TRACER FLOW MODELING AND EFFICIENT SOLVERS FOR GPRS

A REPORT SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

By
Yuanlin Jiang
June 2004

I certify that I have read this report and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Petroleum Engineering.

_____

Dr. Khalid Aziz
(Principal Advisor)

I certify that I have read this report and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Petroleum Engineering.

_____

Dr. Hamdi Tchelepi
(Co-principal Advisor)

Abstract

Reservoir simulators solve equations of fluid flow in porous media on computers to mimic the behavior of petroleum reservoirs. They form one of the most important branches of petroleum reservoir engineering. Aziz and Setatri (1979) summarized the work in the field in their book *Petroleum Reservoir Simulation*. With the dramatic increase in computational speed, and the emergence of new solution techniques, reservoir simulators have turned out to be one of the most powerful tools to manage petroleum reservoirs.

Now, in industry, there are many mature commercial and in-house reservoir simulators, such as Eclipse of Schlumberger, Chears of Chevron-Texaco, Imax of Computer Modeling Group, etc. However, the source codes of these commercial simulators are not available to other researchers, which makes it difficult to test new ideas timely. Cao (2002) developed General Purpose Research Simulator (GPRS) as part of his Ph.D. research. His objective was to produce a simulator that it could be used later by a large research group to preserve and enhance the work of several investigators.

Cao (2002) tested GPRS with several problems. Recently a test project was carried out at Stanford University by Pan (2003) to further evaluate the performance of GPRS. All of these tests show that GPRS is a well-designed, robust simulator. However, GPRS is still young and many necessary options are absent at present. Furthermore, the overall performance is not as good as that of other more mature simulators. In this report, we introduce the implementation of a high order technique known as the Total Variance Diminishing (TVD) method for tracer flow simulation. In addition, we investigate the properties of pressure matrix and three kinds of preconditioners and attempt to improve the efficiency of the simulation by improving the performance of the linear solvers used in the simulator, which is the primary topic of this report. As a result of this investigation, huge improvements in the solver performance have been achieved.

Acknowledgments

I would like to express my sincere gratitude to my advisors Dr. Khalid Aziz and Dr. Hamdi Tchelepi for their encouragement, support and guidance throughout this work. I deeply appreciate their reading of my drafts and their valuable input to this report. I thank both of them for providing me motivation to complete this task.

My thanks also go to Dr. Hui Cao, the original author of GPRS, with whom I had many helpful discussions related to this project. I am grateful to my old friend Hui Zhou for valuable discussions as well. Thanks are also due to Dr. Huanquan Pan, who as the administrator of GPRS, is always ready to help.

Financial support from the Stanford University Petroleum Research Institute programs on Reservoir Simulation & Advanced Well (SUPRI-B & SUPRI-HW) is gratefully acknowledged. This research was also supported by US Department of Energy (contract number DE-AC26-99BC15213).

I also want to give my thanks to my colleagues and friends for their help within and outside my academic life, which started from the first day I came to Stanford.

Special thanks are due to my wife, Weimin Sun, my parents and my sister for their support.

Contents

List of Tables

List of Figures

Chapter 1

1. High Accuracy Simulation of Tracer in Multiphase Flow

A General Purpose Reservoir Simulator (GPRS) developed by Cao (2002) is widely used in Stanford University's SUPRI-B and SUPRI-HW research groups as a general platform for new developments. It is also used by several other research groups and companies for their in-house research. The reservoir simulation group at Stanford is engaged in enhancement and further development of GPRS. One such development is the implementation of a water-carrying tracer model in GPRS.

1.1. Introduction

The ability to model tracer flow is a very useful option in a simulator, which can monitor the movement of water phase. In industry, it is used to identify reservoir characteristics between pairs of injectors and producers. It is also a valuable tool to evaluate the contribution of a specific injector within a group of injection wells (Ali et al. 2000). It is possible with the use of tracers to see how the injected water moves in the reservoir and to identify its contribution to the total produced water.

Since tracer transportation is convection-dominated, a standard 1^{st} order scheme may cause significant numerical dispersion and smear the sharp front of tracer concentration. Hence, higher accuracy methods are needed for tracer simulation.

As an important tool in reservoir engineering, tracer simulation has been widely used and studied for a long time. The general requirements for tracer simulation are low-cost, accuracy and stability. The focus of this research is on improving the accuracy of tracer modeling in a multiphase system.

Palagi (1992) developed a dual timesteps technique to simulate tracer flow. Since an explicit scheme is cheaper in terms of computational cost than an implicit scheme, provided it is stable, Palagi used an explicit scheme. Furthermore, explicit schemes are generally more accurate than implicit schemes. The stability constraint is the only reason why these schemes are not widely used in reservoir simulation. In the dual timestepping approach the maximum stable step for the tracer is calculated at the end of the multiphase flow calculation, which might be less than the timestep in multiphase simulation. This maximum stable step is used to finish the tracer flow simulation.

High order schemes have been in existence for a long time. However, the non-physical oscillation observed with these techniques is a big issue for schemes which have 2^{nd} order or higher accuracy. Godnov (1959) pointed out that no scheme with constant coefficients could achieve 2^{nd} or higher accuracy without oscillations.

Harten (1983) published "high resolution schemes for hyperbolic conservation laws" and introduced the total variance diminishing (TVD) concept for the first time. A limiting function, so called limiter, was introduced to directly modify the calculation of flux across a block boundary. The application of limiter techniques changes the character of traditional constant coefficient schemes.

Rubin and Blunt (1991) were the first to use higher accuracy schemes in reservoir simulation. They simulated multiphase flow in a black oil model and got good results. With a 30 x 30 grid and a TVD scheme, they achieved better results than with an 80 x 80 grid and a conventional upstream scheme. The increase in computational cost for the TVD scheme is very small. So the overall computational cost of TVD schemes is comparable to that of the conventional upstream schemes with the same grid.

Wattenbarger et al. (1997) used TVD schemes to simulate the flow of tracers in single water phase flow. They proposed a High-Throughput (HT) timestepping technique, and combined it with TVD schemes.

1.2. Methodology of the Tracer Flows Simulation

In order to minimize the cost of computer time, explicit methods are employed to simulate tracer flow. Due to the stability issues of these explicit methods, dual timestepping technique (Palagi, 1992) is used to guarantee stability.

1.2.1. Standard Upstream Approximation

Tracer flow is convection dominated. In this study, we assume that the carrier phase is water and the tracer exists only in water. Because of its low concentration, it does not affect the properties of water (Lake, 1989, Chapter 2 and 5).

The governing equation for a tracer is the conventional advection equation with a source term (Lake, 1989):

$$\frac{\partial (m\, c)}{\partial t} + \nabla \cdot (f\, c) = -Q\, c \tag{1-1}$$

In Eq. 1-1, m is the moles of the carrier phase per unit volume, f stands for the flux, c is the concentration of the tracer and Q is the source term. The concentration of tracer is normalized by the injection concentration, which is its largest value. So c is a dimensionless variable in the range 0 to 1.

Eq. 1-2 is the implicit discrete approximation for Eq. 1-1 (GeoQuest Schlumberger, 2001A).

$$R_i = M_i^{n+1} c_i^{n+1} - M_i^{n} c_i^{n} + \Delta T \sum_n F_{i\to l}^{n+1} c_{in}^{n+1} + \Delta T\, c^{n+1} Q_{M_i} = 0 \tag{1-2}$$

M_i is the moles of carrier phase in cell i, c_i is the tracer concentration in cell i, $F_{i\to l}$ is the molar flow rate of the carrier phase from cell i to neighboring cell l, Q_{M_i} is the molar production rate of the carrier phase from well completion in block i, c_{in} is the upstreamed tracer concentration for the boundary between cells i and j, and the superscripts $n, n+1$ stand for the time index.

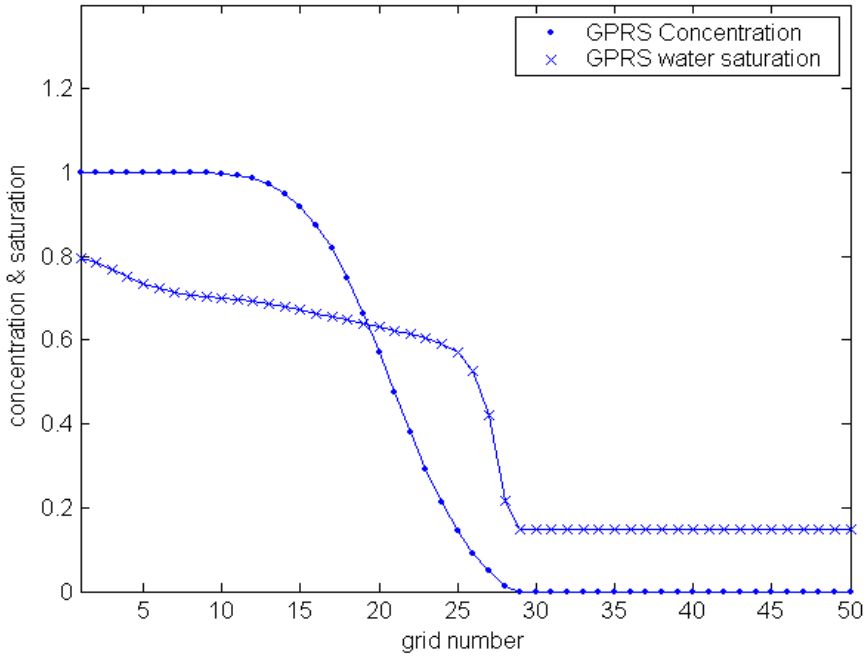

Fig. 1-1: Typical profiles of water saturation and tracer concentration using upstream approximation

Fig. 1-1 shows a typical case of the tracer flow. This is a 1D, oil water system, with a connate water saturation of 0.15. Water with tracer is injected in the first cell at the left, and fluid is produced from the right side. Since the saturation of the carrier phase plays a key role in the movement of tracer, the water saturation curve is also shown. In Fig. 1-1, the tracer profile shows significant dispersion due to the upstream method employed. Since we did not include any physical dispersion, the tracer exclusively exists in the area that has been flooded by water (Fig. 1-1). The results presented above clearly show the need for a scheme that reduces numerical dispersion.

1.2.2. Flux Limiting Scheme

There are many kinds of techniques such as the Lax-Wendroff scheme, Leap Frog scheme and MacCormack scheme as discussed by Zhang and Shen (2003), which have 2nd or higher order accuracy. However, all these methods introduce nonphysical oscillations (Godunov, 1959).

Consider the non-linear Eular equation:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{1-3}$$

where u is a conserved quantity, $f(u)$ is the flux of u. The solution of this equation is monotonic. For u specified on an arbitrary grid, the total variance of u or $TV(u)$ is defined as follows:

$$TV(u) = \sum_{j=1}^{N-1} \left| u^{n}_{j+1} - u^{n}_{j} \right| \tag{1-4}$$

where j is the grid block index, n is the time index, and N is the number of gridblocks. If we want to make $TV(u)$ diminish or not allow it to grow with time, then

$$TV(u^{n+1}) \le TV(u^{n}) \tag{1-5}$$

If the above condition is satisfied, we will generate no extreme points. This means that the solution will be monotonic without non-physical oscillations.

Flux-limiting schemes are one kind of monotonic schemes. As mentioned before, we can not achieve this property by any high order scheme with constant coefficients. Therefore, a limiting function, so called limiter, is introduced to directly modify the flux. Obviously, the limiter will be a function of the gradient of u, and it will adjust the scheme to be upstream, mid-point and downstream in different parts of the domain.

TVD schemes have 2nd order accuracy and they regress to 1st order at extreme points. This can be strictly proved only for one dimensional scalar, linear conservation equations (Harten, 1983). The scheme can be extended to nonlinear equations and two and three dimensional problems. These extensions will be discussed in Section 1.3.

1.2.3. One Dimensional TVD Scheme for Tracers

Rubin and Blunt (1991) applied the TVD scheme to incompressible, multiphase flow. They solved

$$\frac{\partial m_w}{\partial t} + \frac{\partial f_w}{\partial x} = 0 \tag{1-6}$$

with

$$m_w = \phi \rho_w S_w \tag{1-7}$$

$$f_w = -\frac{Kk_{rw}(S_w)\rho_w}{\mu_w}\frac{\partial P_w}{\partial x} \tag{1-8}$$

In Eq. 1-6, 1-7 and 1-8, m_w is the mass of water per unit volume, f_w is the water flux, ϕ is the effective porosity, ρ_w is the density of water, S_w is the water saturation, μ_w is the viscosity of water, K is the permeability of porous media, k_{rw} is the relative permeability of the water phase and P_w is the pressure of the water phase.

The above equations can be written in discretized form as follows. Here, we assume that water flows from left to right, i.e., cell i is the upstream direction of cell $i+1$:

$$f_{w,j+1/2}^{n+1} = f_{w,j}^{n+1} + \frac{1}{2}\varphi(r_j^{n+1})(f_{w,j+1}^{n+1} - f_{w,j}^{n+1}) \tag{1-9}$$

with

$$f_{w,j} = -\frac{K k_{rwj} \rho_{wj}}{\mu_{wj} \Delta x}(P_{w,j+1} - P_{w,j}) \tag{1-10}$$

$$r_j = \frac{f_{w,j} - f_{w,j-1}}{f_{w,j+1} - f_{w,j}} \tag{1-11}$$

where r is the ratio of successive terms. Here, $\varphi(r)$ is the limiter, which is a function of r. If $\varphi(r)$ is equal to 0, the scheme becomes upstream; if $\varphi(r)$ is equal to 1, the scheme becomes midpoint; and if $\varphi(r)$ is equal to 2, the scheme becomes downstream. We have several choices for the limiter function, which will be discussed in details later.

Based on above equations, we write the governing equations of tracer simulation, and derive the discrete TVD schemes. These equations are:

$$\frac{\partial m_c}{\partial t} + \frac{\partial f_c}{\partial x} = 0 \tag{1-12}$$

with

$$m_c = \phi \rho_w S_w c \tag{1-13}$$

$$f_c = -\frac{Kk_{rw}(S_w)\rho_w}{\mu_w}\frac{\partial P_w}{\partial x}c \tag{1-14}$$

In discretized form, the equation is:

$$f_{c,j+1/2}^{n+1} = f_{w,j}^{n+1} c_j^{n+1} + \frac{1}{2}\varphi(r_j^{n+1})(f_{w,j+1}^{n+1} c_{j+1}^{n+1} - f_{w,j}^{n+1} c_j^{n+1}) \tag{1-15}$$

with

$$r_j = \frac{c_j - c_{j\text{-}1}}{c_{j+1} - c_j} \tag{1-16}$$

If the connate water saturation is zero for all cells, the tracer equation will regress to the water phase flow equation (Rubin and Blunt, 1991). If the water saturation is 1.0 for each cell, i.e., the whole reservoir is filled by water, the equation regresses to the single phase tracer flow equations studied by Wattenbarger et al. (1997).

In order to satisfy the TVD property requirements, the limiter function $\varphi(r)$ must be in the shaded region of Fig. 1-2 (Fu, 1993).

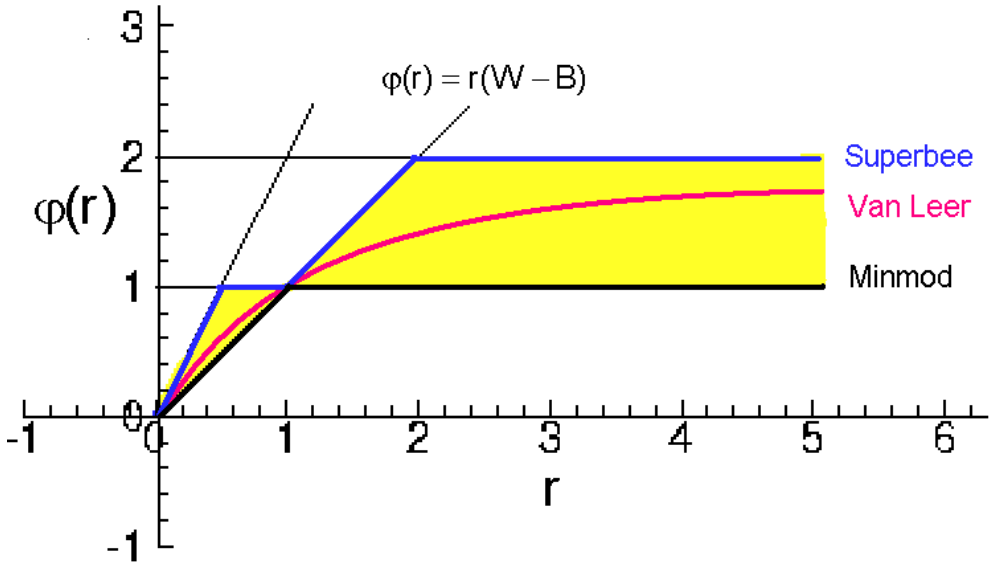

Fig. 1-2: Region of 2nd order accuracy (shaded) for three limiter functions (after Lien and Leschziner, 1994)

As we can see from this figure, there are many choices for the limiter. The most common limiters are Minmod, Van Leer and Superbee (Liu and Shu, 2003). Their curves are also shown in Fig. 1-2, and their expressions are given below:

Minmod limiter function:

$$\phi = \max\{\, 0, \min(\, r, 1)\,\} \tag{1-17}$$

Van Leer limiter function:

$$\phi = \frac{|r| + r}{|r| + 1} \tag{1-18}$$

Superbee limiter function:

$$\phi = \max\{\, 0, \min(\, 2r, 1), \min(\, r, 2)\} \tag{1-19}$$

These limiters introduce an anti-diffusion character to the finite-difference schemes, and different limiters give different amounts of anti-diffusion. We will compare results from these three limiter functions in Section 1.4.

1.3. Implementation of the Tracer Option

In Section 1.2, we have presented the methodology of the tracer flow simulation as implemented in GPRS. Since we assume that the tracer does not affect the properties of the carrier fluid, the tracer concentration term does not appear in the multiphase flow equations. Therefore, the problem of tracer flow is naturally decoupled from the multiphase flow problem. We can solve the multiphase flow equations first, and then update the concentration in an explicit step using the dual timestepping technique. This also makes the tracer option a relatively isolated module in the GPRS code, which is very good from the viewpoint of developing and maintaining the software.

