2016 IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security

Parallel Simulation of Full-Field Polymer Flooding

Kun Wang*, Hui Liu*, Jia Luo*, Song Yu*, Zhangxin Chen*, Peng Zhang†
*Department of Chemical and Petroleum Engineering, University of Calgary
Email: {wang30, liu1, jialuo, song.yu2, zhachen}@ucalgary.ca
†College of Engineering and Applied Sciences, Stony Brook University
Email: peng.zhang@stonybrook.edu

***Abstract*—In recent years, polymer flooding has become a mature technology to improve the reservoir recovery rate and has been successfully implemented in many oil fields. Because of high operating costs of a polymer flooding project and the impact to the environment, the polymer injected per unit volume should be tracked for each polymer flooding pattern. Hence, the full-field reservoir simulations with fine-scale grids are needed to capture fine-scale phenomena and to optimize the process. To meet these objectives, a polymer module has been developed in our in-house parallel black oil simulator, which allows parallel simulations using clusters and supercomputers. With our parallel simulator, the elapsed time of full-field simulations with millions of grid blocks can be reduced from days to hours or minutes, and even for a much larger model with hundreds of millions of grid blocks, the simulations can be finished in practical time. In order to guarantee the computational efficiency and the parallel scalability, an inexact Newton method is applied and a new CPR (Constrained Pressure Residual)-type preconditioner is designed. In this paper, a SPE10-based polymer flooding case is tested, the performance of nonlinear and linear solvers are robust, and an encouraging parallel scalability is obtained. The second case is a full-field polymer flooding case, for which a coarse grid model and a fine grid model are both used. The results show the differences of the oil production rate and the water cut, which illustrates the accuracy of fine-grid simulations. For the fine grid model, up to 1024 CPU cores are employed, and an excellent speedup is achieved.**

I. Introduction

For reservoirs with a highly water mobility ratio or highly heterogeneous permeability, polymer flooding has become a widely applied EOR (Enhanced Oil Recovery) strategy to improve sweep efficiency. The addition of long-chain polymer molecules may cause a reduction in permeability and may also cause water to behave as a non-Newton fluid. Since the viscosity of the water phase is strongly affected by the polymer concentration, it is important to capture polymer fronts sharply, which requires fine-grid large-scale reservoir simulations. The issue of the fine-grid large-scale reservoir simulations is the computational time. Restricted by the memory size and CPU performance, it may take tens of hours or even days for a polymer flooding case with millions of grid blocks. On the other hand, the development of supercomputers and clusters and the growth of their computational performance [25] are rapid. Hence, a parallel computation technology is a natural choice for reservoir simulations to improve the efficiency.

A linear solver is one the biggest challenges for large-scale reservoir simulations. The ILU (incomplete LU factorization) preconditioners [1] are usually employed as the default solvers by most of the commercial software. Due to the low efficiency of ILU preconditioners for large-scale problems, the CPR (constrained pressure residual) preconditioner was developed [17], [5]. To improve the effectiveness and efficiency of the CPR preconditioner, various CPR-type preconditioners were developed [10], [18], [11].

During decades of the development of reservoir simulation, several parallel simulators were developed. A parallel compositional simulator was developed by Killough et al., which demonstrated that a highly efficient parallel model could be generated for an n-component, three-phase, EOS (equations of state) reservoir simulator in a distributed-memory parallel computer [21]. Rutledge et al. implemented an IMPES (implicit-pressure explicit saturation) compositional simulator using massive SIMD computers [20]. A two-dimensional two-phase (oil and water) simulator was presented by Kaarstad et al.. In 2009, Dogru et al. developed a parallel simulator GigaPOWERS, which was capable of simulating one billion grid blocks [22]. A black oil simulator coupled with geomechanics for large-scale parallel computation is developed in 2015 [18], [23], [13].

In this paper, a parallel black oil simulator with a polymer flooding option is developed for both personal computers and clusters. The objective of this simulator is to provide the reservoir engineer with a strong and efficient tool for industrial use. The simulator employs parallel computation for every part of the simulation, including the grid generation and partition, data and well management, compressed sparse row strorage for matrices, preconditioners and linear solvers, and implementation of physical models. In §II, a grid and data management system of our simulator are introduced; in §III, the mathematical description of the black oil model as well as the polymer equation are presented; nonlinear and linear solvers are given in §IV and §V; two large-scale numerical experiments are tested in §VI; finally, conclusions are draw in §VII.

