SPE-193218-MS

Advances in Parallel Reservoir Simulation Technology, Using Multi-Threaded Algorithms

Sina Mohajeri, Reza Eslahi, Maryam Bakhtiari, Mostafa Zeinali, Hamed Rajabi, Ali Alizadeh, Ebrahim Sharifi, and Emad Mortezazadeh, Engineering Support & Technology Development Company

This paper was prepared for presentation at the Abu Dhabi International Petroleum Exhibition & Conference held in Abu Dhabi, UAE, 12-15 November 2018.

Abstract

The popularity of parallel reservoir simulation has increased in recent years due to availability of multi-core CPUs and the opportunity for run-time reduction, especially in complex and fractured reservoir models that include large amounts of data. However, the accuracy of classical domain decomposition techniques significantly decreases as the number of processing units increases. The scope of this work is mainly concentrated on development of multi-threaded algorithms to overcome the mentioned problem in parallel reservoir simulations. In this work, at first, a newly developed, parallel, three dimensional, fully implicit, three-phase black-oil reservoir simulator is introduced. For solving large scale linear systems produced in giant black-oil models, which are probable cases in real world reservoir models, BiCG-Stabilized linear solver, enhanced with various preconditioners including, Blocked Incomplete LU factorization, Constrained Pressure Residual using Algebraic Multi-Grid as its pressure smoother is constructed with a variety of parallelization techniques applied in each part. The presented multi-threaded algorithms can utilize any user-defined number of CPU cores. The algorithm has been evaluated by SPE10 with 1122000 grid blocks and a real data file containing 277875 grid blocks. The result indicates that increasing the number of CPU threads does not have any negative effect on the simulation performance in case of high memory bandwidth limits. For SPE10 data file, 1.70, 2.4, and 2.3 speedup ratios (in comparison with a single thread) has been observed for 2, 4, and 6 parallel threads, respectively. In the real case, the simulation speed is increased by a factor of 1.49, 1.74, and 1.75 for 2, 4, and 6 parallel threads, respectively. Using multi-threaded algorithms as a novel approach in parallel simulations, the solution configuration does not change from the single core case and follows the same exact path. This however is not the case for other simulators that use domain-decomposition techniques. These techniques insert numerical errors in the solution of linear systems, which lead to convergence problems in the whole system.

Keywords: Parallel Reservoir Simulation, Multi-Threaded Algorithms, Fractured Reservoirs, Shared Memory Systems, Linear Systems

Introduction

A lot of computational power and time is necessary for simulating highly heterogeneous, fractured reservoirs with complex geometry; the efficiency of these computations is a major subject especially in large-scale heterogeneous and fractured reservoirs. Today, the popularity of parallel reservoir simulation has increased due to availability of multi-core CPUs and the opportunity for run-time reduction [1, 2]. However, the accuracy of classical domain decomposition techniques significantly decreases with the increase in number of processing units.

Chien Wasserman et al. [3] is the earliest work on parallel reservoir simulators. They developed a general-purpose reservoir simulator in a multiple vector processor machine. They used micro tasking which creates several independent tasks in a shared memory environment. Killough and Wheeler [4] studied domain decomposition in algorithms for iterative solution of linear systems of equations and performed tests on multitasking and micro tasking based, shared memory parallel systems.

A newly developed, parallel, three dimensional, fully implicit, three-phase [5, 6] black-oil reservoir simulator is used in this work. It has been fully developed by the authors, which utilizes a multi-threading algorithm that inserts no approximation in the solution and therefore is fully stable for large and complex models. This algorithm can utilize all or any user-defined number of CPU cores efficiently. In parallel simulations, the solution does not change from the single core case and follows the exact same path. This is not the case for other simulators, that use domain-decomposition techniques, which change the solution of the original system and therefore decrease the accuracy of the solution.

For solving large scale linear systems produced in giant models, bi-conjugate gradient stabilized (Bi-CGSTAB) linear solver [7], enhanced with various preconditioners including, Blocked Incomplete LU factorization [8], Constrained Pressure Residual using Algebraic Multi-Grid (CPR-AMG) [9, 10] as its pressure smoother is constructed with a variety of parallelization techniques applied in each part.