The implementation of the tracer option includes two parts. The first part requires of us to add the tracer into the logical framework of GPRS. We build up the relations between tracer and other objects, such as reservoir, wells, well controls, and then embed the tracer into the running sequence of the GPRS. This part is mostly pure design and coding work, and the details are provided in Appendix A. In the second part, the algorithm of tracer flow simulation was implemented into *class Tracer*, which includes both the standard upstream scheme and the TVD schemes. To illustrate the design, a flowchart of *class Tracer* is provided in Appendix A, details of the tracer option as implemented in GPRS are offered in Appendix B, and sample input and output files with elaborate explanations are furnished in Appendix C.

1.4. Results and Extensions:

In this section, we will compare the three limiters discussed in Section 1.2 and validate our implementation of the tracer option. The TVD schemes are extended to 2D and 3D modals, and more complex cases are tested for validation.

1.4.1. Comparison of Different Limiter Functions

The problem concerned here is one-dimension two-phase oil-water system with a connate water saturation of 0.15. This system is homogeneous with 0.25 porosity and 500md permeability. The number of cells is 50. One injection well is located in the middle of the

first grid block, and one production well is in the last grid block. The water with tracer is injected from the injection well.

As mentioned before, the saturation of the carrier phase has an important effect on the tracer concentration, so the water saturation curve is always plotted with the tracer concentration curve. Our results are also compared with the tracer results obtained from the Eclipse simulator (GeoQuest Schlumberger, 2001). Fig. 1-3, 1-4 and 1-5 show results for this problem with the three limiters discussed previously when PVI is equal to 0.28.

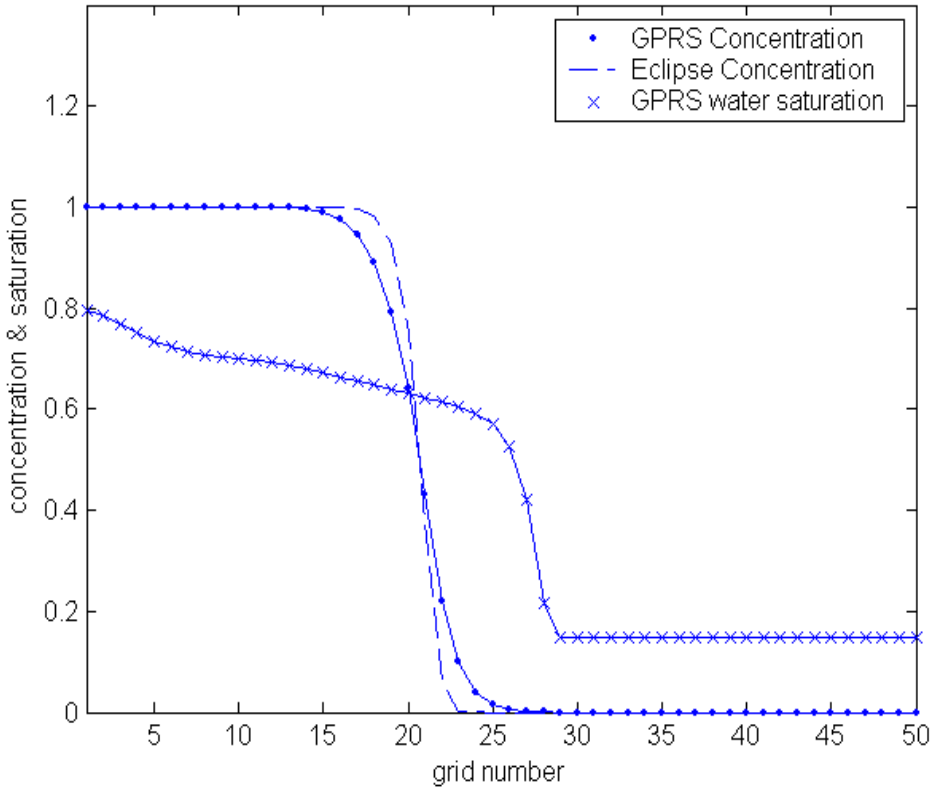

Fig. 1-3: Result of Eclipse and result of GPRS with Minmod limiter

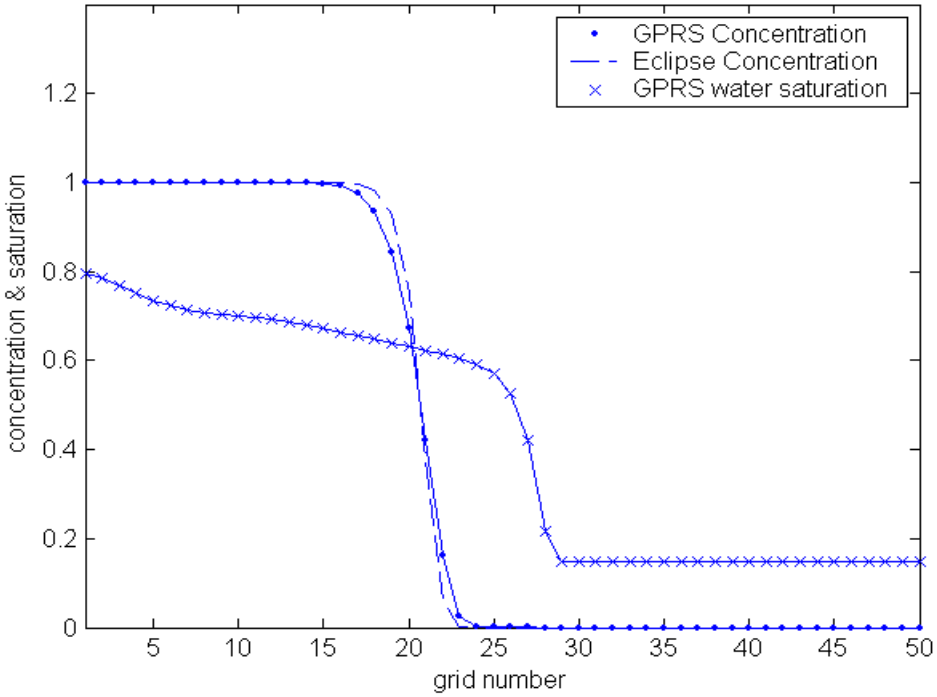

Fig. 1-4: Result of Eclipse and result of GPRS with Van Leer limiter

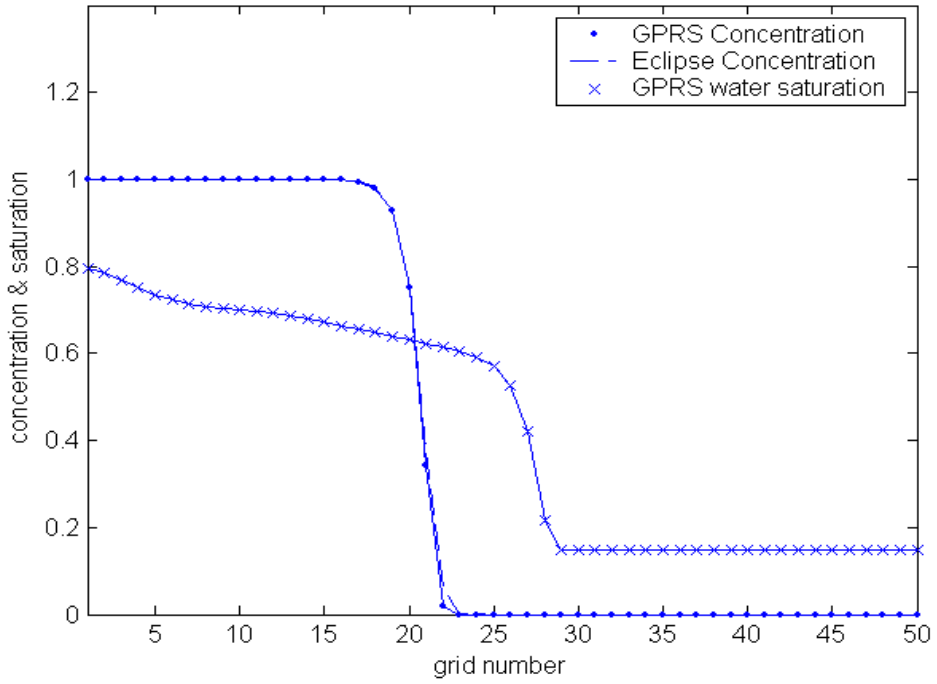

Fig. 1-5: Result of Eclipse and result of GPRS with Superbee limter

We can see that the Superbee limiter (Fig. 1-5) has the best anti-diffusion properties and produces results very close to those from the Eclipse simulator. The anti-diffusion properties of Minmod limiter are not as good as those of the other two, but the TVD scheme with Minmod limiter still performs much better than the upstream scheme (Fig. 1-1). The anti-diffusion characteristics of a particular limiter can be deduced from Fig. 1-2. The higher the limiter curve in the shaded region of Fig. 1-2, the more accurate are the results.

To further test the tracer implementation in GPRS, a slug injection case is considered. The system setting is the same as that in the previous case, except that the tracer injection is stopped after 1000 days to form a concentration slug (PVI is equal to 0.056). Note that water was injected continuously. Because the slug is very sensitive to numerical diffusion, this is a good case to illuminate the difference in the anti-diffusion properties of different limiters. Another goal of this test case is to make sure that the scheme is valid when the extreme point is in the middle of the domain rather than on the boundary. Since the basic implementation has been verified in the previous case, we do not provide the results of the Eclipse simulator in this case. Fig. 1-6 and 1-7 show the results with the application of Minmod and Superbee limiters after 5000 days injection (PVI is equal to 0.28). Again, Superbee limiter produces the best results.

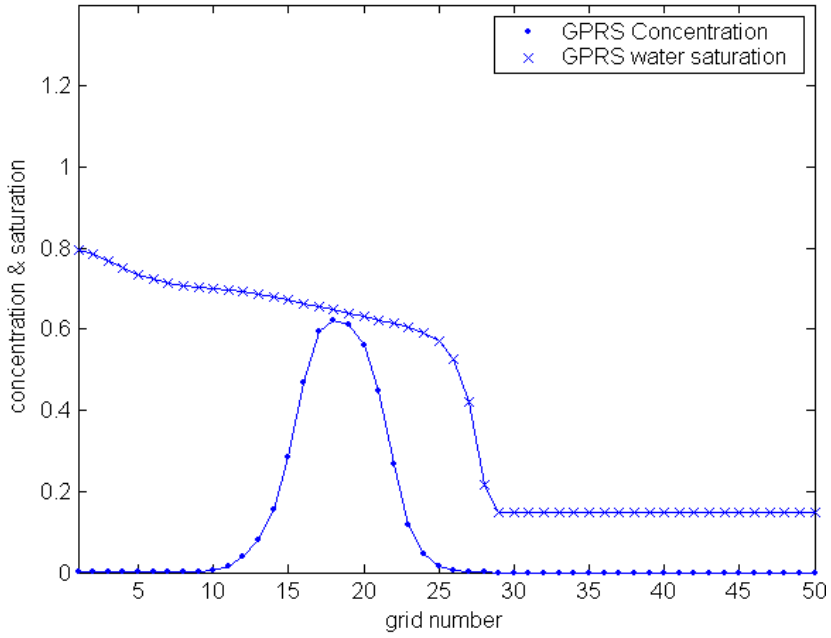

Fig. 1-6: Slug solution with Minmod limiter in GPRS

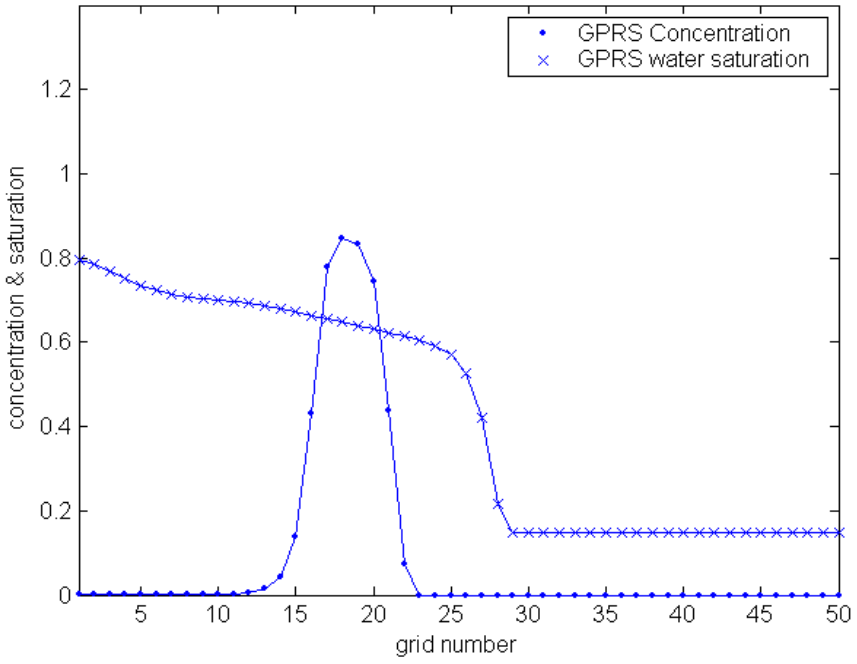

Fig. 1-7: Slug solution with Superbee limiter in GPRS

Since the Superbee limiter has the best performance, we use it in the further testing of the tracer option in GPRS. The TVD model with Superbee limiter is extended to 2D and 3D. More tests are performed and results are compared with the Eclipse simulator in the following section.

1.4.2. Extension of TVD to 2D and 3D and More Validations

TVD scheme is extended to 2D and 3D by applying the 1D scheme in each direction. It is worth mentioning two points here. One, there is no theoretical proof of the TVD property of models other than in 1D. But numerous numerical works show that TVD schemes are valid in 2D and 3D model (Fu, 1993); the other, we did not account for the mixed derivative term. Therefore, the results of the scheme may not be strictly of 2^{nd} accuracy. In order to validate the implementation of the tracer option, more test cases have been performed with GPRS, and results are compared with the results from the Eclipse simulator.

Case 1: Homogeneous 2D test case

In this case, we have an oil-water 2D homogeneous system. The connate water saturation is 0.15. One water injection well is located at the center of the reservoir, and one production well is at the upper right corner. The tracer is injected with water from the injection well. After 1000 days of running (0.075 PVI), the concentration maps of GPRS and the Eclipse simulator are shown in Fig. 1-8.

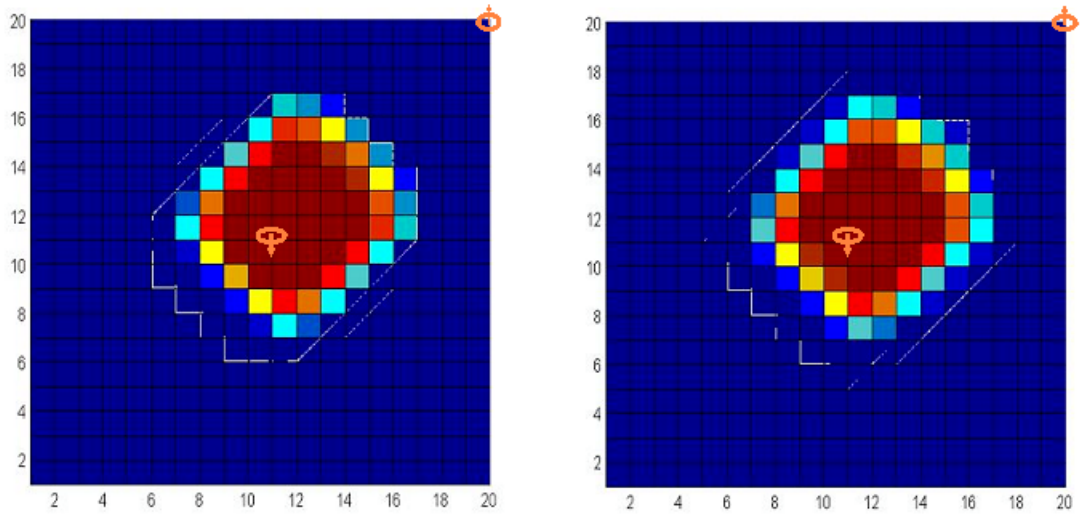

Fig. 1-8: Tracer concentration maps from GPRS (left) and Eclipse (right) results with injection at the center

The results from the two simulators match well. In order to compare the results more accurately, a cross-sectional view is provided in Fig. 1-9.

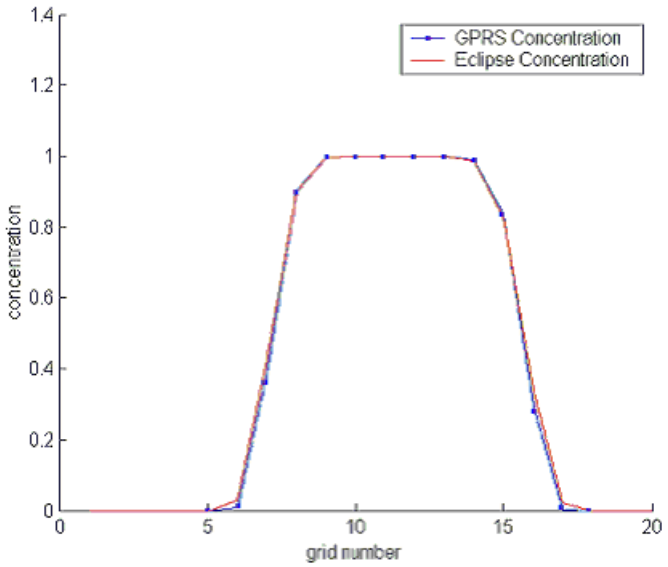

Fig. 1-9: Cross-sectional view of the concentrations at the 11th column

The cross-sectional view is given from the 11th column, in which the injection well is located. Basically, the match is very good and GPRS results seem to be a slightly better than those of the Eclipse simulator.

Case 2: Heterogeneous 2D test case

In this case, the permeability map contains four stripes with 2 kinds of permeabilities, 100md and 1000md as shown in Fig. 1-10. The injection well is located at the left lower corner, and the production well is at the upper right corner. The TVD scheme results of GPRS and the Eclipse simulator are shown in Fig. 1-10. The GPRS upstream results are also provided.