II. Grids and Data Management

Regular Cartesian grids are the most frequently used grid in reservoir simulation. However, it is difficult for the regular Cartesian grids to describe a complex geological model, such as pinchouts and faults. To describe such model precisely, it is necessary to define the position of a grid cell by its corner point locations. Then the grid consists of hexahedra and each hexahedron is shaped with eight corners and bilinear planes

978-1-5090-2403-2/16 $31.00 © 2016 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2016.35

as surfaces. This kind of grid is called corner point grid; see Fig 1. For the Cartesian grid, the flux across a face can be

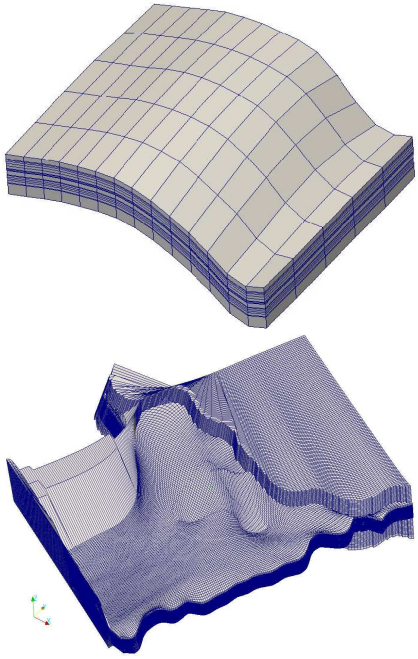

Fig. 1. Two examples of corner point grids

calculated by the properties on the local cell and its directly neighbour cell, which is call two-point flux-approximation (TPFA) scheme. The TPFA scheme results in a 5-point scheme in 2D and a 7-point scheme in 3D. However, when a corner point grid or/and a permeability tensor are used, the TPFA scheme can lead to an inconsistent finite difference method and significant errors if the grid is highly distorted. Hence, the multi-point flux-approximation (MPFA) scheme should be used, which may result in a 9-point scheme in 2D and 27-point scheme in 3D.

After the grid is generated, there is another important thing for parallel computation, which is the grid partitioning. Grid partitioning aims to average the total work to each processor and minimize the data communication between processors. Since the Hilbert Spacing-filling Curve (HSFC) Method, which is a geometry information based method, has good spatial locality, takes low computational cost and guarantees the partitioning quality, it is implemented in our simulator and serves as the default partitioning method. It assumes that a cell has the largest possibility to communicate with its neighbour cells, which is true for finite element methods, finite difference (volume) methods. The HSFC method can be described by three steps: 1) using a linear mapping to map the computational domain Ω to $(0,1)^3$; 2) employing the Hilbert method to define a map from $(0,1)^3$ to $(0,1)$; 3) partitioning $(0,1)$ into N_p sub-intervals that has the same amount of cells. More details about the HSFC method can be found in [19], [13].

All the data related with the grid, such as the porosity and the permeability, are distributed into each MPI process, as well as the matrix and the vector resulted from the Jacobian. For the matrix, the dense storage and compressed sparse row storage (both sequential and parallel versions) are supported.

III. Mathematical Model

The polymer flooding model supported by our simulator is based on the standard black oil model. The standard black oil model assumes that: 1) three phases (oil, water and gas) coexist in the reservoir; 2) oil and water can only exist in the oil and water phase, respectively; 3) gas can exist both in the oil and gas phase; 4) the reservoir is isothermal. Combining Darcy's law and mass conservation law for each component, the black oil model can be written as:

$$\frac{\partial}{\partial t}(\frac{\phi s_o}{b_o}) = \nabla \cdot (\frac{\mathbf{K}K_{ro}}{\mu_o b_o}\nabla\Phi_o) + q_o, \tag{1}$$