Open Multi-Processing (OpenMP) is an application program interface (API) used to develop parallel programs in shared memory systems. It provides several compiler directives, a runtime library, and environment variables [11]. OpenMP uses a multithreading method called the fork-join model in which the program initially starts as a single process, whenever a portion of the program can be executed in parallel the process is "forked" into several sub-processes, and once the parallel part is done all processes "join" back into a single process [12]. This approach to parallel programming is very convenient, especially when there is already a base code that runs serially. This is because there is no need to dramatically change the code to parallelize it.

The scope of this work is mainly concentrated on the development of multi-threaded algorithms to overcome the mentioned problem in parallel reservoir simulations.

Description of Linear Solver

The developed reservoir simulator has the BiCG-Stab standard iterative linear solver which is powered by several pre-conditioners including CPR-AMG-ILU0, ILU0, CPR-AMG-ILU1, and ILU1. The solver and all pre-conditioners have been fully developed in ESTD[1] and are customized for reservoir simulation needs. Among them, the CPR method is the cutting edge pre-conditioner in numerical science, which is based on the AMG technique. The AMG technique along with the CPR method to solve the pressure equation is implemented in the powerful numerical solver of the mentioned simulator. Note that using the CPR method without the AMG technique will present no industrial benefits.

The AMG technology is one of the most recent strategies for solving linear systems generated in elliptic partial differential equations that can produce suitable solutions for complicated problems. Our simulator has its own customized AMG based linear solver, which can solve models that are simply not solvable by other well-known commercial simulators. The developed simulator usually has large time steps with more realistic results especially in case of fractured reservoirs. It can also solve such models with less time step cut back and has far less non-linear and linear convergency problems.

Algorithm

For a black oil model - in which the saturation related properties are strongly nonlinear and the pressure related properties are weakly nonlinear - the following linear system, which is produced in each nonlinear iteration, must be solved.

$$Ax = b \qquad \text{(Eq.1)}$$

Where A in an $n \times n$ non-singular matrix, x is the desired solution, and b is a residual. In our simulator, a Krylov subspace solver, namely, BiCG-Stab is used. To speedup this solver, the following equivalent system is solved:

$$M^{-1}Ax = M^{-1}b \qquad \text{(Eq.2)}$$

Where M is called a preconditioner.

Due to the elliptic or parabolic nature of the pressure equations in black oil model, the AMG method is much more efficient than the ILU methods. Assume we have a model including n grid blocks with global index of 1 through n. For each model grid block i ($1 \leq i \leq n$), three primary variables of oil pressure $p_{o,i}$, and saturations in $\vec{x}_i$ are present. Depending on the model phase status, $\vec{x}_i$ can contain one or two variables. Assuming $\vec{w}$ represents well unknown variables (equal to number of wells in the model), we can define the pressure vector as equation 3:

$$p = \begin{pmatrix} p_{o,1} \\ p_{o,2} \\ \cdots \\ p_{o,n} \end{pmatrix} \qquad \text{(Eq.3)}$$

The global unknown vector, x, is defined as:

$$x = \begin{pmatrix} p_{o,1} \\ p_{o,2} \\ \cdots \\ p_{o,n} \\ \vec{x}_1 \\ \vec{x}_2 \\ \cdots \\ \vec{x}_n \\ \vec{w} \end{pmatrix} \qquad \text{(Eq.4)}$$

The global restriction operator is defined, then:

$$\Pi_r x = p \qquad \text{(Eq.5)}$$

This operator is not applied to well variables; it is only applied to fluid flow variables. The prolongation operator Π_p is defined as:

$$\Pi_p P = \begin{pmatrix} p_{o,1} \\ p_{o,2} \\ \cdots \\ p_{o,n} \\ \vec{0} \\ \vec{0} \\ \cdots \\ \vec{0} \\ \vec{0} \end{pmatrix} \qquad \text{(Eq.6)}$$

The $\Pi_p P$ has the same dimension as x.