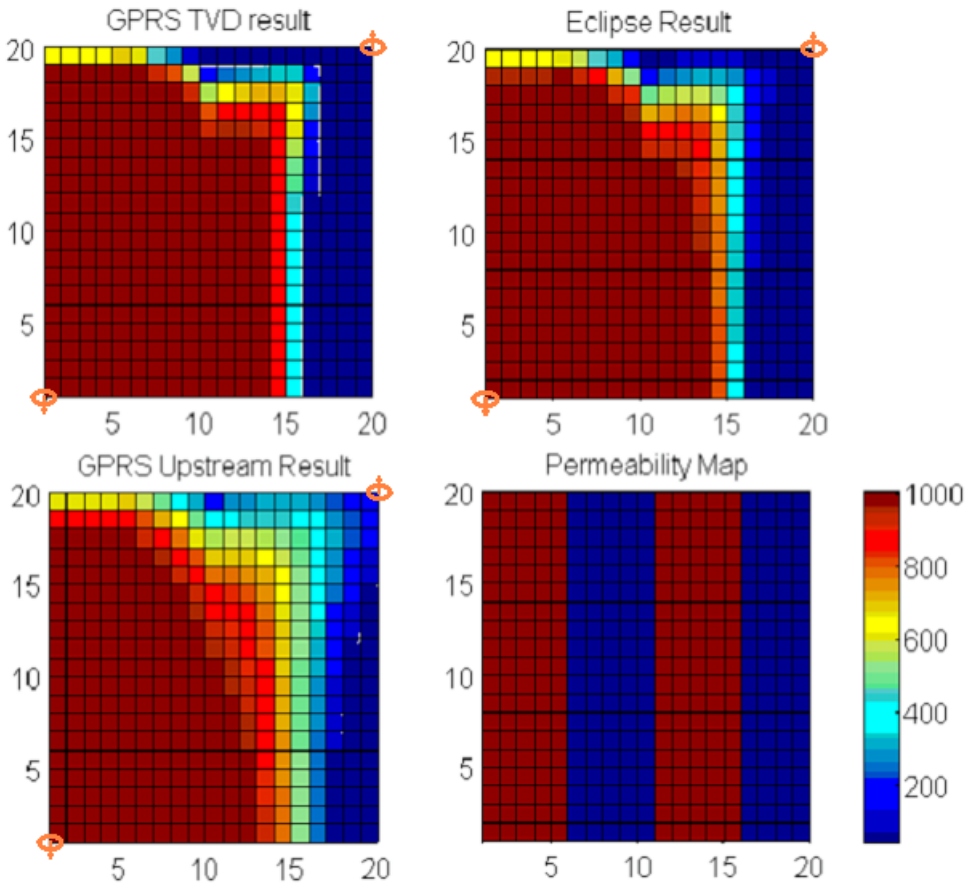

Fig. 1-10: Tracer concentration maps of GPRS and Eclipse and the permeability field map

GPRS TVD results are very similar to the results from the Eclipse simulator and both of them are much better than the upstream results. A cross-sectional view given from the 15th row provides more details about the results, as shown in Fig. 1-11.

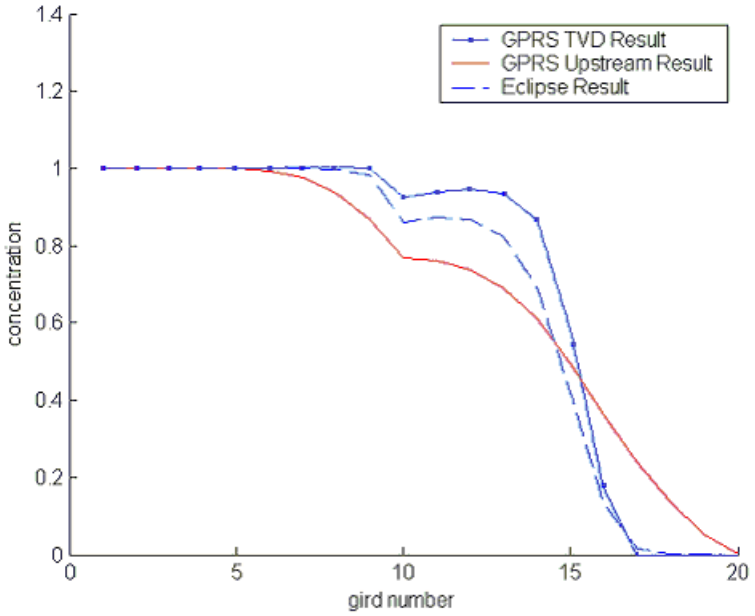

Fig. 1-11: Cross-sectional view of tracer concentration at the 15th row

GPRS TVD scheme gives the sharpest profile. From the plots in this figure, it may seem that there is some "mass balance error" among these three results. But, remember that Fig. 1-11 only shows a slice of the whole 2D problem. We do not have any significant overall mass balance errors in the tracer simulation. The tracer distributions are different, due to the different anti-diffusion properties of various schemes.

Case 3: Homogeneous 3D test case

In this case, we have an oil-water 3D homogeneous system with 20 x 20 x 3 grid. The connate water saturation is 0.15. One water injection well is located at the left lower corner of the system, and it is perforated in the top layer; the production well is at the upper right corner of the system, and it is penetrated in the bottom layer. Tracer is

injected with water from the injection well. After a few days running (PVI is equal to 0.3), the concentration maps of GPRS and Eclipse are shown in Fig. 1-12.

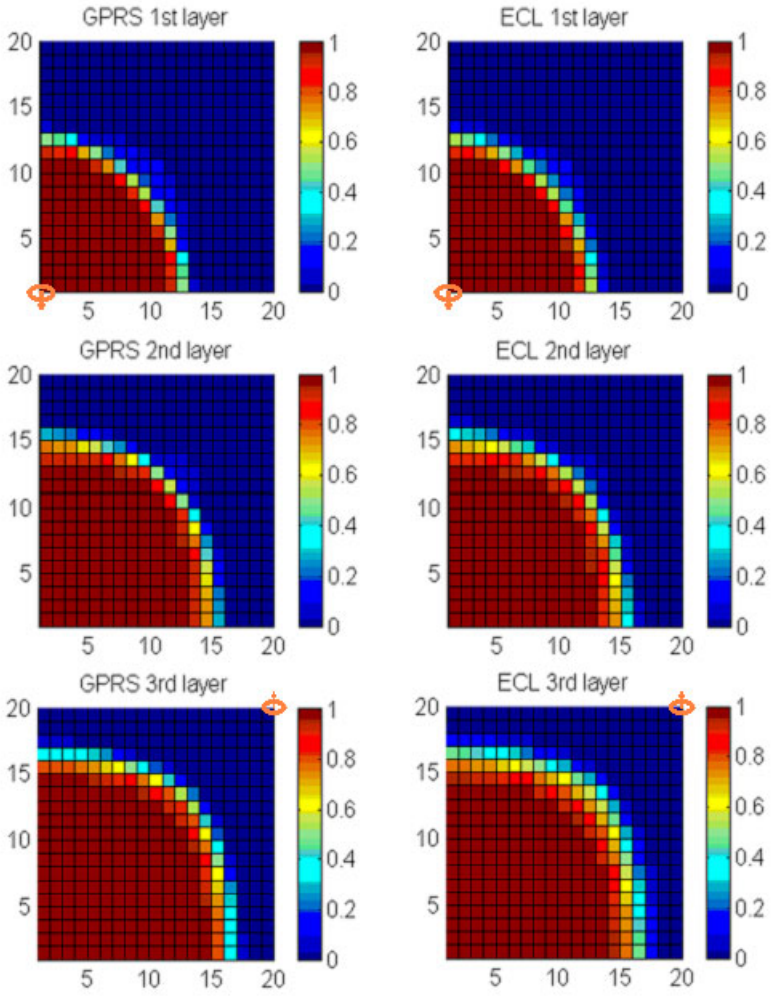

Fig. 1-12: Tracer concentration map of GPRS (left) and Eclipse (right) results

Fig. 1-13 shows the GPRS and Eclipse tracer distribution in the 1st row of the bottom layer. The two results are very similar, and the one from GPRS is a little sharper than the one from Eclipse.

From the test cases discussed above, we can conclude that the implementation of tracer and the extensions to 2D and 3D are valid. GPRS gives slightly lower numerical dispersion of the tracer front.

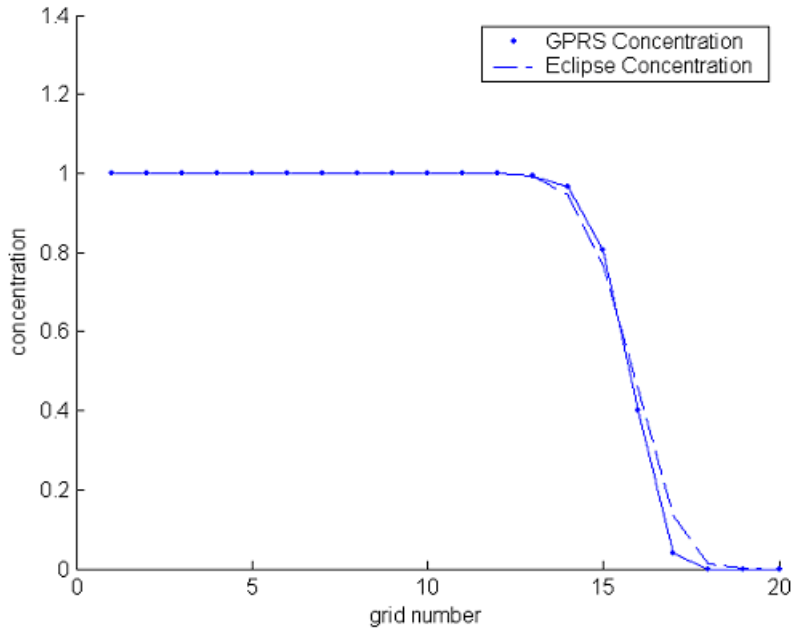

Fig. 1-13: Cross-sectional view of tracer concentration at the 1st row

In this chapter, we gave a brief introduction to tracer flow and discussed the implementation of tracer option in GPRS. We implemented both the conventional upstream scheme and several TVD schemes for tracer flow simulation. An explicit method was used to minimize computational cost and improve accuracy. The dual timestepping technique was adapted to guarantee stability of tracer transport calculations. The implementation was verified by comparing results with the Eclipse simulator. Appendices A, B, and C provide further details of the tracer option in GPRS.

Chapter 2

2. The Performance of Linear Solvers in GPRS

In reservoir simulation, discretization of the governing equations that describe the flow behavior of a particular displacement process leads to systems of nonlinear algebraic equations. Solution of these nonlinear systems is usually obtained using the iterative Newton procedure. Each Newton iteration requires solution of a sparse linear system of equations. The size of the linear system depends on the number of gridblock, or cells, in the model, and it depends on the number of unknowns per gridblock. In the IMPES - IMplicit Pressure, Explicit saturations - method (Garder and Stone, 1961; Coats, 2000), pressure is the only unknown in a gridblock. The linear system that describes the pressure in the IMPES formulation is usually diagonally dominant and nearly symmetric. The linear systems associated with the fully implicit method (FIM), on the other hand, have multiple unknowns per gridblock, are not necessarily diagonally dominant, and are highly nonsymmetric (Wallis, 1983).

In the course of a simulation run, we often take large numbers (hundreds, even thousands) of timesteps. For each timestep, several Newton iterations are usually needed to solve the nonlinear system. Repeated solution of the different linear systems, one per Newton iteration, in the course of a simulation is usually the most time consuming activity. As a result, the overall utility and performance of a reservoir simulator depends on having a robust and computationally efficient linear solver.

The performance of a particular preconditioning method depends on the nature of the target linear system. Several preconditioning methods have been used for reservoir simulation. Examples include Incomplete LU decomposition (ILU), Successive Over Relaxation (SOR), Gauss-Siedel (GS). More recently, multigrid methods have become

popular. For near-elliptic systems, where long-range coupling is important, multigrid works well. For near-hyperbolic systems, where short-range error components tend to dominate, ILU, SOR, GS and many other approaches work fine. In reservoir simulation, the pressure equation is usually near-elliptic, while the saturation equations are typically close to being hyperbolic. The mixed character of the equations and variables that describe multi-component multiphase flow poses a significant challenge to preconditioning methods.

The objective here is to improve the robustness and the performance of GPRS, especially for large-scale heterogeneous problems with coupled wells, which have multi-block completions. We focus on a two-stage preconditioning strategy that is tuned to the mixed character of the linear systems encountered with fully implicit simulation. Specifically, we concentrate on the Constrained Pressure Residual (CPR) method developed by Wallis (1983). The effectiveness of CPR for FIM linear systems on structured grids is well established (Wallis, 1983,1985). Here, we describe the components of a CPR preconditioning strategy that is particularly suited for fully implicit simulation of large-scale heterogeneous reservoir models discretized using unstructured grid. One focus area is on proper handling of implicit wells that penetrate several gridblocks.

Several preconditioners are investigated, and their performance is analyzed. Finally a set of 'optimal' parameters is found for the CPR method in GPRS.

2.1. Formulations and Solvers in GPRS

2.1.1. Formulations

GPRS is a general-purpose unstructured-grid, reservoir simulator that offers the following formulations:

- IMPES: Implicit pressure, explicit saturation and component mole fractions (Stone and Garder, 1961; Coats, 2000)
- IMPSAT: Implicit pressure and saturation, explicit component mole fractions (Cao and Aziz, 2002)

- AIM: Adaptive implicit method (Thomas and Thurnau, 1983; Forsyth and Sammon, 1986) with three levels of implicitness.
- FIM: Fully implicit method (Aziz and Settari, 1979)

The IMPSAT formulation has not been frequently used, but it is particularly suited for compositional problems. The AIM formulation in GPRS is still under further development and testing. Our effort focuses on the IMPES and FIM formulations.

In the IMPES formulation, the total number of reservoir equations is equal to the number of cells because pressure is the only unknown. Hence the computational cost per Newton step can be significantly less than other methods. IMPES is more accurate than other methods as well (Russell, 1983). However, IMPES is only conditionally stable. The time step limitation is due to the CFL condition (Russell, 1983; Aziz and Settari, 1979). Often for difficult problems, the time step is too small to be practical.

FIM treats all primary variables implicitly. It is unconditionally stable, which is the major advantage of FIM. However, the computational cost is expensive, especially for compositional models. Relatively large numerical diffusion is another drawback of the FIM model.

In GPRS, the well model is relatively simple. We have the conventional multi-block completion well model (Aziz et. al, 2003). A recent extension makes GPRS capable of handling multi-lateral wells.

GPRS provides the most frequently used well control methods, including BHP control and rate controls, namely, water, oil, liquid, and gas rate control, as shown in Fig 2-1.

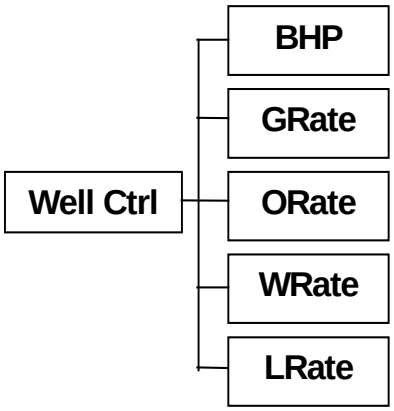

Fig. 2-1: Available well controls in GPRS

2.1.2. Linear Solvers

The conservation (nonlinear) equations are linearized by the Newton-Raphson method (Aziz and Settari, 1979; Fussel and Yanosik, 1978). A linear solver is called to deal with the resulting linear system. Due to the size and difficulty of the linear equation system, efficient linear solvers are necessary.

The available solver packages in GPRS are listed as follows:

- Solver 1: Full matrix direct solver from Linear Algebra Package (LAPACK) (Anderson, 1999)
- Solver 2: Band matrix direct solver from LAPACK (Anderson, 1999)
- Solver 3: BlitzPak iterative solver (Landmark Graphics Corporation, 1998)
- Solver 4: Generalized Minimum Residual iterative solver (GMRES) (Saad and Schultz, 1986) from the Iterative Math Library (IML) (Dongarra et al., 1996)

For large problems, iterative solvers are memory efficient and much faster than direct solvers. Preconditioners are used to accelerate the convergence rate of iterative methods. Even for relatively simple cases, solvers may not converge without a high-quality preconditioner. Beside the conventional preconditioners, like diagonal, ILU decomposition, GPRS has the Algebraic Multi-Grid preconditioner (AMG), and the Constrained Pressure Residual (CPR), which is a two-stage preconditioner (Wallis, 1983). AMG preconditioner is suitable for near-elliptic problems, e.g. the pressure

equation in IMPES. CPR can handle FIM formulation and is based on a pressure de-coupling strategy.

The available preconditioners in GPRS are:

- Pre1: Diagonal scaling
- Pre2: Block diagonal scaling
- Pre3: Incomplete LU factorization
- Pre4: TrueIMPES CPR (AMG + ILU)
- Pre5: QuasiMPES CPR (AMG + ILU)
- Pre6: AMG (for IMPES only)
- Pre7: TrueIMPES CPR (SAMG + ILU)
- Pre8: QuasiIMPES CPR (SAMG + ILU)
- Pre9: SAMG (for IMPES only)

We use the Algebraic Multi-Grid (AMG) code developed by Ruge and Stuben (1987). We recently obtained Stuben's latest AMG solver – SAMG (Stuben, 2003), which will be discussed in Section ***2.3.2***.

2.1.3. Constrained Pressure Residual (CPR) Method

While there are several preconditioning options in GPRS, the most successful one, and the subject of this report, is the two-stage CPR approach. The general form of a sequential two-stage preconditioner is given by

$$M_{1,2}^{-1} = M_2^{-1} [\, I - A M_1^{-1} \,] + M_1^{-1} \tag{2-1}$$

where M_1 and M_2 are the preconditioning matrices for the first and second stages, respectively. Wallis (1983) proposed a special two-stage preconditioner to solve the linear systems associated with the fully implicit method, which he called the CPR (Constrained Pressure Residual) method. In CPR, the first stage is an approximate solution of a global system that resembles the pressure equation of the IMPES

formulation. The second stage preconditioner is a local operator applied to the fully implicit matrix *A*. The CPR method has proved to be an excellent approach, in terms of robustness and computational efficiency (Wallis, 1983, 1985).