$$\frac{\partial}{\partial t}(\frac{\phi s_w}{b_w}) = \nabla \cdot (\frac{\mathbf{K}K_{rw}}{\mu_w b_w}\nabla\Phi_w) + q_w, \tag{2}$$

$$\frac{\partial}{\partial t}(\frac{\phi r_s s_o}{b_o}) + \frac{\partial}{\partial t}(\frac{\phi s_g}{b_g}) = \nabla \cdot (\frac{\mathbf{K}K_{ro}r_s}{\mu_o b_o}\nabla\Phi_o) + \nabla \cdot (\frac{\mathbf{K}K_{rg}}{\mu_g b_g}\nabla\Phi_g) + q_o r_s + q_g, \tag{3}$$

where ϕ and $\mathbf{K}$ are porosity and permeability, r_s is the solution gas-oil ratio for saturated oil, for phase α ($\alpha = o, w, g$), Φ_α is potential, and s_α, μ_α, p_α, b_α, $K_{r\alpha}$ and q_α are the saturation, viscosity, pressure, formation volume factor, relative permeability and production rate, respectively. These variables have the following relations:

$$\begin{aligned}
&\Phi_\alpha = p_\alpha + \rho_\alpha^m \wp z,\\
&s_o + s_w + s_g = 1,\\
&p_w = p_o - p_{cow}(s_w),\\
&p_g = p_o + p_{cog}(s_g),
\end{aligned}$$

where ρ_α^m is the mass density of phase α, $p_{co\alpha}$ are the capillary pressure between oil phase and α phase, $\wp$ is the gravitational constant and z is the reservoir depth.

When polymer flooding is applied to improve the oil recovery rate, the mobility of the water decreases due to the polymer retention. Modification is required to the water equation and an additional equation for polymer mass conservation should be added to the black oil systems. The water and polymer equations can be written as follows

$$\frac{\partial}{\partial t}(\frac{\phi s_w}{b_w}) = \nabla \cdot (\frac{\mathbf{K}K_{rw}}{\mu_{w,\text{eff}} b_w R_k}\nabla\Phi_w) + q_w, \tag{4}$$

$$\frac{\partial}{\partial t}(\frac{\phi s_w C_p}{b_w} + \rho_{r,\text{ref}}(1-\phi_{\text{ref}})C_p^a) = \nabla \cdot (\frac{\mathbf{K}K_{rw}}{\mu_{p,\text{eff}} b_w R_k}\nabla\Phi_o) + C_p q_w, \tag{5}$$

where C_p is the polymer concentration, ϕ_{ref} is the porosity at reference pressure, $\rho_{r,\text{ref}}$ is the rock density at reference pressure, C_p^a is the polymer adsorption concentration, $\mu_{\alpha,\text{eff}}$ is the effective viscosity of polymer ($\alpha = p$) and water ($\alpha = w$),

and R_k is the relative permeability reduction factor for the aqueous phase due to polymer retention.

IV. Nonlinear Method

The black oil based models and the compositional model are highly nonlinear systems. The Newton (also called Newton-Raphson) method is the most commonly used method for nonlinear problems. At each Newton iteration, the nonlinear system is linearized and a linear system is required to be solved accurately. For large-scale reservoir simulations, the computational cost of the solution of linear systems is very expensive. Hence, an inexact Newton method is employed by our simulator, which can guarantee the convergence of the nonlinear iterations and prevent the linear system from oversolved. The inexact Newton method can be described in Algorithm 1. Actually, θ_l in Algorithm 1 is the relative error for the solution of the linear system. More details about the inexact Newton method can be found in [6]. In order to improve the convergence rate of the nonlinear iterations, linear extrapolation of the values from the former several time steps is used as the initial guess for the current time step. Taking the pressure p as an example, let p^{n-1} and p^n be the pressure at time t_{n-1} and t_n, respectively. The initial guess of pressure at time t_{n+1} is

$$p^{n+1} = p^n + (p^n - p^{n-1})\frac{t_{n+1} - t_n}{t_n - t_{n-1}}. \tag{7}$$

However, the linear extrapolation only can be used when the well constraints keep the same as the former time step.