The generated matrix A using Newton and Newton Raphson methods can be written in equation 7 as:

$$\vec{A} = \begin{pmatrix} A_{pp} & A_{ps} & A_{pw} \\ A_{sp} & A_{ss} & A_{sw} \\ A_{wp} & A_{ws} & A_{ww} \end{pmatrix} \qquad \text{(Eq.7)}$$

Where A_{pp}, A_{ss}, and A_{ww} are matrices corresponding pressure, saturation and well unknowns, respectively.

As mentioned before, the linear system in equation 1 is solved by Krylov subspace solvers. A preconditioner system, which has the same form as equation 1, has to be solved during the solution process:

$$Mx = f \qquad \text{(Eq.8)}$$

The CPR preconditioner is introduced here. Let r be the residual, defined by: r=f-Ax. Equation 9 describes the algorithm of our CPR preconditioner. According to our experience, the CPR preconditioner has a better convergency rate than other preconditioners in presence of an AMG solver.

$$\begin{aligned} &1{:}\, x = R(A)^{-1} f \\ &2{:}\, x = x + \Pi_p AMG\left(A_{pp}\right)^{-1} \Pi_r r \\ &3{:}\, x = x + R(A)^{-1} r \end{aligned} \qquad \text{(Eq.9)}$$

This is a three stage preconditioner. In the first stage the problem is solved initially, in the second stage, the pressure matrix is solved, and in the final stage, the entire problem is solved using the preconditioner and the remaining pressure residuals. In case of parallel computers, a subdomain problem is solved using ILU methods on each processor. Note that the size of the subdomain and overlap is the controlling factor of each subdomain problem.

For solving the matrix A_{pp}, our costumized AMG solver was used. Exhaustive tests and experiences show that only one iteration of V-cycle AMG method is enough for most problems.

Parallel Simulation Technology

Our developed simulator utilizes a multi-threading algorithm that imposes no approximations upon the solution and therefore is fully stable for large, heterogeneous, and complex models. This algorithm can utilize all or any user-defined number of CPU cores efficiently. In our parallel simulations, the solution does not change from the single core case and follows the exact same path. This, however, is not the case for other simulators that use domain-decomposition techniques. These techniques change the solution of the original system and therefore decrease the accuracy of the solution.

Result and discussion

Two models with the following descriptions were used to evaluate the implemented parallel algorithm. In order to tackle the complexity of a non-linear system with cross variable dependencies, we use a new and creative method to reduce the number of non-linear iterations. The developed simulator uses higher order terms in its Jacobian matrix in addition to the Newton's basic linear terms to account for cross variable dependencies. This scheme leads to a smaller number of non-linear iterations compared to basic Newton iterations. The number of Newton iterations in our non-linear solver is now comparable to other known commercial simulators, but, in our case, with larger time-steps.

First model description: This case is the tenth SPE comparative project fine model: A large structured fine grid with a reverse five-spot pattern water injection. It is a single porosity model with 1122000 grid blocks, highly heterogeneous channel, and floodplain sands. It is initially under-saturated and remains in two-phase state throughout the simulation. A water injection well is drilled at the center of the model and four oil production wells are present at the corners.

Second model description: The case is a real offshore oil field. Production from this field started in 1972. It is a dual porosity model with some single porosity layers. The total number of grid blocks in the model is 277875. This model is an under-saturated reservoir with a moderate aquifer as its boundary condition. During its history-matching period, this model has 8 active oil production wells.

First Model Results

Figure 1 displays elapsed time for four simulations with 1, 2, 4, and 6 parallel threads. This is the time taken to run the model from time 0 to the specified simulation time (days). It shows that increasing parallel threads decreases elapsed time. However, in this case, for six threads, the elapsed time is increased by a small amount. This is due to the memory bound nature of reservoir simulation calculations. The final elapsed time for 1, 2, 4, and 6 parallel threads are 5112, 3007, 2130, and 2222 second.