A description of the CPR components follows. We are given the following FIM linear system

$$A\,\vec{u} = \vec{r}\,, \tag{2-2}$$

where *A* is the coefficient matrix of the FIM linear system, $\vec{u}$ is the vector of unknowns, and $\vec{r}$ is the residual. We permute the above system by ordering the pressure equations and unknowns last and ordering all the other equations and unknowns first. This yields

$$\begin{bmatrix} A_{SS} & A_{SP} \\ A_{PS} & A_{PP} \end{bmatrix} \begin{bmatrix} u_S \\ u_P \end{bmatrix} = \begin{bmatrix} r_S \\ r_P \end{bmatrix} \tag{2-3}$$

where subscript *P* refers to pressure and subscript *S* refers to the other unknowns. The first step in CPR is to apply an IMPES-like reduction to Eq. 2-3. Both sides of Eq. 2-3 are multiplied by a matrix of the form

$$N = \begin{bmatrix} I & 0 \\ Q & I \end{bmatrix}, \tag{2-4}$$

and that gives

$$\begin{bmatrix} A_{SS} & A_{SP} \\ \hat{A}_{PS} & \hat{A}_{PP} \end{bmatrix} \begin{bmatrix} u_S \\ u_P \end{bmatrix} = \begin{bmatrix} r_S \\ \hat{r}_P \end{bmatrix}. \tag{2-5}$$

The pressure matrix is extracted as follows

$$A_p = C^T N\, AC\,, \tag{2-6}$$

where

$$C^T = \begin{bmatrix} 0 & I \end{bmatrix}. \tag{2-7}$$

The effectiveness of the CPR method depends, to a great extent, on the quality of the pressure matrix, A_p. We employ two ways for the IMPES-like reduction step that produces A_p, namely, Quasi-IMPES and True-IMPES. For the Quasi-IMPES scheme, we ignore the off-diagonal A_{PS} terms, and we eliminate the diagonal A_{PS} part using the inverse of the block-diagonal of A. We use a similar approach for the True-IMPES scheme, except that we work with a Jacobian matrix consistent with the IMPES formulation. That is, derivatives of transmissibility with respect to pressure and saturation(s) are zeroed out in the Jacobian matrix before the reduction is performed.

Using the previous definitions, the first-stage preconditioner can be expressed as

$$M_1^{-1} = C\, A_p^{-1}\, C^T N\,. \tag{2-8}$$

The second-stage preconditioner, M_2^{-1}, is a local operator that is applied to the entire system. Finally, the full form of the CPR operator is

$$M_{CPR}^{-1} = M_2^{-1}\,[\,I - AC\, A_p^{-1}\, C^T N\,] + C\, A_p^{-1}\, C^T N\,. \tag{2-9}$$

In practice, the above operations are performed in the following sequence:

(1)	$r_p = C^T N r$	restrict
(2)	$A_p p = r_p$	solve
(3)	$\delta u = Cp$	interpolate
(4)	$r = r - A\delta u$	update
(5)	$M_2 u = r$	solve
(6)	$u = u + \delta u$	correct

We use algebraic multigrid (AMG) preconditioned GMRES to solve the pressure system of the first stage. We pay special attention in the construction of A_p so that the system is well suited for AMG. In the second stage, we operate on the whole system to reduce the error, which is dominated by high frequency components. A local preconditioner, such as ILU, is an appropriate choice for the second stage. In the CPR approach, we pay for de-coupling the system, but we gain in two aspects: first, working on the low frequency error

components, which are usually dominated by the near-elliptic pressure field, separately from the typically high frequency errors associated with the saturation field(s) is very efficient. Second, AMG is very efficient for the near-elliptic pressure-like system of equations. CPR has been proved to be the most promising preconditioner for fully implicit simulations (Cao, 2002).

2.2. IMPES Formulation

Since the first stage of the CPR method is very similar to the IMPES formulation, it is worthwhile to investigate the IMPES model in detail before we get into the more complex FIM model. In this section, we analyze the properties of the pressure matrix. Then, we will discuss the performance of several AMG-based preconditioners for systems with coupled wells.

2.2.1. Properties of the Pressure Matrix

Understanding the properties of the discrete system of equations is crucial for devising efficient linear solution strategies. In fact, many solvers and preconditioners have some requirements on the matrix. Generally, solvers designed for special kinds of matrices are more efficient than general, or black-box, solvers. For example, AMG is very well suited for elliptic systems, but is not an appropriate choice for the FIM linear systems, which have a mixed hyperbolic-elliptic character.

We analyze the IMPES method. First, we show the analytical expressions for the important terms in the Jacobian matrix. Then, we discuss the properties of the linear system for various well settings. Since it is very easy to guarantee diagonal elements to be positive, the focus of the investigation is the row sum of the IMPES Method. AMG suffers from the presence of positive off-diagonal entries as well as from negative row-sums (Ruge and Stuben, 1987).

To simplify the analysis, we work with an oil-water system. We ignore the rock compressibility. In fact, we can include the rock compressibility into the liquid compressibility. This is a good assumption when the fluid compressibilities are small,

which is reasonable for liquid systems. Capillary pressure is not considered in the following discussion.

PART I: Reservoir Equations

In order to keep the formulae relatively simple, we consider a small linear system, i.e. each cell has only two neighboring cells. In fact, the dimensions have nothing to do with the following procedure and conclusion.

The governing equations of the oil and water components are (Aziz et al. 2003):

$$R_{o,i} = -T_{o,i-1/2}(P_{o,i-1} - P_{o,i}) - T_{o,i+1/2}(P_{o,i+1} - P_{o,i}) - T_o^w(P^w + G - P_{o,i}) + \frac{V_i}{\Delta t}\Delta_t(\phi S_o b_o)_i = 0 \quad (2\text{-}10)$$

$$R_{w,i} = -T_{w,i-1/2}(P_{w,i-1} - P_{w,i}) - T_{w,i+1/2}(P_{w,i+1} - P_{w,i}) - T_w^w(P^w + G - P_{w,i}) + \frac{V_i}{\Delta t}\Delta_t(\phi S_w b_w)_i = 0 \quad (2\text{-}11)$$

The only difference between Eq. 2-10 and 2-11 are the subscripts w and o, which stand for water and oil, respectively. The superscript w stands for the well. R_w is the residual of the water component equation and R_o is the residual of the oil component equation. T_w is the transmissibility of the water phase between cells, T_o is the transmissibility of the oil phase between cells, T_w^w is the well transmissibility of water phase and T_o^w is the well transmissibility of oil phase. P_w is the pressure of the water phase and P_o is the pressure of the oil phase. Since we do not consider the capillary pressure, there is no difference between P_w and P_o. G is the hydraulic head from the pressure reference point to the current perforation, and V_i is the bulk volume of cell i. S_w and S_o stand for the water and oil saturations, respectively. b_w and b_o are the reciprocals of the formation volume factor (FVF) of water phase and oil phase. Δt is the timestep. These two equations contain well terms. These well terms are for a production well. This is because

we have written them in terms of the phase mobility; otherwise, the total mobility should be used instead (Almehaideb and Aziz, 1989).

After linearization of Eq. 2-10 and 2-11 using the Newton-Raphson method, we obtain the Jacobian matrix, which has two linear equations per cell. We write the equations for one cell as Eq. 2-12:

$$\begin{array}{c} \\ o \\ w \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} J^o_{pp} & J^o_{ps} \\ J^o_{sp} & F^o_{ss} \end{bmatrix}_{i-1} \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} J_{pp} & J_{ps} \\ J_{sp} & J_{ss} \end{bmatrix}_{i} \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} J^o_{pp} & J^o_{ps} \\ J^o_{sp} & J^o_{ss} \end{bmatrix}_{i+1} \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} P^w \\ \begin{bmatrix} J^w_{pp} \\ J^w_{sp} \end{bmatrix}_{w} \end{array} \quad (2\text{-}12)$$

In Eqs. 2-10 and 2-11, the oil component equation is always selected as the primary equation and the water component equation as the secondary equation. The subscript p stands for primary, if it is the first one in subscript, and for pressure, if it is the second subscript; the subscript s stands for secondary, if it is the first, and for saturation, if it is the second. The superscript o stands for the off-diagonal term. Term J is equal to the sum of an accumulation term, A, and a flux term, F. Since we do not have an accumulation contribution in the off-diagonal blocks, Eq. 2-12 reduces to Eq. 2-13:

$$\begin{array}{c} \\ o \\ w \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} F^o_{pp} & F^o_{ps} \\ F^o_{sp} & F^o_{ss} \end{bmatrix}_{i-1} \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} J_{pp} & J_{ps} \\ J_{sp} & J_{ss} \end{bmatrix}_{i} \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} F^o_{pp} & F^o_{ps} \\ F^o_{sp} & F^o_{ss} \end{bmatrix}_{i+1} \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} P^w \\ \begin{bmatrix} F^w_{pp} \\ F^w_{sp} \end{bmatrix}_{w} \end{array} \quad (2\text{-}13)$$

For example, F^o_{ps} denotes the derivative of the flux to the neighboring cell in the primary equation with respect to the neighbor's saturation. Eq. 2-13 is for one cell. If we combine the equations for all the cells, we obtain the FIM system, as shown in Fig. 2-2 (2D system).

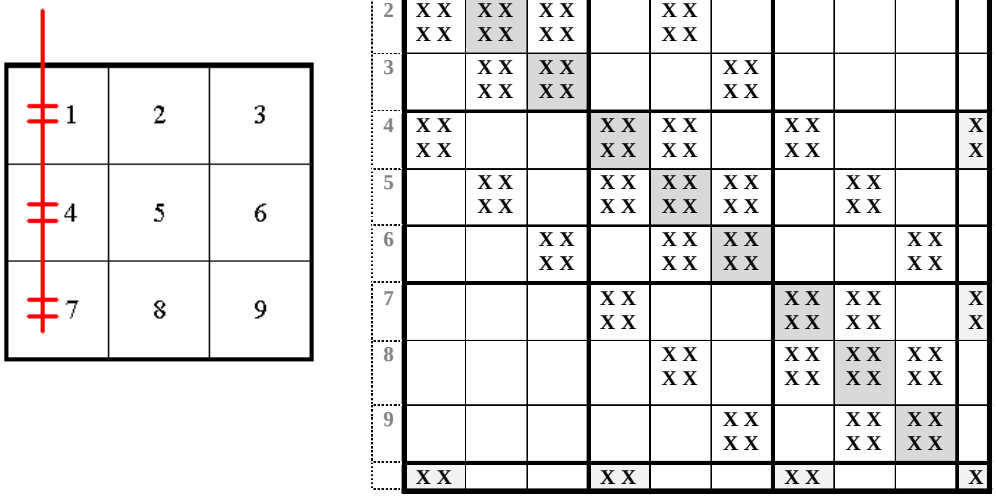

Fig. 2-2: A 3x3 system with one well and its FIM matrix with block priority ordering (after Aziz et al. 2003)

In order to reduce the matrix in Fig. 2-2 to an IMPES system, we treat the transmissibility between cells explicitly. Then the flux terms between cells are not functions of saturation any more, i.e. the derivatives of flux term with respect to saturation are zero. However, the well terms are always treated implicitly. Consequently, Eq. 2-13 reduces to Eq. 2-14:

$$\begin{matrix} o \\ w \end{matrix} \quad \cdots \quad \begin{matrix} P & S \end{matrix} \begin{bmatrix} F^{o}_{pp} & 0 \\ F^{o}_{sp} & 0 \end{bmatrix}_{i-1} \begin{matrix} P & S \end{matrix} \begin{bmatrix} J_{pp} & A_{ps} \\ J_{sp} & A_{ss} \end{bmatrix}_{i} \begin{matrix} P & S \end{matrix} \begin{bmatrix} F^{o}_{pp} & 0 \\ F^{o}_{sp} & 0 \end{bmatrix}_{i+1} \cdots \begin{matrix} P^{w} \end{matrix} \begin{bmatrix} F^{w}_{pp} \\ F^{w}_{sp} \end{bmatrix}_{w} \tag{2-14}$$

Define the compressibilities of the oil and water phases as follows:

$$C_o = \partial b_o / \partial P > 0 \tag{2-15}$$

$$C_w = \partial b_w / \partial P > 0 \tag{2-16}$$

The accumulation terms for oil and water can be expressed as Eq. 2-17 and Eq. 2-18 (Aziz, et al. 2003):

$$A_o = \frac{V\phi}{\Delta t}[(b_o S_o)^{n+1} - (b_o S_o)^n] \tag{2-17}$$

$$A_w = \frac{V\phi}{\Delta t}[(b_w S_w)^{n+1} - (b_w S_w)^n] \tag{2-18}$$

The derivatives of accumulation terms with respect to pressure and saturation take the following form:

$$A_{pp} = \partial A_o / \partial P = \frac{V\phi}{\Delta t}(C_o S_o)^{n+1} \tag{2-19}$$

$$A_{sp} = \partial A_w / \partial P = \frac{V\phi}{\Delta t}(C_w S_w)^{n+1} \tag{2-20}$$

$$A_{ps} = \partial A_o / \partial S_w = -\frac{V\phi}{\Delta t} b_o^{n+1} \tag{2-21}$$

$$A_{ss} = \partial A_w / \partial S_w = \frac{V\phi}{\Delta t} b_w^{n+1} \tag{2-22}$$

The derivatives of flux terms with respect to pressure can be written as follows:

$$F_{pp} = \sum [T_o + T_o^w - \frac{\partial T_o^w}{\partial P}(P^w + G - P_{o,i})] \tag{2-23}$$

$$F_{sp} = \sum [T_w + T_w^w - \frac{\partial T_w^w}{\partial P}(P^w + G - P_{o,i})] \tag{2-24}$$

$$F_{pp}^{\ o} = -T_o \tag{2-25}$$

$$F_{sp}^{\ o} = -T_w \tag{2-26}$$

$$F_{pp}^{\ w} = -T_o^w \tag{2-27}$$

$$F_{sp}^{\ w} = -T_w^w \tag{2-28}$$

Local Gaussian elimination is used to manipulate Eq. 2-14. Use the secondary equation to eliminate the only saturation term in the primary equation; the resulting equation is written as

$$\begin{array}{c} \\ o \\ w \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} F_{pp}^{o} - A_{ps}A_{ss}^{-1}F_{sp}^{o} & 0 \\ F_{sp}^{o} & 0 \end{bmatrix}_{i-1} \end{array} \begin{array}{c} \begin{array}{cc} P & S \end{array} \\ \begin{bmatrix} J_{pp} - A_{ps}A_{ss}^{-1}J_{sp} & 0 \\ J_{sp} & A_{ss} \end{bmatrix}_{i} \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} \\ \cdots \end{array} \begin{array}{c} P^{w} \\ \begin{bmatrix} F_{pp}^{w} - A_{ps}A_{ss}^{-1}F_{sp}^{w} \\ F_{sp}^{w} \end{bmatrix}_{w} \end{array}$$

(2-29)

The corresponding system is shown in Fig. 2-3. The pressure matrix can be extracted as shown in Fig. 2-4. We might get some fill-in terms, shown as ‘o’ in Fig. 2-4, due to the fully implicit treatment of the well. This matrix is what we need to solve in the IMPES model.

	P S	P S	P S	P S	P S	P S	P S	P S	P S	P
1	**X** X X	**X** X		**X** X						**X** X
2	**X** X	**X** X X	**X** X		**X** X					
3		**X** X	**X** X X			**X** X				
4	**X** X			**X** X X	**X** X		**X** X			**X** X
5		**X** X		**X** X	**X** X X	**X** X		**X** X		
6			**X** X		X X	X X X X			**X** X	
7				**X** X			**X** X X	**X** X		**X** X
8					**X** X		**X** X	**X** X X	**X** X	
9						**X** X		**X** X	**X** X X	
	X X			X X			X X			**X**

Fig. 2-3: Matrix after local Gaussian elimination

	P	P	P	P	P	P	P	P	P	P
1	X	X		X						X
2	X	X	X		X					
3		X	X			X				
4	X			X	X		X			X
5		X		X	X	X		X		
6			X		X	X			X	
7				X			X	X		X
8					X		X	X	X	
9						X		X	X	
	X	O		X	O		X	O		X

Fig. 2-4: IMPES pressure matrix

Based on previous formulas, the terms in the pressure matrix can be written as follows:

$$A_p{}^d = (A_{pp} + F_{pp}) - A_{ps} \cdot A_{ss}{}^{-1} \cdot (A_{sp} + F_{sp}) \tag{2-30}$$

$$A_p{}^o = F_{pp}{}^o - A_{ps} \cdot A_{ss}{}^{-1} \cdot F_{sp}{}^o \tag{2-31}$$

$$A_p{}^w = F_{pp}{}^w - A_{ps} \cdot A_{ss}{}^{-1} \cdot F_{sp}{}^w \tag{2-32}$$

Here, $A_p{}^d$ is the diagonal term of the pressure matrix, $A_p{}^o$ is the off-diagonal term of the pressure matrix and $A_p{}^w$ stands for the well term.

We are interested in the sign of the row sum of the pressure matrix.

$$Rowsum = \frac{V\phi}{\Delta t}[C_o(1 - S_w) + \frac{b_o}{b_w}(C_w S_w)] - (\frac{\partial T_o{}^w}{\partial P} + \frac{b_o}{b_w}\frac{\partial T_w{}^w}{\partial P})(P^w - P) \tag{2-33}$$

The first term of Eq. 2-33 is the accumulation term, which is positive and the second term is the well term. Eq. 2-33 has nothing to with flux term between cells. If the phase compressibilities of the system are in the normal range, the accumulation term is much

larger than the well term, so the row sum is positive. If the system is incompressible, we can ignore the first term and the well term becomes dominant.

Since the well transmissibilities are defined as (Aziz et al. 2003):

$$T_o^w = \frac{k k_{ro} b_o}{\mu_o} WI \tag{2-34}$$

$$T_w^w = \frac{k k_{rw} b_w}{\mu_w} WI \tag{2-35}$$

where, k is the permeability of the cell, k_{ro} and k_{rw} are the relative permeabilities of oil and water phase respectively, μ_o and μ_w are the viscosities of oil and water phases and WI is the well index.

Then,

$$Rowsum = -b_o WI\,[k_{ro}\frac{\partial}{\partial P}(\frac{1}{\mu_o}) + k_{rw}\frac{\partial}{\partial P}(\frac{1}{\mu_w})](P^{well} - P) \tag{2-36}$$

Generally, $\frac{\partial}{\partial P}(\frac{1}{\mu_o}) < 0$ and $\frac{\partial}{\partial P}(\frac{1}{\mu_w}) < 0$, so $(P^{well} - P)$, i.e. the well type determines the sign of the row sum. It could be negative if the well is a producer. If the system is incompressible and there is no well perforation in the cell, then the row sum is zero, which makes sense.

PART II: Well Equations

Here, we analyze the well terms in the Jacobian matrix. The objective is to see how the IMPES procedure will effect the row sum of the well equations. We focus on the row-sum property.

Assume we have an oil rate controlled production well (Aziz et al., 2003):

$$R^w = \sum_{i=1}^{NP} \sum T_o^w (P^w - P_{o,i}) + q_o = 0 \tag{2-37}$$

where, q_o is the specified oil rate, R^w is well equation residual and NP is the number of perforations of the well.