Algorithm 1 The inexact Newton Method for a nonlinear problem $F(x) = 0$

1: Given an initial guess x^0 and stopping criterion ϵ, let $l = 0$ and calculate $F(x^l)$.
2: **while** $\left\|F(x^l)\right\| \geq \epsilon$ **do**
3: Calculate the Jacobian matrix $J = \frac{DF(x^l)}{Dx^l}$.
4: Find δx such that

$$\left\|F(x^l) - J\delta x\right\| \leq \theta_l, \tag{6}$$

where θ_l is set to $\gamma \left(\frac{\|F(x^l)\|}{\|F(x^{l-1})\|}\right)^{\beta}$, $\gamma = 0.5$ and $\beta = \frac{\sqrt{5}+1}{2}$.
5: Let $l = l + 1$ and $x^l = x^{l-1} + \delta x$.
6: **end while**
7: $x^* = x^l$ is the solution of the nonlinear system.

V. Linear Solver

For the linear system resulted in Algorithm 1, the ILU [1] preconditioned linear solvers are usually used as the default solvers. However, for large-scale reservoir simulations, especially for the reservoirs with highly heterogeneous permeability, the ILU preconditioners are not effective enough. The constrained pressure residual (CPR)-type preconditioners [17], [5], [11], [12], [18] are considered as the most effective and efficient method for reservoir simulations. In our simulator, a family of CPR-type preconditioners are implemented, and the CPR-FPR [11], [12], [18] preconditioner is considered as the most efficient one and is used as the default preconditioner. We will briefly describe the CPR-FPF preconditioner in the following. Let $Jx = b$ be the linear system resulted in Algorithm 1. For the standard black oil model, the matrix J can be represented as:

$$J = \begin{pmatrix} J_{ws_w} & 0 & J_{wp} & J_{wC} \\ J_{os_w} & J_{oX} & J_{op} & 0 \\ J_{gs_w} & J_{gX} & J_{gp} & 0 \\ J_{ps_w} & 0 & J_{pp} & J_{pC} \end{pmatrix}, \tag{8}$$

where matrix $J_{\alpha s_w}(\alpha = w, o, g, p)$ is the submatrix corresponding to the water saturation unknowns, matrix $J_{\alpha X}(\alpha = o, g)$ is the submatrix corresponding to the gas saturation or bubble point pressure unknowns depending on the status of the reservoir (undersaturated or saturated), and matrix $J_{\alpha p}(\alpha = w, o, g, p)$ is the matrix corresponding to the pressure unknowns. Before the CPR-FPF preconditioning process, the alternative block factorization (ABF) decoupling method is used as the pre-processing step to weaken the strong nonlinear coupling between the pressure unknowns and other unknowns. The linear system after the ABF decoupling process can be written as $D^{-1}Jx = D^{-1}b$, where

$$D = \begin{pmatrix} \mathbf{D}(J_{ws_w}) & 0 & \mathbf{D}(J_{wp}) & \mathbf{D}(J_{wC}) \\ \mathbf{D}(J_{os_w}) & \mathbf{D}(J_{oX}) & \mathbf{D}(J_{op}) & 0 \\ \mathbf{D}(J_{gs_w}) & \mathbf{D}(J_{gX}) & \mathbf{D}(J_{gp}) & 0 \\ \mathbf{D}(J_{ps_w}) & 0 & \mathbf{D}(J_{pp}) & \mathbf{D}(J_{pC}) \end{pmatrix},$$

for the matrix (8) and $\mathbf{D}(\cdot)$ stands for a diagonal matrix. For simplicity, we still denote the pre-processed linear systems by $Jx = b$. The main idea of the CPR-FPF preconditioner is to use the algebraic multigrid (AMG) method for the submatrix corresponding to the pressure unknowns, and then employ a ILU method for the whole linear system both before and after the AMG method to guarantee the effectiveness and efficiency of the preconditioner. Introducing a restrict operator Π_p, which can restrict from the full system vector to a pressure system vector, and denoting the ILU and AMG preconditioners by R^{-1} and M_p respectively, we can represent the CPR-FPF preconditioner as in Algorithm 2.