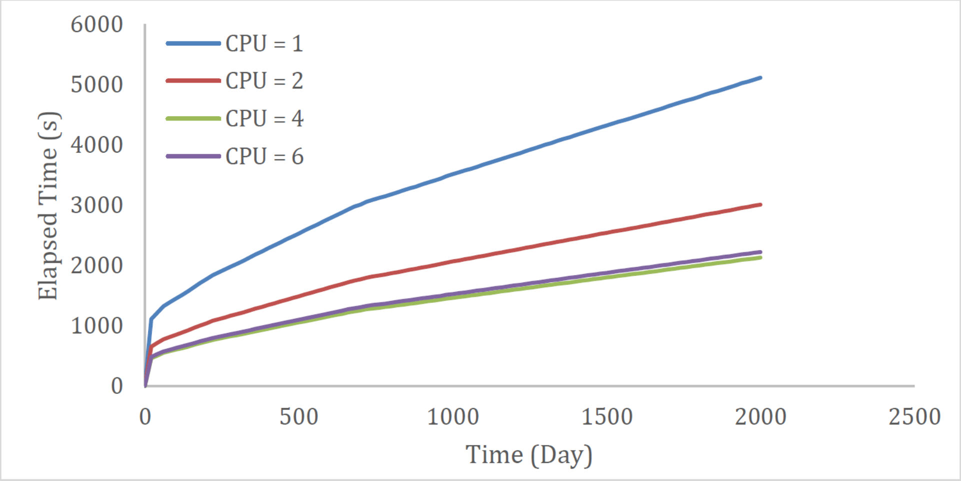

Figure 1—Elapsed Time for Various CPU Numbers (First Case)

Figure 2 shows the speedup ratio based on a single thread run versus number of parallel threads. For this case, 1.70, 2.4, and 2.3 speedup ratios were observed for 2, 4, and 6 parallel threads, respectively. As mentioned before, the observed speedup degradation from 4 to 6 thread, is due to the memory hierarchy bounded nature of reservoir simulation calculations. Beyond six threads, minimal speedups were observed due to memory bandwidth limits on the test machine.

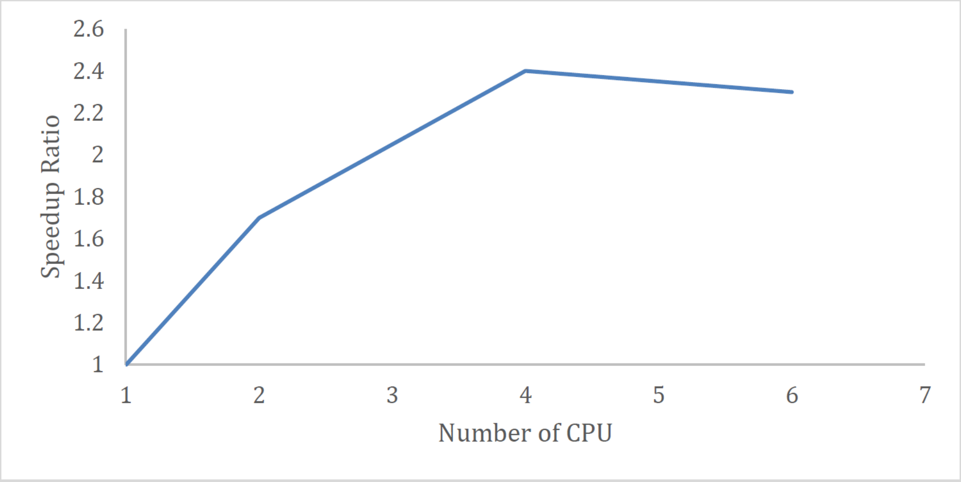

Figure 2—Speedup Ratio (First Case)

As mentioned in this article, in our parallel simulations algorithm, the solution does not change from the single core case and follows the exact same path and configuration. This approach prevents loss of solution accuracy due to approximation. The time step size and Newton iteration numbers has been displayed in Figure 3 as "run configuration". According to this figure, run configuration is the same for parallel and single core cases. It is clear that the report step size (time at which well rates are entered and results are printed for the model) is the limiting factor for time steps in the simulator. This limit is 20 days for this case.

