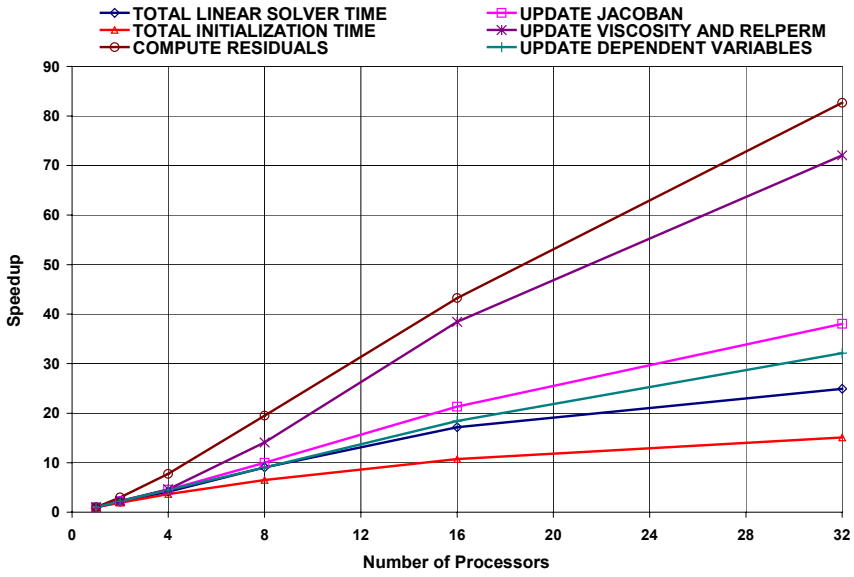

Figure 7: Speedup curves of each section for the simulation case (77x256x10) using Myrinet as the interconnect.

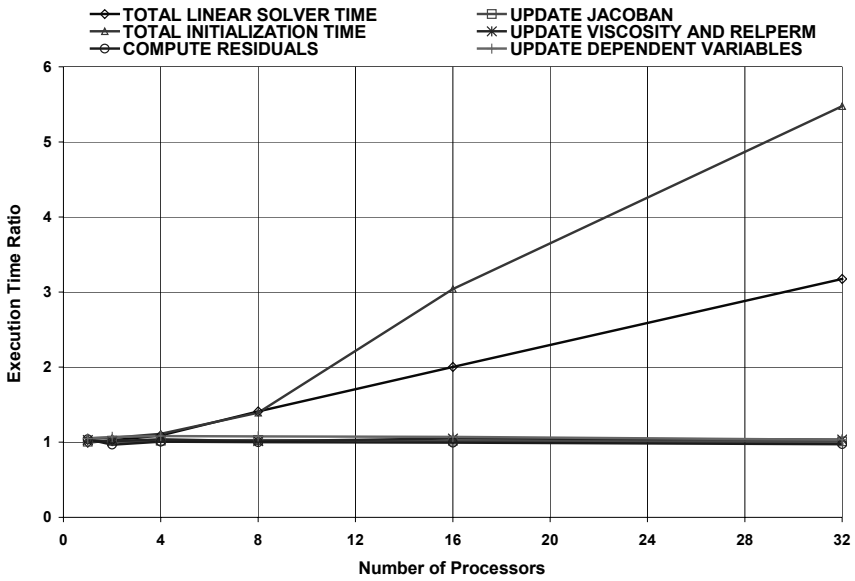

Figure 8: Execution time ratio of Fast Ethernet to Myrinet for each sections of the larger simulation case (77x256x10).

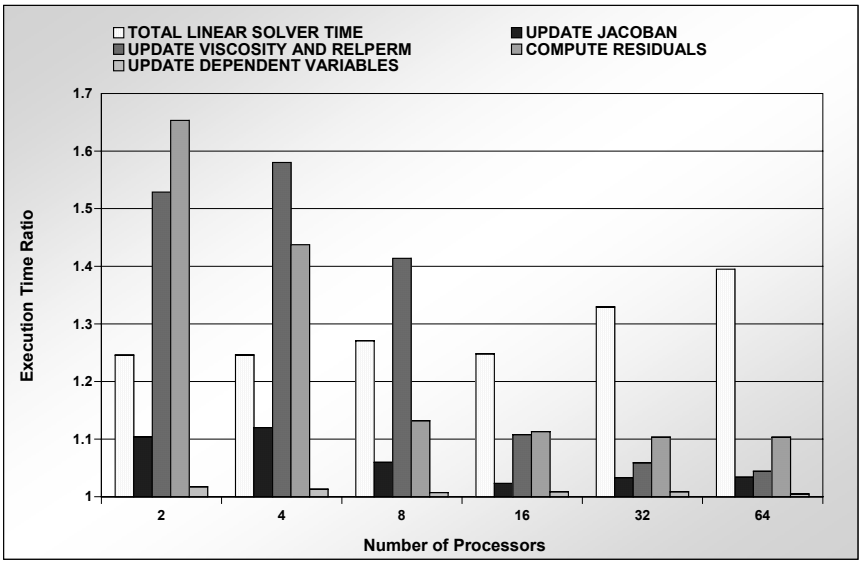

Figure 9: Execution time ratio of dual-processor to single-processor for each sections of the simulation case (77x256x10) on the cluster with Myrinet.