Linearization of Eq. 2-37 adds one row to the matrix shown in Fig. 2-2.

$$\begin{array}{cccccccccc} & & P & S & P & S & P & S & & P^w \\ well & ... & [0 & 0] & [F_p^w & F_s^{well}] & [0 & 0] & ... & \sum_l^{NP} T_o^w \end{array} \tag{2-38}$$

$$F_p^w = -T_o^w - \frac{\partial T_o^w}{\partial P}(P_i - P^w) \tag{2-39}$$

$$F_s^w = -\frac{\partial T_o^w}{\partial S_w}(P_i - P^w) \tag{2-40}$$

When we use the IMPES method, we always treat the well implicitly. To eliminate the saturation derivation in the well equation, new pressure derivative terms will be introduced to the neighbors of the current perforated well cell. After the elimination, we have

$$\begin{array}{cccccccccc} & & P & S & P & S & P & S & & P^w \\ rate\ control & ... & [0 & 0] & [F_p^w & 0] & [N_p^w & 0] & ... & \sum_l^{NP} T_o^w \end{array} \tag{2-41}$$

where, the new fill-in term is

$$N_p^w = -\frac{\partial T_o^w}{\partial S_w}(P_i - P^w) A_{ss}^{-1} T_w \tag{2-42}$$

We notice that $\frac{\partial T_o^w}{\partial S_w}$ is always negative. So N_p^w is positive when the well is a producer. This is a positive off-diagonal term, which is harmful to the performance of AMG. Generally, this term is quite small.

The row-sum of the well equation for a producer is

$$\begin{aligned} Rowsum &= \sum_i^{nPerf} \{\frac{\partial T_o^w}{\partial S}(P_i - P^w) A_{ss}^{-1}[A_{sp} - \frac{\partial T_w^w}{\partial P}(P^w - P_i)]\} \\ &- \sum_i^{nPerf} \frac{\partial T_o^w}{\partial P}(P_i - P^w) \end{aligned} \tag{2-43}$$

If the system is compressible, we have

$$A_{sp} = \frac{V\phi}{\Delta t}(C_w S_w)^{n+1} >> \frac{\partial T_w^w}{\partial P}(P^w - P_i) \tag{2-44}$$

then

$$Rowsum \approx \sum_i^{NP} (\frac{\partial T_o^w}{\partial S_w}\frac{C_w S_w}{b_w} - \frac{\partial T_o^w}{\partial P})(P_i - P^w) \tag{2-45}$$

$\frac{\partial T_o^w}{\partial S_w}$ is always negative and $\frac{\partial T_o^w}{\partial P}$ is positive, since the FVF usually is much more important than the viscosity. Hence, we have a negative row sum when the well is a producer.

If we have an incompressible system, the row sum can be written as Eq. 2-46:

$$\begin{aligned} Rowsum &= \sum_i^{nPerf} \{\frac{\partial T_o^w}{\partial S_w}(P_i - P^{well}) A_{ss}^{-1}[-\frac{\partial T_w^w}{\partial P}(P^w - P_i)]\} \\ &- \sum_i^{nPerf} \frac{\partial T_o^w}{\partial P}(P_i - P^w) \end{aligned} \tag{2-46}$$

Generally, $\frac{\partial T_w^w}{\partial P}$ is negative, so the first term is positive. The sign of second term depends on the type of the well. If the system is incompressible and viscosity is not a function of pressure, the row sum is constant zero.

Using a similar procedure, we can derive the row-sum of the well equation for a water injector as follows:

$$Rowsum = \sum_i^{nPerf} \{\frac{\partial T_T^w}{\partial S_w}(P_i - P^w) A_{ss}^{-1}[A_{sp} - \frac{\partial T_T^w}{\partial P}(P^w - P_i)]\} - \sum_i^{nPerf} \frac{\partial T_T^w}{\partial P}(P_i - P^w) \quad (2\text{-}47)$$

where T_T^w is the total transmissibility of the perforations. Eq. 2-47 is slightly different from the equation for a producer.

Table 2-1: Summary of properties of the IMPES matrix

Equations	Well Type	Diagonal	Row Sum
Incompressible System			
Reservoir Equ.	Producer	+	-
	Injector	+	+
Well Equ.	Producer	+	0
	Injector	+	0
Compressible System			
Reservoir Equ.	Producer	+	+
	Injector	+	+
Well Equ.	Producer	+	-
	Injector	+	+

Table 2-1 summarizes the properties of the IMPES pressure matrix for different conditions. For the reservoir equations, the row sum is positive as long as the system is compressible. However, if rate controlled production wells are present in a compressible system, we may get a negative row sum for the well equations, which may cause problems for the AMG solver.

In order to demonstrate the properties listed in Table 2-1, we setup a 2D oil-water compressible system, with 2x2 grid. One well is completed in 2 blocks, as shown in Fig. 2-5. Unit transmissibilities for both cells and perforations are used. Appendix D gives the GPRS data files, in which you can find out the detail setting.

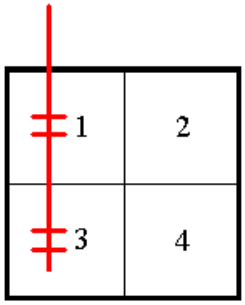

Fig. 2-5: 2x2 system used to demonstrate the row sum properties

If the well is a producer with oil rate control, the IMPES pressure matrix and row sum results are shown in Fig. 2-6.

$$A_p = \begin{bmatrix} 3.0502 & -1.0000 & -1.0000 & 0 & -1.0000 \\ -1.0000 & 2.0497 & 0 & -1.0000 & 0 \\ -1.0000 & 0 & 3.0502 & -1.0000 & -1.0000 \\ 0 & -1.0000 & -1.0000 & 2.0497 & 0 \\ -1.0005 & 0 & -1.0005 & 0 & 2.0000 \end{bmatrix} \quad RS(A_p) = \begin{bmatrix} 0.0502 \\ 0.0497 \\ 0.0502 \\ 0.0497 \\ -0.0011 \end{bmatrix}$$

Fig. 2-6: IMPES matrix and the row sums for a 2x2 system with an oil rate controlled producer

If the well is an injector with water rate control, its IMPES matrix and row-sum results are shown in Fig. 2-7.

$$A_p = \begin{bmatrix} 3.0753 & -1.0000 & -1.0000 & 0 & 0 \\ -1.0000 & 2.0769 & 0 & -1.0000 & 0 \\ -1.0000 & 0 & 3.0753 & -1.0000 & 0 \\ 0 & -1.0000 & -1.0000 & 2.0769 & 0 \\ -1.2090 & 0 & -1.2090 & 0 & 2.4217 \end{bmatrix} \quad RS(A_p) = \begin{bmatrix} 1.0753 \\ 0.0769 \\ 1.0753 \\ 0.0769 \\ 0.0037 \end{bmatrix}$$

Fig. 2-7: IMPES matrix and the row sums for a 2x2 system with a water rate controlled injector

We notice that all the elements in the row-sum vector for a system with an injector are positive. However, the row-sum of the well equation in a system with a producer is negative. Hence these numerical tests demonstrate the properties listed in Table 2-1.

The three preconditioning matrices for the 2x2 system with a production well are shown below:

$$Schur = \begin{bmatrix} 2.5499 & -1.0000 & -1.5003 & 0 \\ -1.0000 & 2.0497 & 0 & -1.0000 \\ -1.5003 & 0 & 2.5499 & -1.0000 \\ 0 & -1.0000 & -1.0000 & 2.0497 \end{bmatrix} \quad RS(Schur) = \begin{bmatrix} 0.0497 \\ 0.0497 \\ 0.0497 \\ 0.0497 \end{bmatrix}$$

(a) Schur complement matrix and its row-sum vector

$$RS\_pre = \begin{bmatrix} 2.0497 & -1.0000 & -1.0000 & 0 \\ -1.0000 & 2.0497 & 0 & -1.0000 \\ -1.0000 & 0 & 2.0497 & -1.0000 \\ 0 & -1.0000 & -1.0000 & 2.0497 \end{bmatrix} \quad RS(RS\_pre) = \begin{bmatrix} 0.0497 \\ 0.0497 \\ 0.0497 \\ 0.0497 \end{bmatrix}$$

(b) Row-sum preconditioning matrix and its row-sum vector

$$A_{rr}\_pre = \begin{bmatrix} 3.0502 & -1.0000 & -1.0000 & 0 \\ -1.0000 & 2.0497 & 0 & -1.0000 \\ -1.0000 & 0 & 3.0502 & -1.0000 \\ 0 & -1.0000 & -1.0000 & 2.0497 \end{bmatrix} \quad RS(A_{rr}\_pre) = \begin{bmatrix} 1.0502 \\ 0.0497 \\ 1.0502 \\ 0.0497 \end{bmatrix}$$

(c) A_{rr} preconditioning matrix and its row-sum vector

Fig. 2-8: Three preconditioning matrices and their row sums

2.2.2. Row-Sum Preconditioner

The pressure matrix can be divided into four parts, A_{rr}, A_{rw}, A_{wr} and A_{ww} as shown in Fig. 2-9, where the first subscript refers to the equation, and the second subscript refers to the variable. Here, *r* and *w* denote reservoir and wells related quantities, respectively.

Since we found that implicit wells can cause problems for AMG, we can use the rate constraint to eliminate the well couplings from the reservoir equations. The resulting Schur complement matrix has reservoir equations only, but with additional fill-in (see A_{rr}^{*} in Fig. 2-9). The Schur complement can be written as follows:

$$A_{rr}^{*} = A_{rr} - A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr} \tag{2-48}$$

The second term in the right hand side has the same size as A_{rr}; however, A_{rr}^{*} has new fill-in terms compared with A_{rr}. The number of induced terms in a cell penetrated by a well is equal to the number of perforations in the well. It is likely that many of the induced terms will occupy new fill-in positions. As a result, the computational cost of calculating the full Schur complement can be quite large.

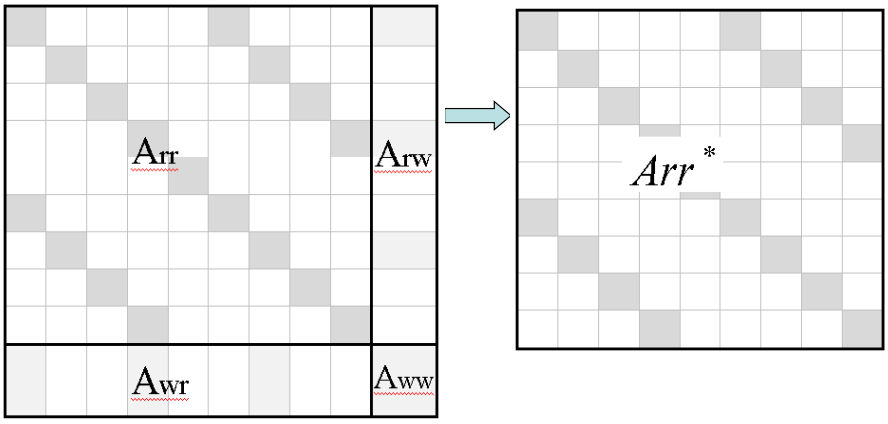

Fig. 2-9: Structure of IMPES pressure matrix and reduced system

The explicit form of the coefficient matrix is not needed by GMRES since only the result of applying the matrix to a vector (i.e. matrix-vector multiplication) is of interest.

$$A_{rr}^{*} \cdot \vec{v} = A_{rr} \cdot \vec{v} - A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr} \cdot \vec{v} \tag{2-49}$$

As shown in Eq. 2-49, we can get the result of matrix-vector multiplication $A_{rr}^{*} \cdot \vec{v}$ without knowing the explicit form of the coefficient matrix A_{rr}^{*}.

However, it is not the case for the AMG preconditioner. AMG needs to know the explicit form of the matrix, i.e. every element in the matrix should be known. Table 2-2.lists the three different matrices we use as precondtioners.

Table 2-2: Summary of the properties of pressure matrix

	AMG Pre
1	$S = A_{rr} - A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr}$
2	A_{rr}
3	$A_{rr} - diag(RS(A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr}))$

The first choice in Table 2-2 is the Schur complement. As mentioned before, it contains full information, but it can be very expensive to compute. In the second choice, the well terms are simply dropped, and A_{rr} is directly used as a preconditioner. This proves to be a good preconditioner in many cases. For example, A_{rr} is a good approximation of the overall system if the well indices are relatively small. The third choice is the row-sum preconditioner, which is the preferred method.

The row-sum preconditioner is an approximation to the Schur complement and can be performed efficiently as follows:

$$RS\,(A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr}) = A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr} \cdot \vec{e} \qquad (2\text{-}50)$$

where $\vec{e}$ is the vector with all elements equal to unity. We do not need to work out $A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr}$, we just calculate Eq. 2-50 from right to left. The interim results are always vectors, and the computational cost is low. We subtract the row-sum result of $A_{rw} \cdot A_{ww}^{-1} \cdot A_{wr}$ from the diagonal of A_{rr}. In another words, the well terms are approximately compensated for in the diagonal of matrix A_{rr}.

We analyzed the performance of the three preconditioners. Fig. 2-10 shows a 3D oil-water system, which has 141,900 cells (110x30x43). There are 11 vertical fully

penetrating wells in the reservoir. Ten injection wells with water rate control are located near a corner. There is a production well with BHP control in the middle of the reservoir.

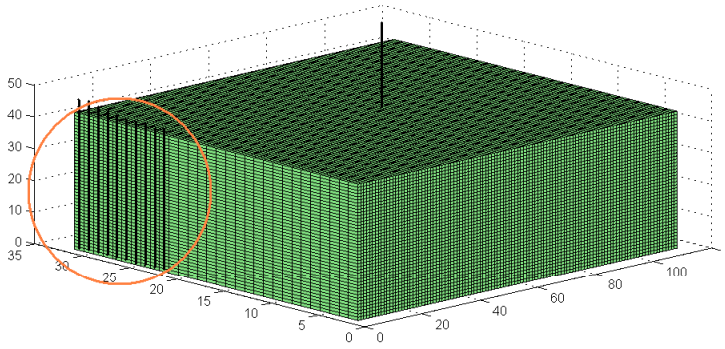

Fig. 2-10: Reservoir map of test case

We solved the test case with the IMPES method. The solver is GMRES with the three preconditioners listed in Table 2-2. Figs. 2-11 to 2-13 show the solver performance. In each figure, there are several lines. Each line stands for a Newton literation. For each Newton iteration, the linear system is solved by preconditioned GMRES. The convergence rate tell us how good these preconditioners are.

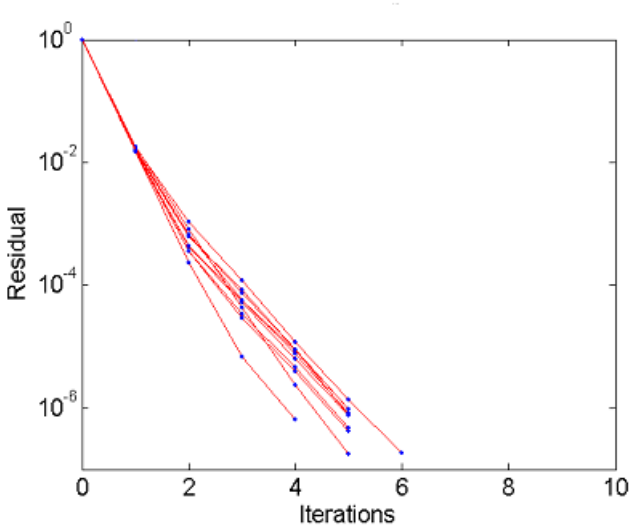

Fig. 2-11: Residual reduction for GMRES with the Schur complement preconditioner

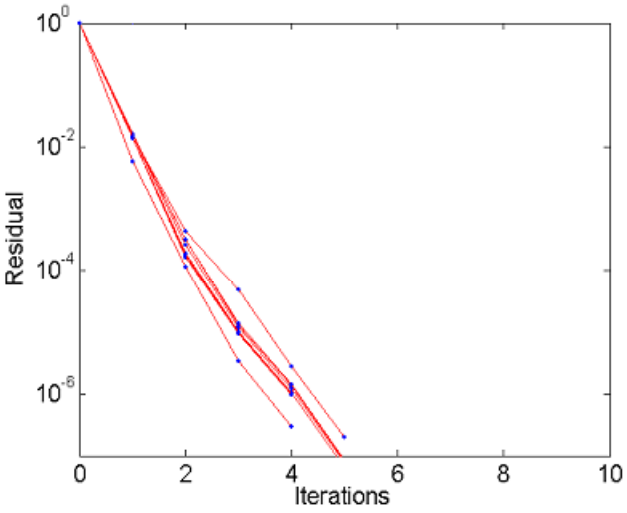

Fig. 2-12: Residual reductions for GMRES with the A_{rr} preconditioner

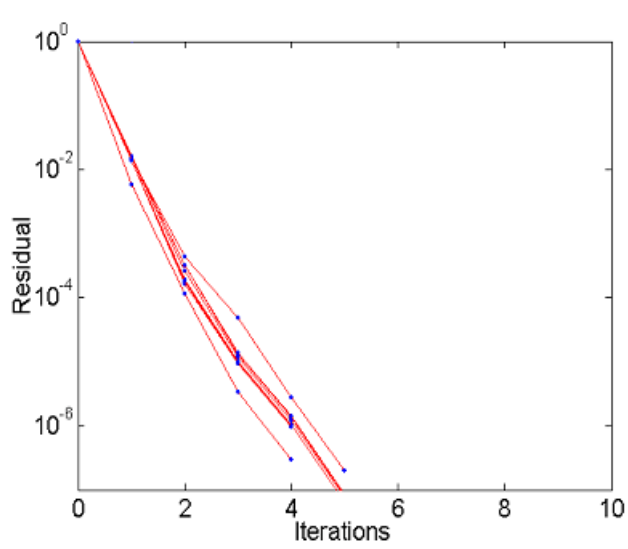

Fig. 2-13: Residual reductions for GMRES with the row-sum preconditioner

From Figs. 2-11 to 2-13, we see that the three preconditioners work very well. The average number of iterations for convergence per linear system is about five. Fig. 2-12 and Fig. 2-13 are almost identical.

The second test case is very similar to the first one, but the well roles are switched. The corner wells are producers with rate control, and the well in the middle is an injector with

BHP control. Figs. 2-14 to 2-16 show the performance of the three preconditioners for this case. We notice that the Schur complement preconditioner still works very well, because it has full information of the matrix. The A_{rr} preconditioner does not work well in this case; however, the performance of the row-sum preconditioner is quite good.