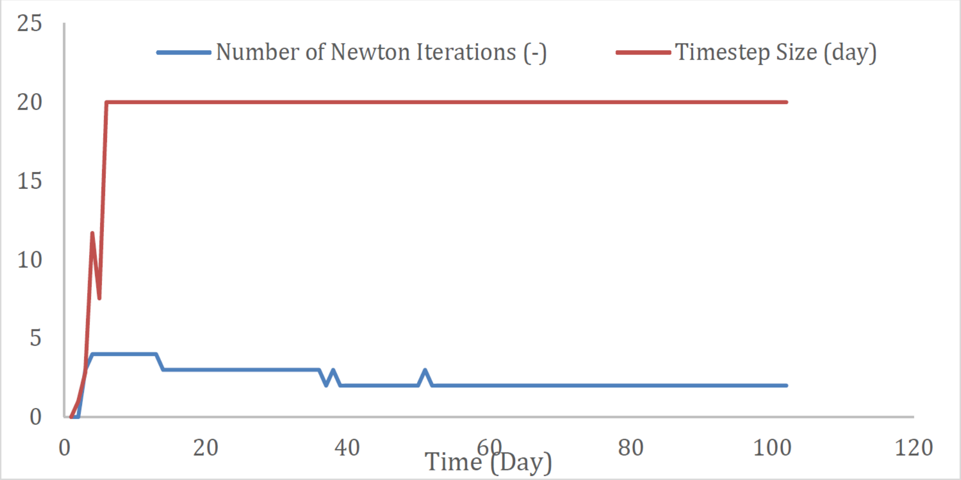

Figure 3—Run Configuration (First Case)

Second Model Results

Figure 4 displays the simulation elapsed time for 1, 2, 4, and 6 parallel threads for the second case. This is the time taken to run the model from time 0 to the specified simulation time (days). It shows that increasing parallel threads will decrease the elapsed time. The final elapsed time for 1, 2, 4, and 6 parallel threads are 22755, 15539, 13786, and 13003 second.

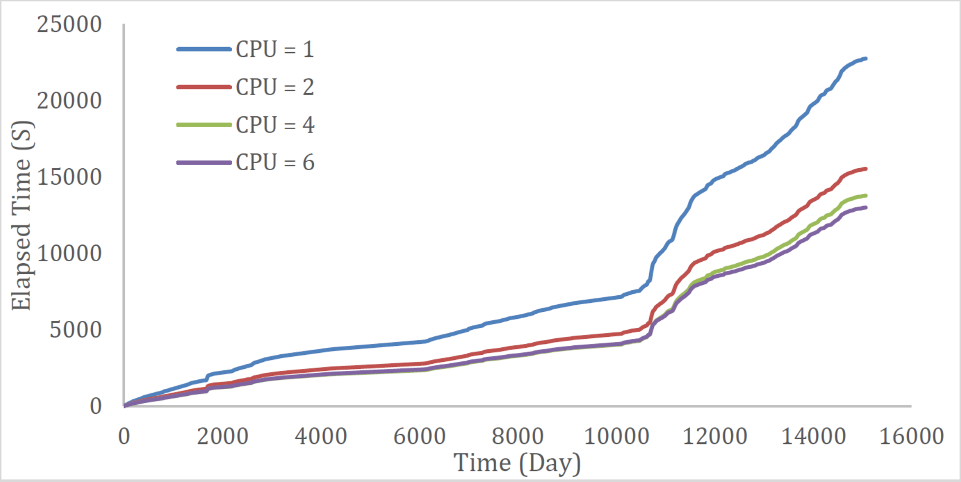

Figure 4—Elapsed Time for Various CPU Numbers (Second Case)

Figure 5 shows the speedup ratio based on the single thread run versus parallel threads number. For this case, 1.49, 1.74, and 1.75 speedup ratios were observed for 2, 4, and 6 parallel threads respectively. The observed speedup degradation, again is due to the memory hierarchy bounded nature of reservoir simulation calculations. Beyond six threads, minimal speedups were observed due to memory bandwidth limits on the test machine.