VI. Numerical Experiments

Numerical experiments in this section are tested on the cluster GPC (General Purpose Cluster) from Canadas largest supercomputer centre SciNet [24]. The GPC consists of 3,780 nodes (IBM iDataPlex DX360M2) with a total of 30,240 cores (Intel Xeon E5540) at 2.53GHz, with 16GB RAM per node (some larger-memory nodes up to 32GB). Our jobs were run on the nodes with 16GB memory connected with non-blocking DDR InfiniBand. The 16GB memory and 8 cores of each employed node are fully used in our computation, and each MPI process runs on a core.

Algorithm 2 CPR-FPF Preconditioner, CPR-FPF(x^0)

1: Given an initial guess x^0,

2: Employ the first stage of CPR-FPF preconditioning process, ILU process to the global system, and get the approximate solution x^1

$$x^1 = x^0 + R^{-1}(b - Jx^0). \tag{9}$$

3: Employ the second stage of CPR-FPF preconditioning process, AMG to the pressure system, and get the approximate solution x^2

$$x^2 = x^1 + \Pi_p M_p^{-1} \Pi_p^T (b - Jx^1). \tag{10}$$

4: Employ the third stage of CPR-FPF preconditioning process, another ILU process to the global system, and get the approximate solution x^3

$$x^3 = x^2 + R^{-1}(b - Jx^2). \tag{11}$$

5: Output x^3.

A. A SPE10-based Polymer Flooding Case

The tenth SPE comparative project is a challenging test case due to its highly heterogeneous permeability and porosity. Its permeability varies between 6.65×10^{-11} and 20 Darcy, and its porosity varies between 0 and 0.5, see Fig 2.

The dimensions are $1200 \times 2200 \times 170$ (ft) and has 1.1 million ($60 \times 220 \times 85$) grid cells. The top 35 layers (70 ft) represent the Tarber formation and the bottom 50 layers (100 ft) represent the Upper Ness formation. Four producers are placed at four corners of the reservoir, and one injector in the centre. The total simulation time is 2,000 days. More details can be found in [8]. In the case tested in this subsection, the SPE10 case, which is originally a oil-water two phase case, is modified to a black oil model case. The PVT properties of oil and gas phase can be found in Table I. The polymer

TABLE I
OIL AND GAS PVT TABLE

Pressure psia	R_s MSCF/STB	B_o RB/STB	μ_o cp	B_g SCF/STB	μ_g cp
400	0.0165	1.012	3.5057	1.96	0.0140
4000	0.1130	1.01278	2.9972	0.84	0.0160
8000	0.1583	1.10759	2.7775	0.59	0.0175
10000	0.1810	1.155	2.6675	0.42	0.0195

injection begins at the 300th day, then stops at the 800th day. The concentration of the injected polymer is 1.67kg/m^3. The properties of polymer is given in Table II. In Table III, "NP" is the number of MPI processes employed in the computation, "Newton" is the number of Newton iterations, "Linear" is the number of linear solver iterations, "Time" is the elapsed computation time, and "Scalability" is the parallel scalability. We can see that the scalability is around 80 percent and the Newton and linear iterations are robust when up to 128 MPI processes are used, which can illustrate the efficiency and the effectiveness of our simulator. Since this case only contains 1