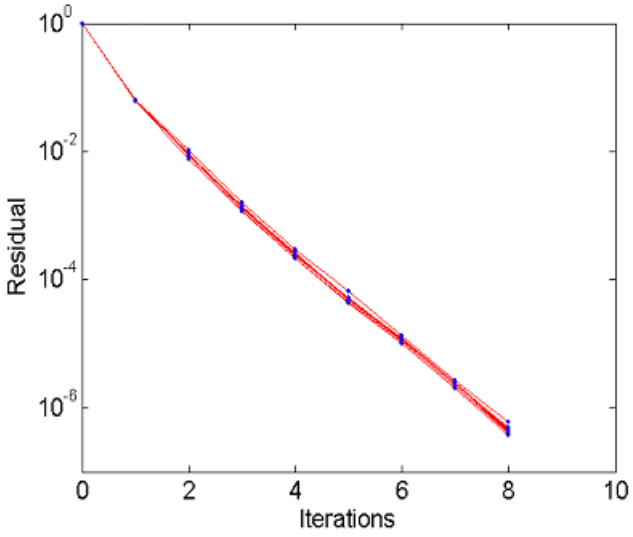

Fig. 2-14: Residual reductions for GMRES with Schur complement preconditioner

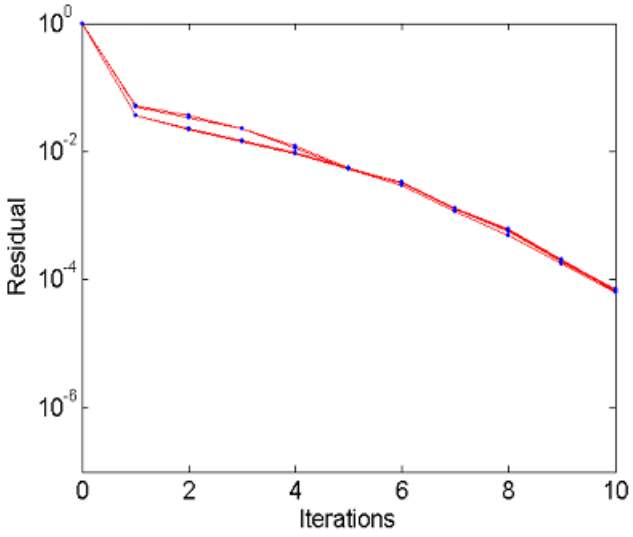

Fig. 2-15: Residual reductions for GMRES with Arr preconditioner

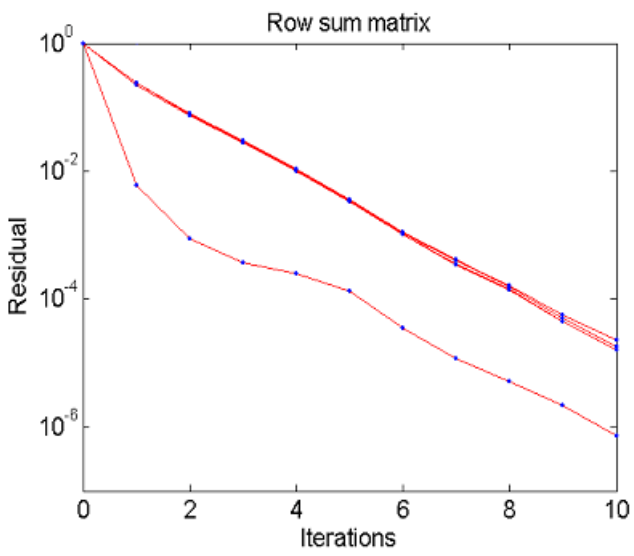

Fig. 2-16: Residual reductions for GMRES with row sum preconditioner

From above two cases, we see that the row-sum preconditioner works well. Overall, the preconditioners work much better in the first case than in the second case. This is evidence that rate controlled production wells may cause difficulty to the AMG solver.

2.3. FIM Formulation

Due to its unconditional stability, FIM is widely used in reservoir simulation. In large-scale problem, most of the run-time is spent on the linear solver. As a result, improving the solver performance leads to significant gains in the overall performance of the simulator. We are trying to find a set of optimal parameters for GPRS. In this section, all cases are performed with the FIM formulation.

2.3.1. Parameter Optimization.

First, we need to decide on the primary choice for the solver and preconditioner in the simulator. As mentioned in Section ***2.1.1***, GMRES solver is the best choice for our general-purpose simulation.

We study three FIM preconditioners: ILU, CPR with TrueIMPES, and CPR with QuasiIMPES. We illustrate the behavior of these preconditioners using a simple problem.

The test is based on six different model sizes for a homogeneous, 3D, oil-water system (see Table 2-3).

Table 2-3: Dimensions of the six cases

case	Dimension	Cells
1	15x55x2	1,650
2	30x110x2	6,600
3	30x110x10	33,000
4	30x110x20	66,000
5	30x110x30	99,000
6	30x110x40	132,000

There are two wells in the reservoir. The production well with BHP control is located at one corner; the injection well with rate control is in the middle of reservoir. Both wells penetrate all the layers. We report on the performance of the different preconditioners for each case. Fig. 2-17 shows the CPU time spent on the linear solver versus the model size.

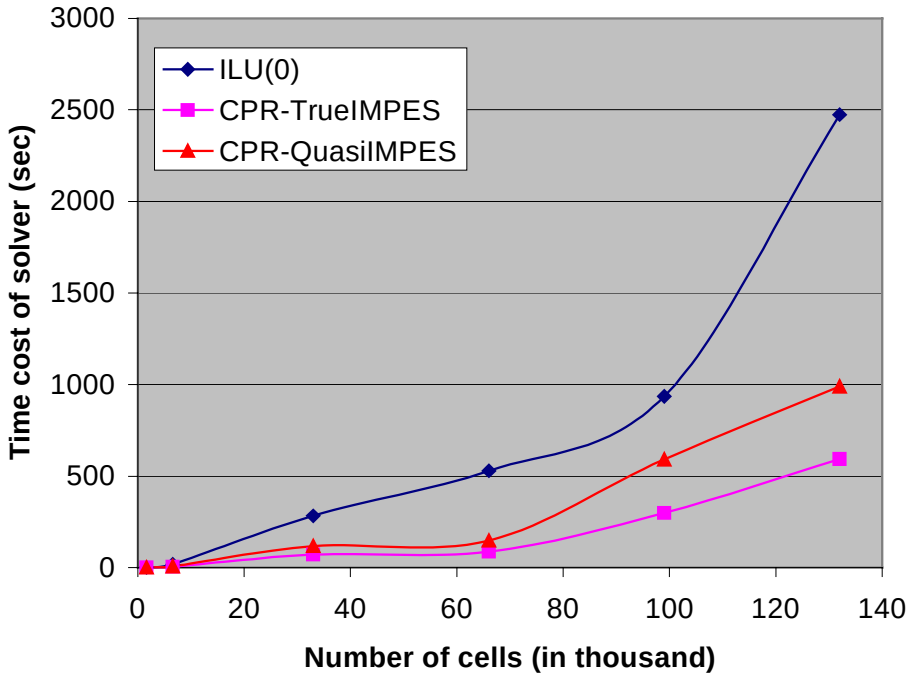

Fig. 2-17: Time cost of the linear solver with different preconditioners

From the Fig. 2-17, we notice that CPR with TrueIMPES has the best performance. The results of Fig 2-17 are typical across a wide range of problems. The performance advantage of CPR with TrueIMPES grows with problem size.

GMRES preconditioned by CPR with TrueIMPES is the best choice in GPRS. Now, we address the proper choice of the parameters of the preferred approach. Fig. 2-18 shows a flowchart for the implementation of the GMRES solver and the CPR preconditioner in GPRS.

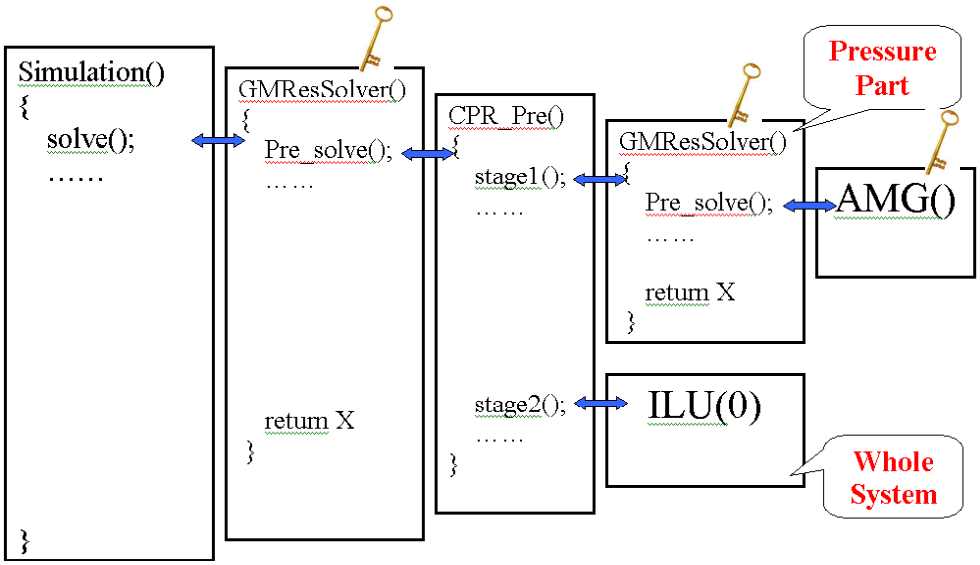

Fig. 2-18: Flowchart of simulation with GMRES solver and CPR preconditioner

We describe the procedure briefly. In the main loop, a linear system is built in each Newton iteration. The solver is GMRES and the preconditioner is CPR. There are two stages in the CPR method. A pressure matrix is formed and extracted from the whole system in the first stage. Again, GMRES is used to solve the pressure system, but for this phase, the preconditioner is AMG, which is perfectly suited for the pressure equation. In the second stage of CPR, ILU is applied to the entire matrix.

There are three keys in Fig. 2-18, which stand for the tolerances. These three tolerances are the most important parameters controlling the performance of the solver and preconditioner.

The first key is the tolerance of the overall linear solution. Generally, this value is defined by the user. The default for this value in GPRS is 10e-6. So we will not discuss it further in this report. The second key is the tolerance of the first stage solver, i.e. the pressure solver. We determined a reasonable value for this tolerance by performing a series of simulations. For each run, we keep all the settings exactly the same, but only change the tolerance of the first-stage solver. Fig. 2-19 shows the relationship between the CPU time spent on the linear solver and the tolerance for a typical 3D oil-water problem.

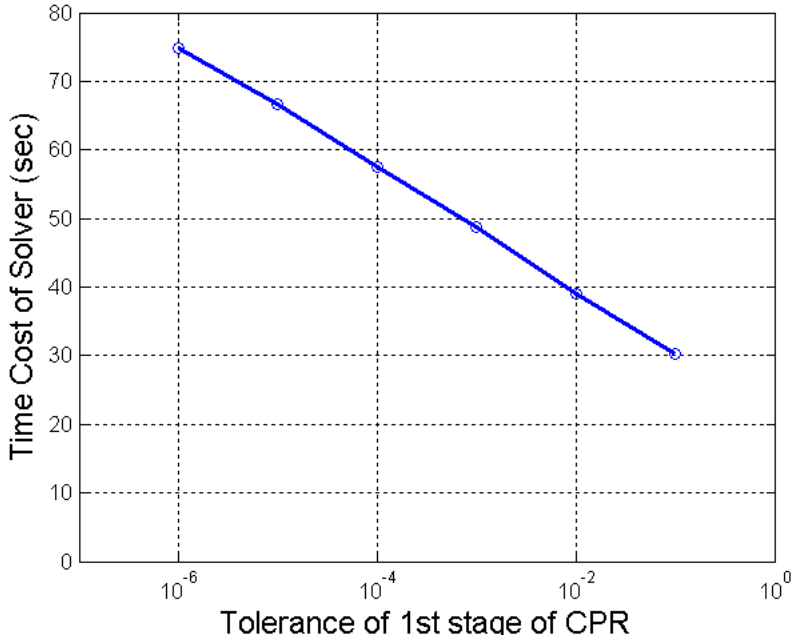

Fig. 2-19: Relation between solver time and tolerance of 1st stage of CPR

From Fig. 2-19, we see that the time-cost appears to have a linear relation with the tolerance. Generally, we found that the tolerance of the first stage solver can be loose. A reasonable choice is 10e-1.

The third key controls the accuracy of AMG, which is the preconditioner for the pressure matrix. Basically, there are two methods to achieve this. One is to use the l_2 norm as a measure of accuracy; the other is to use a fixed number of V-cycles (Ruge and Stuben, 1987). As the number of V-cycles increases, the solution becomes more accurate.

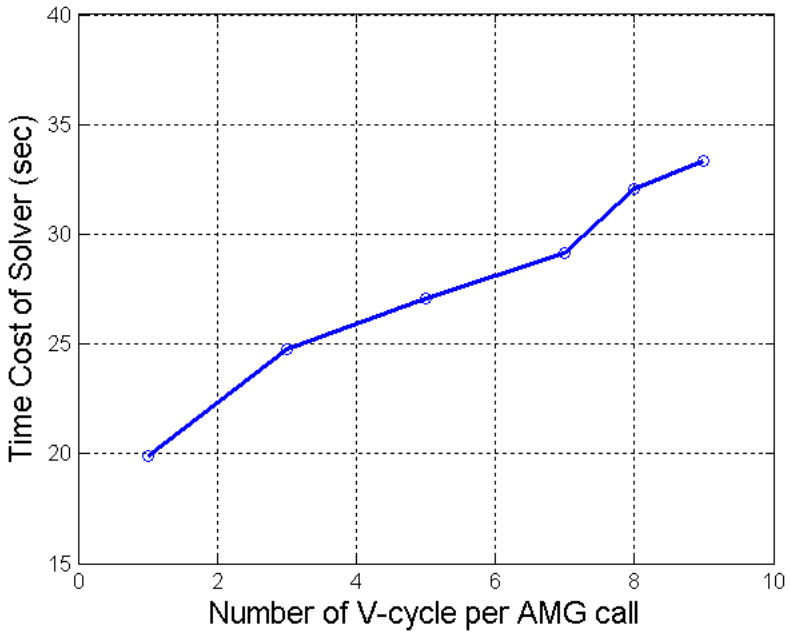

Fig. 2-20: Relation between solver time and number of V-cycle

To study the effect of the second tolerance on the overall linear solver time-cost, we performed several simulations where we only change the number of V-cycles per AMG call. Fig. 2-20 shows the CPU time spent on the linear solver part versus the number of V-cycles. We notice that the cost increases when a large fixed number of V-cycles is used. So in order to achieve the best performance, we need to fix the V-cycle per AMG call to one. This finding is consistent with our wider experience, and it is the recommended choice.

2.3.2. SAMG Solver

SAMG is the new generation of AMG, which was developed by Klaus Stuben (2003). It has much lower CPU and memory cost. In Table 2-4, item 1 and 2 show the performance

of the same problem with AMG and SAMG solvers respectively. We notice that SAMG only needs about half of the time and memory compared to that required by AMG. The test case has 141,900 cells. We expect the savings due to using SAMG to be even more substantial as the problem size grows. While savings in memory use in AMG are important, the overall memory consumption by the simulator in FIM mode is dominated by other factors, such as the storage of the FIM matrix as well as intermediate computations.

Compared to the original AMG code, the new package – SAMG - provides several parameters to help control the behavior of the various AMG components. For example, we reuse the hierarchy of coarse-grid choices and interpolation operators for all Newton iterations over a timestep, and we only update (recompute) the Galerkin operators in the hierarchy (Stuben, 2003). This approach is based on the assumption that changes in the Jacobian between Newton iterations over a timestep are relatively small. The third row of Table 2-4 shows the results. We can save 20% in the solver time for the pressure. In this particular test case, the average is about two Newton iterations per timestep. We can save even more time if we have more Newton iterations per timestep (i.e. make the strategy for increasing the timestep size more aggressive).

Table 2-4: Memory and time cost of AMG and SAMG solvers

Method	Memory Cost of solver (M)	Pressure solver Time (sec)
AMG	80	170.3
SAMG	45	84.98
SAMG*	45	70.94

2.3.3. Applications

SPE10 is described on a Cartesian grid, with 1,122,000 cells (60x220x85), with highly heterogeneous permeability. The dimensions of SPE10 are 1200x2200x170 (ft). “The top 70 ft (35 layers) represents the Tarbert formation, and the bottom 100 ft (50 layers) represents Upper Ness” (Christie and Blunt, 2001). SPE10 is widely used in research work, especially as a challenging test example.

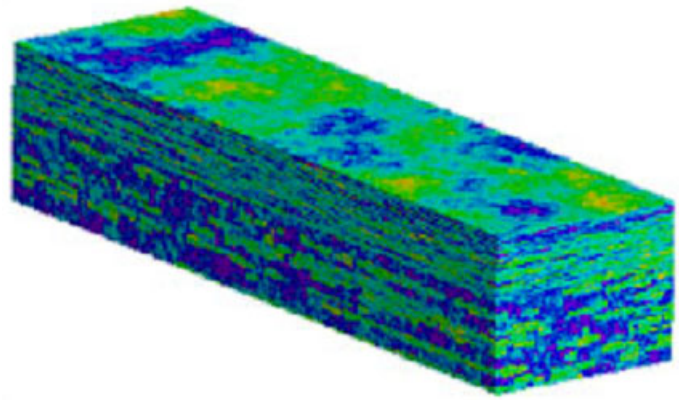

Fig. 2-21: Porosity map of SPE10 (Christie et al., 2001)

In order to validate the performance of the optimized solver and preconditioners, we ran the same cases on the Eclipse simulator (GeoQuest, Schlumberger, 2001) and GPRS in the same computing environment. The testing platform is an NTTS server, which has four 2.8G Hz CPUs, 4G memory. The operating system is Microsoft Windows 2000, Server package 4, which support up to 2G memory for one program. Since the current version of GPRS does not support parallel computing, we do not get much gain from the multi-CPU system.

Case 1: Upscaled (1/8) SPE10

In the first case, we upscaled the SPE10 case in 3 directions by a factor of two. The grid is 110 by 30 by 43 (141,900 cells). An injection well is located at the corner of the reservoir with water rate control; A production well is in the middle of reservoir using BHP control. Both wells fully penetrate through the 43 layers.