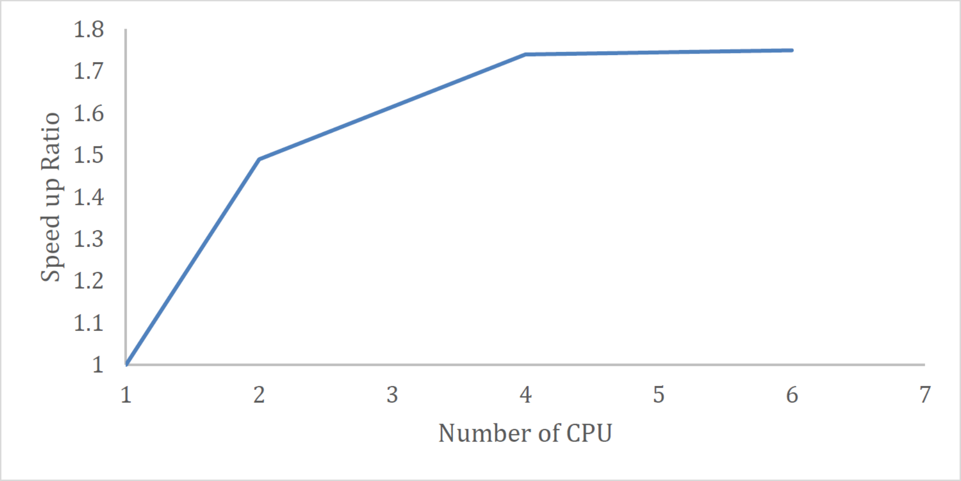

Figure 5—Speedup Ratio (Second Case)

In our parallel simulations algorithm, the solution does not change from the single core case and follows the exact same path and configuration; this approach prevents the loss of solution accuracy. The time step size and Newton iteration numbers has been displayed in Figure 6 as "run configuration". According to this figure, run configuration is the same for parallel and single core cases. The report step size (time at which well rates are entered and results are printed for the model) is the limiting factor for time steps in the simulator. This limit is 30 days for this case. From the accuracy point of view, it is evident that large time step sizes do not lead to higher time truncation error in our simulator.

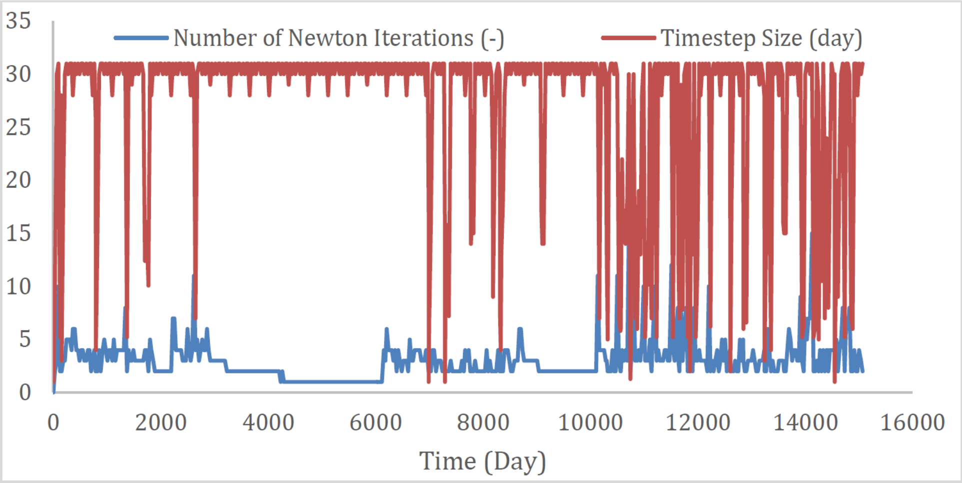

Figure 6—Run Configuration (Second Case)

Conclusion

A newly developed, parallel, three dimensional, fully implicit, three-phase black-oil reservoir simulator was used for solving large scale linear systems produced in giant black-oil models, using CPR-AMG linear solver. The presented multi-threaded algorithms can utilize any user-defined number of CPU cores. The algorithm has been evaluated by SPE10 with 1122000 grid blocks and a real data file containing 277875 grid blocks. Results indicate that increasing the number of CPU threads does not have any negative effect on the simulation performance in case of high memory bandwidth limits. The observed speedup degradation in high number of threads is due to memory hierarchy bounded nature of reservoir simulation calculations. According to our experiences, the CPR preconditioner has a better convergency rate than other preconditioners in presence of an AMG solver.