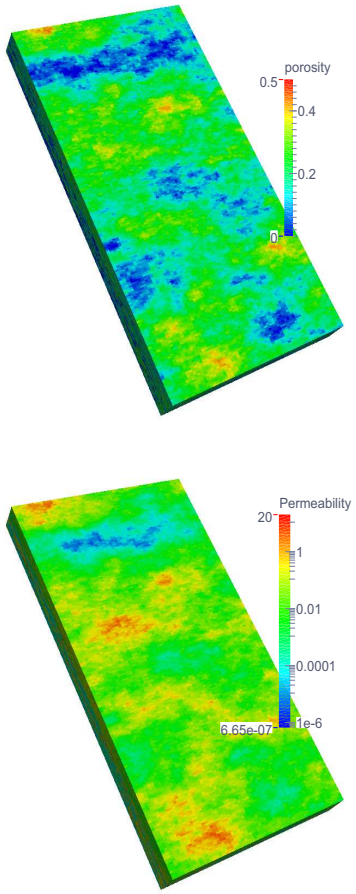

Fig. 2. SPE10 Geological Information

TABLE II
POLYMER PROPERTIES

Properties		
Adsorption	concentration (kg/m^3)	adsorption (kg/kg)
	0.0	0.0
	2.0	0.0015
	8.0	0.0025
Rock	dead pore space	0.15
	residual resistance factor	2.67
	rock density	1000.0 kg/m^3
	maximum polymer adorption	0.0035 kg/kg
Viscosity	concentration	water viscosity multiplyer
	0.0	1.0
	7.0	20.0

million of grid blocks and each process deals with less than five thousand blocks when 256 MPI processes are used, the effectiveness of the domain decomposition method becomes weak and communication between MPI processes becomes dominant, which leads to the increase of the Newton/linear iterations and the loss of the parallel scalability. The speedup of this case can be seen in Fig 3.

TABLE III
SPE10-BASED POLYMER FLOODING CASE

NP	Newton	Linear	Time	Scalability
32	2721	67,986	28,385	-
64	2896	75,975	16,840	0.843
128	2978	80,578	8941	0.794
256	3956	119,604	5469	0.649

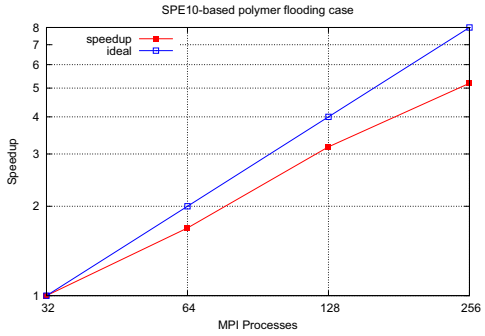

Fig. 3. SPE10-based Polymer Flooding Case Speedup

TABLE IV
FULL-FIELD POLYMER FLOODING CASE

NP	Newton	Linear	Time	Scalability
64	666	13,097	10,291	-
128	669	13,496	5482	0.939
256	674	13,642	3004	0.856
512	689	15,332	1731	0.743
1024	689	15,604	809	0.795

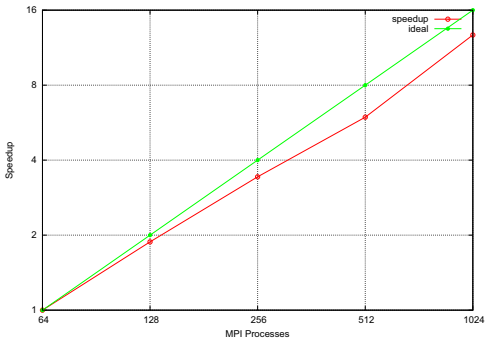

Fig. 5. Full-field polymer flooding case: speedup of fine-grid model

B. A Full-Field Polymer Flooding Case

In this case, two models with different grid are tested. The coarse grid has $40 \times 40 \times 2 = 3200$ blocks, while the fine grid contains $400 \times 400 \times 20 = 3,200,000$ blocks. Two wells, including one producer and one injector, are considered in this case. The process of the polymer injection is the same as the former case, which begins at the 300th day and stops at the 800th day. The concentration of the injected polymer is 5kg/m^3. The properties of polymer is the same as in Table II.

Firstly, we compared the results of fine-grid model and coarse-grid model; see Fig 4. The oil production rate and the water cut (water/liquid ratio) from the two models are different, which draws the importance of the large-scale fine-grid simulations. In Table IV, the performance of our simulator

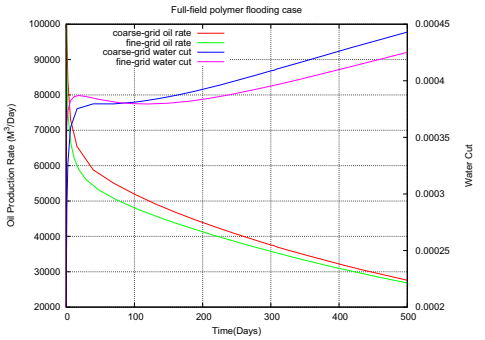

Fig. 4. Full-field polymer flooding case: comparison of fine-grid model and coarse-grid model

on this full-field polymer flooding case is given. We can see that the simulation is scalable up to 1024 MPI processes, and around 80 percent or even more parallel efficiency is obtained. The speedup of this case is shown in Fig 5.