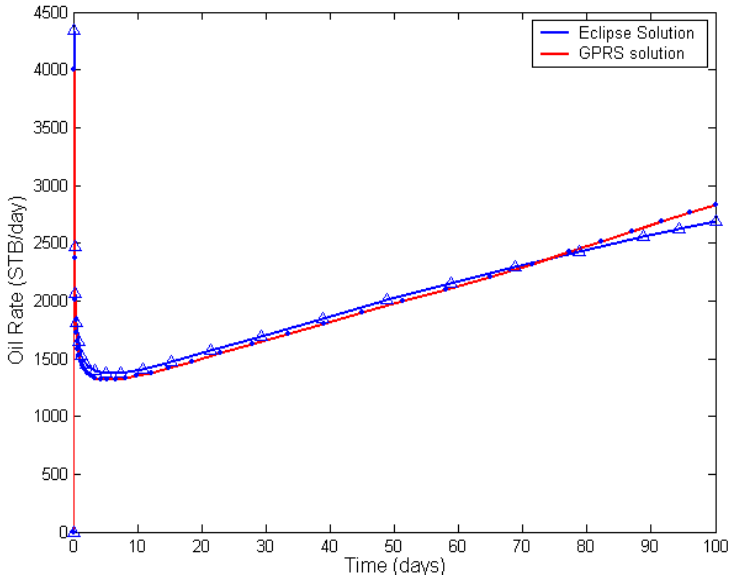

Fig. 2-22: Oil rate curves of Eclipse and GPRS in upscaled SPE10

Table 2-5: Performances of GPRS and Eclipse with upscaled SPE10 case

	GPRS	Eclipse
Pressure Solver Time (sec)	371 (39%)	-
Linear Solver Time (sec)	802 (84%)	-
Total Time (sec)	956 (100%)	5160
Time Step	19	23
Newton Iteration	102	88
Newton Iteration / Time Step	5.4	3.8
Linear Iteration	574	5515
Pressure Iteration	815	-

The oil production rate obtained from GPRS and the Eclipse simulator are shown in Fig. 2-22. The match between the two simulators is very good. The detailed performance of the two simulators is shown in Table 2-5. The time cost of GPRS is only about 20% of the Eclipse simulator's time cost. There are no significant differences in the number of timesteps and Newton iterations. However, the numbers of linear iteration for the two simulators are very different. The Eclipse simulator uses Orthomin preconditioned by nested factorization (GeoQuest, Schlumberger, 2001).

We can see that the linear solver in GPRS is quite efficient for this case. This is primarily due to the use of optimal parameters in our CPR approach, which employs AMG as a preconditioner for the first stage.

Case 2: Bottom 50 layers of SPE10

Due to memory constraints, we can only run simulations with up to 50 layers of the original SPE10 model. We selected the bottom 50 layers of SPE10 as our model, which is the most difficult part of SPE10. In this case, we did not get convergence from the Eclipse simulator.

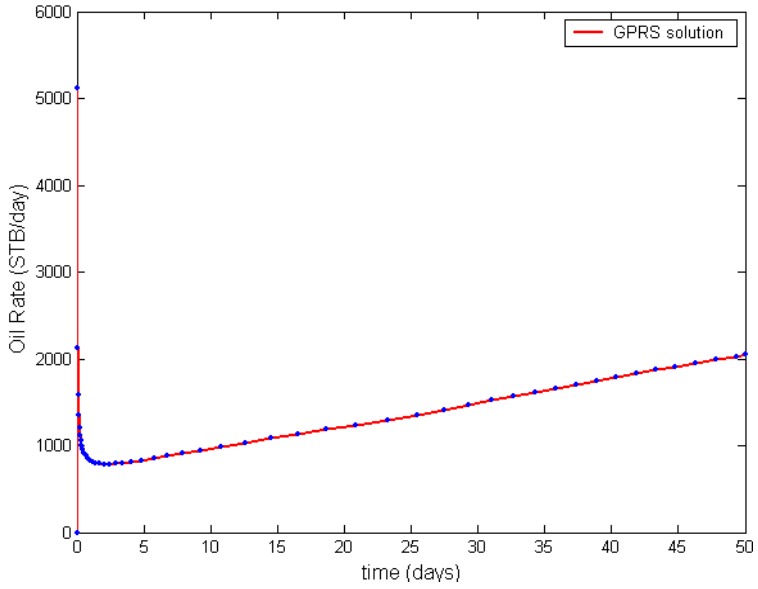

Fig. 2-23: Oil rate curve of GPRS in bottom 50 layers of SPE10

Table 2-6: Performances of GPRS and Eclipse with bottom 50 layers of SPE10

	GPRS
Pressure Solver Time (sec)	4160 (41%)
Linear Solver Time (sec)	8724 (86%)
Total Time (sec)	10156 (100%)
Time Step	50
Newton Iteration	195
Newton Iteration / Time Step	3.9
Linear Iteration	1342
Pressure Iteration	2076

We note here that default parameter settings were used for Eclipse, and no attempt was made to optimize the parameter choices. We only increased the maximum number of linear solver iterations because the simulator prompted us to do so. For GPRS, we used 1.e-6 as the tolerance for the linear solver.

GPRS is not as efficient in memory as the Eclipse simulator. For example, for **Case 2**, GPRS needs about 1.8 Giga bytes; while the Eclipse simulator needs only about 0.6 Giga bytes. In order to make the coding work clear and easy to understand, GPRS stores the same matrix in two different ways, which is one reason for the excessive memory use. GPRS needs to form a pressure matrix separately, as well. Beside the basic variables, such as saturations and porosity, GPRS stores all intermediate results, such as mobility, derivative of mobility with respect to pressure, to save computing time cost. These are some of the reasons for excessive memory use. Some effort has been invested to reduce the memory cost of GPRS, but we cannot get significant improvements without rewriting significant parts of low-level code implementations.

2.4. Summary and Future Work

In this chapter we focused on improving the performance of the linear solver in GPRS. We paid particular attention to the FIM systems with coupled wells. Because the first stage in CPR, which is our preferred method, deals with a pressure system, we studied the behavior of IMPES matrices. We report on the performance of three AMG-based preconditioners for pressure systems with coupled wells. We also integrated a new pressure solver, SAMG, into GPRS and investigated its properties. We reported a set of ‘optimal’ CPR parameters for the simulation of large-scale problems using GPRS.

This effort has resulted in significant improvements in the robustness and performance of the linear solver in GPRS. We demonstrate that the CPR approach, which is a two-stage preconditioning strategy, is quite effective when it is put together as follows:

First stage

- Construct a pressure system from the full coefficient matrix using the True-IMPES reduction scheme.
- If implicit wells are present, eliminate the well terms from the reservoir equations using the row-sum preconditioner.
- Use GMRES with Algebraic MultiGrid (AMG) preconditioning to solve the pressure system.
- Compute the residual of the full system, and pass it to the second stage.

Second stage

- Use a local preconditioner, such as ILU0, and apply it to the whole system.

Based on this investigation, there are several interesting areas of future work that can be pursued. Examples include:

- Improving the row-sum preconditioner.
- Testing problems with large numbers of wells with multiple completions.
- Evaluating the performance of various preconditioners for the second stage, e.g. diagonal preconditioner.
- Improving memory use in GPRS.
- Testing the CPR solver on AIM systems.
- Investigating extending the linear solver implementation to parallel platforms.
- Deriving efficient preconditioning methods for multi-segmented wells.

Nomenclature

A	=	Coefficient matrix of the linear system
A_p	=	IMPES pressure matrix
$A_p^{\ d}$	=	Diagonal term of the pressure matrix
$A_p^{\ o}$	=	Off-diagonal term of the pressure matrix
$A_p^{\ w}$	=	Well term of the pressure matrix
A_{pp}	=	Derivative of accumulation term in primary equation wrt. Pressure
A_{ps}	=	Derivative of accumulation term in primary equation wrt. Saturation
A_{sp}	=	Derivative of accumulation term in secondary equation wrt. Pressure
A_{ss}	=	Derivative of accumulation term in secondary equation wrt. Saturation
A_{rr}	=	Sub-matrix whose elements are the derivatives of reservoir equations wrt. reservoir variables
A_{rw}	=	Sub-matrix whose elements are the derivatives of reservoir equations wrt. well variables
A_{wr}	=	Sub-matrix whose elements are the derivatives of well equations wrt. reservoir variables
A_{ww}	=	Sub-matrix whose elements are for the derivatives of well equations wrt. well variables
$A_{rr}^{\ *}$	=	Schur complement
b_o	=	Reciprocals of the formation volume factor (FVF) of oil phase.
b_w	=	Reciprocals of the formation volume factor (FVF) of water phase
c_i	=	Tracer concentration in cell i
c_{in}	=	Upstream tracer concentration of two cells
C	=	Column operation matrix

C_o = Water phase compressibility

C_w = Oil phase compressibility

$\vec{e}$ = Vector with all ‘1’ elements

f = Flux

$f(u)$ = Flux of u.

f_w = Water flux

$F_{i \to l}$ = Molar flow rate of the carrier phase from cell i to neighboring cell l

F_{pp} = Derivative of flux term in primary equation wrt. Pressure

F_{ps} = Derivative of flux term in primary equation wrt. Saturation

F_{sp} = Derivative of flux term in primary equation wrt. Pressure

F_{ss} = Derivative of flux term in secondary equation wrt. Saturation

F_{pp}^{o} = Derivative of flux term in primary equation wrt. neighbor cell pressure

F_{ps}^{o} = Derivative of flux term in primary equation wrt. neighbor cell saturation

F_{sp}^{o} = Derivative of flux term in secondary equation wrt. neighbor cell pressure

F_{ss}^{o} = Derivative of flux term in secondary equation wrt. neighbor cell saturation

F_{pp}^{w} = Derivative of well flux term in primary equation wrt. neighbor cell pressure

F_{sp}^{w} = Derivative of well flux term in secondary equation wrt. neighbor cell pressure

G = Hydraulic head

k = Permeability of porous media

k_{ro} = Relative permeability of the oil phase

k_{rw} = Relative permeability of the water phase

m = Moles of the carrier phase

m_w = Mass of water per unit volume

M_i = Moles of carrier phase in cell i,

M_1 = The first stage preconditioner of a two-stage preconditioner

$M_{1,2}$ = Two-stage preconditioner

M_2 = The second stage preconditioner of a two-stage preconditioner

NP = Number of perforations of a well

N_p^w = New fill-in term in well equation

P = Pressure

P_o = Pressure of oil phase

P_w = Pressure of water phase

q_o = Specified oil rate

Q = Source term

Q_{M_i} = Molar production rate of the carrier phase from well completion in block i

r = Ratio of successive terms

$\vec{r}$ = Residual vector

R^w = Residual of the well equation

R_w = Residual of the water component equation

R_o = Residual of the oil component equation

S_o = Oil saturation

S_w = Water saturation

Δt = Timestep

T_o = Transmissibility of oil phase between cells

T_w = Transmissibility of water phase between cells

T_T^w = Total well transmissibility

T_o^w = Well transmissibility of oil phase

T_w^w = Well transmissibility of water phase

$\vec{u}$ = Solution vector

V_i = The bulk volume of cell i

W^T = Row operation matrix

WI = Well index

Acronyms

AIM = Adaptive implicit method

AMG = Algebraic multi-grid

BHP = Bottom hole pressure

CPR = Constrained pressure residual

FIM = Fully implicit method

FVF = Formation volume factor

GMRES = Generalized minimum residual

GPRS = General Purpose Research Simulator

ILU = Incomplete LU decomposition

IMPES = Implicit pressure and explicit saturations

IMPSAT = Implicit pressure and saturations

PVI = Pore Volume Injected

TVD = Total variance diminishing

Greek

Δ = Change

ρ_o = Density of oil phase

ρ_w = Density of water phase

μ_o = Viscosity of oil phase

μ_w = Viscosity of water phase

ϕ = Porosity

$\varphi(r)$ = Limiter function

Subscripts

p	=	Pressure
P	=	Pressure
o	=	Oil
s	=	Saturation
S	=	Saturation
w	=	Water

Superscripts

d	=	Diagonal
w	=	Well
o	=	Off-diagonal
n	=	Time level n
$n+1$	=	Time level $n+1$

References

Ali, E., Chatzichristos, C., Aurdal, T. and Muller, J. "Tracer Simulation to Improve the Reservoir Model in the Snorre Field". *SPE 64796*, SPE International Oil and Gas Conference and Exhibition in China held in Beijing, China, November 7–10, 2000

Almehaideb, R.A., Aziz, K. and Pedrosa, D.A.: "A Reservoir Wellbore Model for Multiphase Injection and Production". *SPE 17941*, proceedings of the SPE Middle East Oil Technical Conference and Exhibition, Manama, Bahrain, March 11-14, 1989

Anderson, E. LAPACK (Linear Algebra Package) this is a standard linear algebra package, which can be downloaded from www.netlib.org freely. 1999

Aziz, K, Durlofsky, L. J. and Gerritsen, M. *"Notes for Petroleum Reservoir Simulation"*. Stanford University, Winter 2003

Aziz, K. and Settari, A. *"Petroleum Reservoir Simulation"*. Applied Science Publishers, London**,** 1979

Behie, G. A. and Forsyth, P. A. "Incomplete Factorization Methods for Fully Implicit Simulation of Enhanced Oil Recovery". SIAM *J. Sci. Stat. Comp. Vol. 5, 543-561*, 1984

Behie, G. A. and Vinsome, P. K. W. "Block Iterative Method for Fully Implicit Reservoir Simulation". *Soc. of Pet. Eng. J.*, pages 658--668, Oct. 1982

Cao, H., Ph.D. dissertation "Development of Techniques for General Purpose Simulators". Stanford University, June 2002

Cao, H. and Aziz, K. "Performance of IMPSAT and IMPSAT-AIM Models in Compositional Simulation". *SPE 77720*, SPE Annual Technical Conference and Exhibition held in San Antonio, Texas, 29 September-2 October 2002

Christie, M. A. and Blunt, M. J. "Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques". *SPE 66599*, SPE Reservoir Simulation Symposium, TX, February 11-14, 2001

Coats, K.H.: "A Note on IMPES and Some IMPES Based Simulation Models", *SPE J. (SPE 65092)*, September, 2000

Demmel, W. J. *"Applied Numerical Linear Algebra"*. Society for Industrial and Applied Mathematics, Philadelphia, 1997

Forsyth, P.A. and Sammon, P.H. "Practical Considerations for Adaptive Implicit Methods in Reservoir Simulation", *J. Comp. Phys.*, Vol. 62, pp265-281, 1986

Fu, D. X. *"Numerical Simulation of Fluid Dynamics" (in Chinese)*. Beijing, 1993

Fussell, D. D. and Yanosik, J. L. "An Iterative Sequence for Phase-Equilibria Calculations Incorporating the Redlich-Kwong Equation of State" *SPEJ*, pp173-182, June 1978

GeoQuest, Schlumberger "Eclipse 200 Reference Manual – Multi-Segment Wells", 2000 Release, 2000

GeoQuest, Schlumberger "Eclipse Technical Description 2001A", 2001

Godunov, C. K. "Difference method for computing the discontinuity in fluid dynamics". Math Sbornik, 1959, 47(3):271

Harten A. "High resolution schemes for hyperbolic conservation laws". *J. Comput Phys*, 49:357~393, 1983

Landmark Graphics Corporation: "BlitzPak User's Guide", 1998, http://www.lgc.com

Lake, W. L. *"Enhanced Oil Recovery"*. Prentice-Hall, Inc, University of Texas, Austin, 1989

Lien, F. S. and Leschziner website: http://mecheng1.uwaterloo.ca/~fslien/tvd/ umist. html. 1994

Liu, R. X. and Shu, Q. W. *"Several new methods in Computational Fluid Dynamics" " (in Chinese)*. Hefei, 2003

Palagi, C. Ph.D. dissertation **"**Generation and Application of Voronoi Grid to Model Flow in Heterogeneous Reservoirs". Chapter 10, "Dual Timesteps to Model Tracer Flow", Stanford University, 1992

Pan, H. Q. "Test Plan and Test Case Specifications for Testing General Purpose Research Simulator (GPRS) Software". Personal communication, October, 2003

Rubin, B. and Blunt, M. J. "Higher-Order Implicit Flux Limiting Schemes for Black Oil Simulation". 11th SPE Symposium on Reservoir Simulation held in Anaheim, California, February 17-20, 1991

Ruge, J. W. and Stuben, K. "*Multigrid Methods –Chapter 4 Algebraic Multigrid"*. SIAM, 1987

Russell, T. F. "Stability Analysis and Switching Criteria for Adaptive Implicit". *SPE 18416* , SPE Symposium on Reservoir Simulation in Houston, TX, February 6-8, 1989

Saad, Y. and Schultz, M.: "GMRES: A Generalized Minimum Residual Algorithm for solving non-symmetric linear systems", SIAM J. Sci. Statist. Comput., Vol. 7, pp856-869, 1986

Stone, H. L. and Garder, A. O., Jr. "Analysis of Gas-Cap or Dissolved Gas Drive Reservoir". *SPE J.*, June, 1961 (AIME Trans. Vol. 222)

Stuben, K. "Algebraic Multigrid (AMG): Experiences and Comparisons", proceedings of the International Multigrid Conference, Copper Mountain, CO, April 6-8, 1983

Stuben, K. and Clees, T. "*User's Manual of SAMG"*, Release 21c August, 2003

Thomas, G. W. and Thurnau, D. H. "Reservoir Simulation Using Adaptive Implicit Method". *SPE J.* , October, 1983

Wallis, J. R. "Incomplete Gaussian Eliminate as Preconditioner for Generalized Conjugate Gradient Acceleration". *SPE 12265*, this paper was presented at the Reservoir Simulation Symposium held in San Francisco, CA, November 15-18, 1983

Wallis, J. R., Kendall, R. P., Little, T. E. “Constraint Residual Acceleration of Conjugate Residual Method”. *SPE 13536*, this paper was presented at the 1985 Reservoir Simulation Symposium held in Dallas, Texas, February 10-13, 1985.