Nomenclature

A	An *n*×*n* non-singular matrix
A_{pp}	Matrices corresponding the pressure unknowns
A_{ss}	Matrices corresponding the saturation unknowns
A_{ww}	Matrices corresponding the well unknowns
A_{ps}	Combined matrices corresponding the pressure and saturation unknowns
A_{pw}	Combined matrices corresponding the pressure and well unknowns
A_{sp}	Combined matrices corresponding the saturation and pressure unknowns
A_{sw}	Combined matrices corresponding the saturation and well unknowns
A_{wp}	Combined matrices corresponding the well and pressure unknowns
A_{ws}	Combined matrices corresponding the well and saturation unknowns
AMG	Algebraic Multi-grid
API	application program interface
BiCG	bi-conjugated gradient
Bi-CGSTAB	bi-conjugated gradient stabilized
CPR	Constrained Pressure Residual
CPR-AMG	Constrained Pressure Residual using Algebraic Multi-grid
CPU	Central processing unit
ESTD	Engineering Support & Technology Development
M	Preconditioner
OpenMP	Open Multi-Processing

$p_{o,i}$	Oil pressure
r	residual
SPE10	Tenth SPE comparative model
$\vec{w}$	Well unknowns

Greek Letters

Π_p	Prolongation operator
Π_r	Global restriction operator

References

1. Liu, H., Wang, K., Chen, Z., Luo, J., Wu, S., & Wang, B. (2015, September 14). *Development of Parallel Reservoir Simulators on Distributed-memory Supercomputers*. Society of Petroleum Engineers. doi:10.2118/175573-MS
2. Zhlyuan, M., Fengjiang, J., Xiangming, X., Sunjiachang: Simulation of Black oil Reservoir on Distributed Memory Parallel Computers and Workstation Cluster. In: International meeting on petroleum engineering, Beijing, pp. 14–17 (1995)
3. Chien, M., Wasserman, M., Yardumian, H., Chung, E., Nguyen, T., and Larson, J., "The Use of Vectorization and Parallel Processing for Reservoir Simulation," *in: SPE-16025-MS, SPE: Society of Petroleum Engineers*, Jan. 1, 1987
4. Killough, J. E. and Wheeler, M. F., "Parallel Iterative Linear Equation Solvers: An Investigation of Domain Decomposition Algorithms for Reservoir Simulation," *in: SPE-16021-MS, SPE: Society of Petroleum Engineers*, Jan. 1, 1987.
5. Z. Chen, G. Huan, Y. Ma, *Computational methods for multiphase flows in porous media, Society for Industrial and Applied Mathematics (SIAM)*, Philadelphia, PA, 2006
6. K. Aziz and A. Settari, *Petroleum Reservoir Simulation*, London: Applied Science Publishers, 1979, p. 398.
7. Van der Vorst, H. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems." *SIAM J. Sci. Statist. Comput.* **13**, 631–644, 1992
8. Meijerink, J. A.; Vorst, Van Der; A, H. (1977). "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix". *Mathematics of Computation*. **31** (137): 148–162. doi:10.1090/S0025-5718-1977-0438681-4. ISSN 0025-5718.
9. T. Clees and L. Ganzer. An efficient algebraic multigrid solver strategy for adaptive implicit methods in oil-reservoir simulation. *SPE Journal*, **15**(03):670–681, 2010.
10. K. Stueben, T. Clees, H. Klie, B. Lu, and M.F. Wheeler. Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation. In SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (SPE), 2007.
11. Chapman, B. and LaGrone, J., "OpenMP," in: *Encyclopedia of Parallel Computing*, ed. By D. Padua, Boston, MA: Springer US, 2011, pp. 1365–1371, url: http://dx.doi.org/10.1007/978-0-387-09766-4
12. Blaise, B., "Introduction to Parallel Computing," in: *Lawrence Livermore National UCRLMI-133316 (EC3500 2015), url*: https://computing.llnl.gov/tutorials/parallel_comp/.