Wattenbarger, R.C., Aziz, K. and Orr, Jr. F. M. "High-Throughput TVD-Based Simulation of Tracer Flow". SPE 29097, Proceedings, 13th Symposium for Reservoir Simulation, San Antonio, Texas, February 12-15, 1995. Published *SPE Journal*, Vol 2, No. 3, pp. 254-267 (September 1997)

Zhang, H.X. and Shen, M.Y. *“Computational Fluid Dynamics --- Fundamentals and Applications of Finite Difference Methods” (in Chinese)*. Beijing, 2003

Appendix A

A. The major updates for tracer in GPRS

- Add new *class Tracer* to GPRS (Files Tracer.h and Tracer.cpp are added into 'fluid' folder, and insert them into the project)
- Add head file "../fluid/Tracer.h" to file reservoir.h
- Add new variable *std::vector<Tracer*> mTracers* to *class Reservoir*
- Add new keyword "**TRACER_DATA**" to function *void Reservoir::readData(FileIO &fio)*
- Add a new member variable *bool TracerOn* to *class Reservoir* and initialize it in constructor
- Add codes for tracer in function *void Reservoir::initialize()*
- Add codes for tracer in function *void Reservoir::simulate()*
- Add codes for tracer in function *void Reservoir::report()*
- Add new member function *int getType()* into *class Well*
- Add new member function *double* getWPhaseMassRate()* into *class Well*
- Add new member variable *double* mWPhaseMassRate* into *class Well* (including declaration, memory allocation, deletion and evaluation)
- Add new member variable *double* mWPhaseMassRate* into *class WellCont*
- Bond two variables *WellCont::mWPhaseMassRate* with *Well::mWPhaseMassRate* in void function *void WellCont::initialize()*

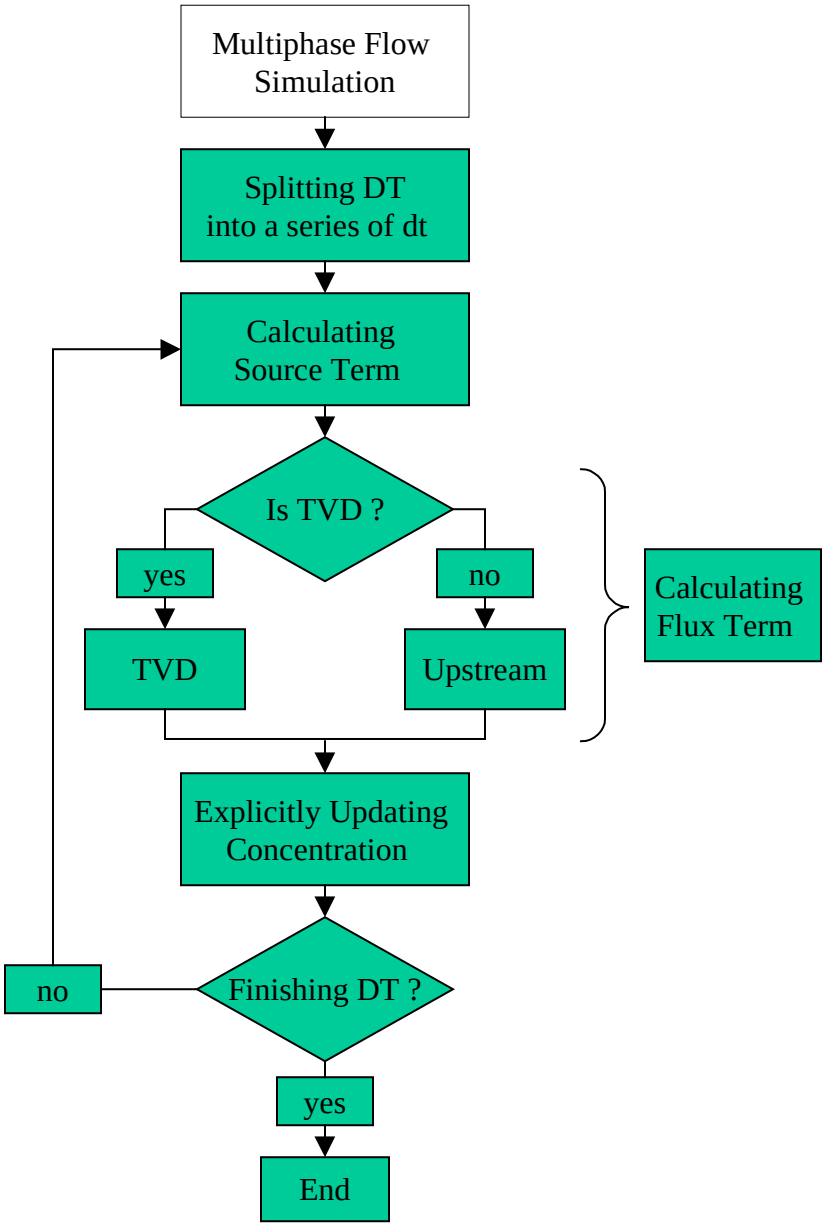

Fig. A-1: The flowchart of tracer flow simulation

Appendix B

B. The Characters of Tracer Option in GPRS

- The carrier phase is water
- Support TVD scheme and upstream scheme
- Support file IO
- Compatible with original files
- Use explicit scheme & duel time steps
- Consider capillary pressure and gravity
- Support all kinds of reservoirs
- Support structure and unstructured grid (no TVD option for unstructured grid)
- Support all kinds of wells
- Support multi-tracer
- Support multi-period tracer injection
- Support error protection
- Output of tracer is the concentration of cells and produced water

Appendix C

C. Sample Input & Output files of Tracer Option

To active the tracer option in GPRS, you need to add several lines in the input file of GPRS. This block should be inserted in to the reservoir part of the input file, i.e. after the keyword "RESERVOIR_NAME", and before the keyword "END_RESERVOIR". We already have other data blocks between them, such as "grid data", "fluid data", "rock data" and "equilibrium data". The order of these blocks is arbitrary. If we have more tracers, just add more tracer data blocks. The input files without tracer data block can still work well through GPRS. So it's completely compatible with old input files.

Input File

Tracer data block begins with a keyword "`TRACER_DATA`", and ends with a keyword "`END`". We have a sample as follows. The items will be introduced in turn.

Example

```
TRACER_DATA
#  Tracer name
   tracer1
#  TVD or not, keyword TVD_ON or TVD_OFF
   TVD_ON
#  Associate well, begin with 0
   1
#  injection periods
   1
#  begin time1, end time1, begin time2, end time2
   0  6000
```

*** The line begins with "#" is a comment line.**

Tracer Name:

The first line gives out the tracer name. It's the ID of a tracer. If you have more than one tracer in your simulation, they should have different names. The tracer name will also be the output file name of the tracer, with a extension ".out".

TVD option:

We have two keywords for this option--"TVD_ON" and "TVD_OFF". Only one can be applied one time. If "TVD_ON" is applied, tracer simulation will be performed with TVD scheme with Superbee limiter function, otherwise, upstream scheme will be used.

`Associated Well`

This integer parameter indicates the well in which the tracer will be injected. **You should guarantee that the associated well is a water injector.** Please see the well part of input files to find the value. And it starts from 0. One tracer can only associate with one well.

Injection periods

This integer parameter indicates the how many tracer injection periods in the simulation process. Generally, it's 1. It's larger than 1, if we have slug tracer injection.

Beginning & ending time

In this line, 2×*Injection periods* integer parameters will be specified, in the order of beginning time1, ending time1, beginning time2, ending time2 ……

Output File

We have one output file per tracer. The name of output file is the ID of the tracer, with a extension of ".out". Output file(s) will be created at the beginning of simulation, and refresh at the end of each time step. A typical output file is provided as follows. It's generated for a tracer named "tracer1". We will introduce the components in turn.

Example: tracer1.out

```
Time: 53 days

#Wells Head Tracer Concentration
0
-1

#Tracer Concentration in Each Cell
0.999966
0.999934
0.999841
0.999598
0.99893
0.996775
0.98779
0.942259
0.698739
0.163503
0
0
```

Time

In this line, the current time will be given out.

Wells Head Tracer Concentration

In this part, the concentration in produced water of each well is given out in the order of wells (see input files well part). The wells with -1 value are the injectors. All the injectors have -1 as their concentration**. You can not tell which injection well is the tracer-injecting well from this part.**

Tracer Concentration in Each Cell

This is the last part of output file. The tracer concentration of each cell is given out in the order of cells. Find the information about the grid in the grid part of input files

Appendix D

D. Data Files of Test Cases

Date File 1

The data file for the case to demonstrate the properties of pressure matrix, which is discussed in Section ***2.2.1***.

```
# --- Field Data ----------------------------------
Field_Name: Pressure matrix_properties_demonstration_case

# ---------- Reservoirs Data ----------#

NumOfReservoirs                 1

RESERVOIR_NAME   RES1

GRID_DATA
GRIDSIZE        2       2       1

DX
        40
DY
        40
DZ
        20

PERMX
        1
PERMY
        1
PERMZ
        1
PORO
        0.3
TOPS
        12000
TEMP
        500
END

FLUID_DATA
FLUID_TYPE BLACK_OIL
NPHASES         2
```

```
NCOMPONENTS          2

# Phase data ----------
PHASE_NAME OIL
STANDARD_DENS        53.0
NUM_OF_TABLE_ENTRIES      3
#          P         BO        VISC      RGO
#          PSI       rb/stb    CP        SCF/STB
           300       1.05      2.85      0
           800       1.02      2.99      0
           8000      1.01      3.00      0

PHASE_NAME WATER
STANDARD_DENS 64.0
NUM_OF_TABLE_ENTRIES 4
#          P         BW        VISC      RGO
#          PSI       rb/stb    CP        SCF/STB
           300       1.0       0.3       0
           800       1.0       0.3       0
           6000      1.0       0.3       0
           8000      1.0       0.3       0

#          300       1.027     0.3       0
#          800       1.026     0.3       0
#          6000      1.010     0.3       0
#          8000      1.004     0.3       0

# --- component data ----------
END

PHASE_COMP_RELATION_DATA
# --- component phase relation, nPhases X nComps ---
#comp:  oil       water
           1         0         #oil
           0         1         #water
END

ROCKFLUID_DATA

#OILWATERPERM 1
OW
NUM_OF_TABLE_ENTRIES      14
#Table     SW        KRW       KROW      PCOW
           0.200     0.0000    1.0000    0
           0.250     0.0069    0.8403    0
           0.300     0.0278    0.6944    0
           0.350     0.0625    0.5625    0
           0.400     0.1111    0.4444    0
           0.450     0.1736    0.3403    0
           0.500     0.2500    0.2500    0
           0.550     0.3403    0.1736    0
           0.600     0.4444    0.1111    0
           0.650     0.5625    0.0625    0
           0.700     0.6944    0.0278    0
           0.750     0.8403    0.0069    0
```

```
          0.800     1.0000   0.0000   0
          1.000     1.0000   0.0000   0
END

ROCK_DATA
#         COMP.               REF. PRES
          3.E-16              14.7
END

EQUILIBRIUM_DATA
#         Swi                 Sor                 Sgr
          0.2                 0.2                 0.0

#         pres      @depth    WOC                 GOC
          6000      12000               15000     2370
END
END_RESERVOIR

END_RESERVOIRS


# ----------------- Wells Data ----------------------#
NumOfWells      1

# WELL No        1

#         WELL_NAME   GROUP          RES_NAME      TYPE   STATUS
          P1          PRODW          RES1          P      OPEN
END

# --- completion (COMPDAT)---------------- # 4633 for 110 30 25 system
number_of_connections 2
#LOC(i,j,k)        WI
          0         20
          2         20
END

# --- control (WCONPROD/WCONINJE)--------
#         CTRL              Q         BHP               std_den Ncomp%comp1 %comp2
#         ORATE             1         1000.0            53.0
#         WRATE             1         100000.0          64.0    2         0         1
END

END_WELL

END_WELLS

# ---------------- Control Data -------------------#

# Timestep boundaries
# TSINIT          TSMAX             TSMIN
          5         10        5
# Total Simulation Time (Days)
10
```

```
# Newton iterations control
# minNewtonIter  maxNewtonIter    fixedNewtonIter
        1               50              70
# Timestep increase control
#       dpress  dsat    dx      w(tuning)
        2000    0.9     0.02    0.3

# Newton iteration convergence control
# MB_Error      PE_Error        P_Change        S_Change        MF_Change       Well
        0.1     0.02            1               1               1               0.001

# Formulation (variables and implicit levels) control
#       varType(AorB)           nbImpLevel    %AIM (if nbImpLevel>1)
        1                       1
# for each Implicit Level (1 to nbImpLevel):
# nImpVars      transOption     CFL
        1       1               1.0

# Linear solver control
# LinearSolverChoice    Preconditioner  tol     maxLinearIter
        4               6               1E-6    100             20
# --- debug information ---
#Flag   NumOfTimeSteps
0       10
```

Date File 2

The data file for the upscaled SPE 10 case (1/8), which is discussed in Section ***2.3.3.***

```
# --- Field Data ----------------------------------
Field_Name: Upscaled_SPE10_case

# ---------- Reservoirs Data ----------#
NumOfReservoirs                 1

RESERVOIR_NAME   RES1

GRIDSIZE        110     30      43
DX
        20
DY
        40
DZ
        4

INCLUDE         permx.inc       # do not provide here
INCLUDE         permy.inc
INCLUDE         permz.inc

PORO
        0.3
TOPS
        12000
TEMP
        500
END

FLUID_DATA
FLUID_TYPE BLACK_OIL
NPHASES                 2
NCOMPONENTS             2

# Phase data ----------
PHASE_NAME OIL
STANDARD_DENS       53.0
NUM_OF_TABLE_ENTRIES    3
#       P       BO      VISC    RGO
#       PSI     rb/stb  CP      SCF/STB
        300     1.05    2.85    0
        800     1.02    2.99    0
        8000    1.01    3.00    0


PHASE_NAME WATER
STANDARD_DENS 64.0
NUM_OF_TABLE_ENTRIES 4
#       P       BW      VISC    RGO
```

```
#       PSI     rb/stb  CP      SCF/STB
        300     1.027   0.3     0
        800     1.026   0.3     0
        6000    1.010   0.3     0
        8000    1.004   0.3     0

# --- component data ----------
END

PHASE_COMP_RELATION_DATA
# --- component phase relation, nPhases X nComps ---
#comp:  oil     water
        1       0       #oil
        0       1       #water
END

ROCKFLUID_DATA

#OILWATERPERM 1
OW
NUM_OF_TABLE_ENTRIES    14
#Table  SW      KRW     KROW    PCOW
        0.200   0.0000  1.0000  0
        0.250   0.0069  0.8403  0
        0.300   0.0278  0.6944  0
        0.350   0.0625  0.5625  0
        0.400   0.1111  0.4444  0
        0.450   0.1736  0.3403  0
        0.500   0.2500  0.2500  0
        0.550   0.3403  0.1736  0
        0.600   0.4444  0.1111  0
        0.650   0.5625  0.0625  0
        0.700   0.6944  0.0278  0
        0.750   0.8403  0.0069  0
        0.800   1.0000  0.0000  0
        1.000   1.0000  0.0000  0
END

ROCK_DATA
#       COMP.           REF. PRES
        3.E-6           14.7
END

EQUILIBRIUM_DATA
#       Swi             Sor             Sgr
        0.2             0.2             0.0

#       pres    @depth  WOC             GOC
        6000    12000           15000   2370
END
END_RESERVOIR

END_RESERVOIRS
```

```
#####################################################
# ----------------- Wells Data ---------------------#
NumOfWells         2

# WELL No. 1

# --- definition (WELSPECS) -------------
#  WELL_NAME     GROUP   RES_NAME TYPE    STATUS
   I1            INJW    RES1             I           OPEN
END

# --- completion (COMPDAT)---------------
number_of_connections    43
#LOC(i,j,k)     WI
  1594    4633.03            #(55,15) 1-based 14*110+54=1594
  4894    4633.03
  8194    4633.03
 11494    4633.03
 14794    4633.03
 18094    4633.03
 21394    4633.03
 24694    4633.03
 27994    4633.03
 31294    4633.03            #10th
 34594    4633.03
 37894    4633.03
 41194    4633.03
 44494    4633.03
 47794    4633.03
 51094    4633.03
 54394    4633.03
 57694    4633.03
 60994    4633.03
 64294    4633.03            #20th
 67594    4633.03
 70894    4633.03
 74194    4633.03
 77494    4633.03
 80794    4633.03
 84094    4633.03
 87394    4633.03
 90694    4633.03
 93994    4633.03
 97294    4633.03            #30th
100594    4633.03
103894    4633.03
107194    4633.03
110494    4633.03
113794    4633.03
117094    4633.03
120394    4633.03
123694    4633.03
126994    4633.03
130294    4633.03            #40th
133594    4633.03
```

```
136894  4633.03
140194  4633.03

END
# --- control (WCONPROD/WCONINJE)--------
#         CTRL            Q       BHP             std_den Ncomp%comp1 %comp2
          WRATE           5000    10000.0         64.0    2       0       1
END
END_WELL


#####################################################
# WELL No. 2
#####################################################
# --- definition (WELSPECS) -------------
#         WELL_NAME   GROUP           RES_NAME      TYPE   STATUS
          P1          PROD            RES1          P      OPEN
END

# --- completion (COMPDAT)----------------
number_of_connections 43
#LOC(i,j,k)     WI
0         4633.03
3300      4633.03
6600      4633.03
9900      4633.03
13200     4633.03
16500     4633.03
19800     4633.03
23100     4633.03
26400     4633.03
29700     4633.03         #10th
33000     4633.03
36300     4633.03
39600     4633.03
42900     4633.03
46200     4633.03
49500     4633.03
52800     4633.03
56100     4633.03
59400     4633.03
62700     4633.03         #20th
66000     4633.03
69300     4633.03
72600     4633.03
75900     4633.03
79200     4633.03
82500     4633.03
85800     4633.03
89100     4633.03
92400     4633.03
95700     4633.03         #30th
99000     4633.03
102300    4633.03
105600    4633.03
```

```
108900  4633.03
112200  4633.03
115500  4633.03
118800  4633.03
122100  4633.03
125400  4633.03
128700  4633.03         #40th
132000  4633.03
135300  4633.03
138600  4633.03

END

# --- control (WCONPROD/WCONINJE)--------
#       CTRL            BHP
        BHP             4000
END
END_WELL
END_WELLS

#####################################################
# ---------------- Control Data -------------------#
#####################################################
# Timestep boundaries
# TSINIT        TSMAX           TSMIN
0.1             10              0.1
# Total Simulation Time (Days)
100

# Newton iterations control
# minNewtonIter maxNewtonIter   fixedNewtonIter
        1               50              70
# Timestep increase control
#       dpress  dsat    dx      w(tuning)
        1000    0.8     0.02    0.5

# Newton iteration convergence control
# MB_Error      PE_Error        P_Change        S_Change        MF_Change       Well
        0.1     0.02            1               1               1               0.001

# Formulation (variables and implicit levels) control
#       varType(AorB)           nbImpLevel  %AIM (if nbImpLevel>1)
        1                       1
# for each Implicit Level (1 to nbImpLevel):
# nImpVars      transOption     CFL
        2       4               1.0

# Linear solver control
# LinearSolverChoice    Preconditioner  tol     maxLinearIter
        4               7               1E-6    50              20
# --- debug information ---
#Flag   NumOfTimeSteps
0       10
```