VII. CONCLUSION

In this paper, a black oil simulator with polymer injection option is implemented for large-scale parallel reservoir simulations. Both the Cartesian grid and the corner point grid are supported by our simulator. HSFC method is implemented for the parallel grid partition. Inexact Newton method is used to solve the strongly coupled nonlinear black-oil systems, which can avoid the linear system oversolved. A new CPR-type preconditioner, CPR-FPF, is developed to ensure the effectiveness of the linear solver and the scalability of parallel computation. The SPE10-based polymer flooding case and the full-field case both illustrate that the linear solver is robust and the simulator is scalable. For the full-field case, the results of coarse grid model and fine grid model gives different prediction on oil production rate and water cut. It shows the importance of large-scale fine grid reservoir simulations, especially the cases requiring high accuracy.

ACKNOWLEDGMENT

The support of Department of Chemical and Petroleum Engineering, University of Calgary and Reservoir Simulation Research Group is gratefully acknowledged. The research is partly supported by NSERC/AIEES/Foundation CMG, AITF iCore, IBM Thomas J. Watson Research Center, and the Frank and Sarah Meyer FCMG Collaboration Centre for Visualization and Simulation. The research is also enabled in part by support provided by WestGrid (www.westgrid.ca), SciNet (www.scinethpc.ca) and Compute Canada Calcul Canada (www.computecanada.ca).

REFERENCES

[1] J. R. Wallis, Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration, SPE Reservoir Simulation Symposium, 1983.

[2] T. M. Al-Shaalan, H. M. Klie, A. H. Dogru, M. F. Wheeler, Studies of robust two stage preconditioners for the solution of fully implicit multiphase flow problems, in: SPE Reservoir Simulation Symposium, 2009.
[3] R. E. Bank, T. F. Chan, W. M. C. Jr., R. K. Smith, The alternate-blockfactorization procedure for systems of partial differential equations, BIT Numerical Mathematics 29 (4) (1989) 938954.
[4] X.-C. Cai, M. Sarkis, A restricted additive schwarz preconditioner for general sparse linear systems, SIAM Journal on Scientific Computing 21 (2) (1999) 792797.
[5] H. Cao, H. A. Tchelepi, J. R. Wallis, H. E. Yardumian, Parallel scalable unstructured cpr-type linear solver for reservoir simulation, in: SPE Annual Technical Conference and Exhibition, 2005.
[6] T. Chen, N. Gewecke, Z. Li, A. Rubiano, R. Shuttleworth, B. Yang, X. Zhong, Fast computational methods for reservoir flow models.
[7] Z. Chen, G. Huan, Y. Ma, Computational methods for multiphase flows in porous media, vol. 2, Siam, 2006.
[8] M. A. Christie, M. J. Blunt, Tenth spe comparative solution project: A comparison of upscaling techniques., SPE Reservoir Evaluation & Engineering 4.4 (2001): 308-317. 4 (4) (2001) 308317.
[9] R. D. Falgout, U. M. Yang, hypre: A library of high performance preconditioners, in: Computational Science-ICCS 2002, Springer Berlin Heidelberg, 2002, pp. 632641.
[10] X. Hu, W. Liu, G. Qin, J. Xu, C. Zhang, Development of a fast auxiliary subspace pre-conditioner for numerical reservoir simulators, in: SPE Reservoir Characterisation and Simulation Conference and Exhibition, 2011.
[11] H. Liu, K.Wang, Z. Chen, A family of constrained pressure residual preconditioners for parallel reservoir simulations, Numerical linear algebra with applications 00 (00) (2015) 132.
[12] H. Liu, K.Wang, Z. Chen, K. E. Jordan, Efficient multi-stage preconditioners for highly heterogeneous reservoir simulations on parallel distributed systems, in: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, 2015.
[13] H. Liu, K. Wang, Z. Chen, K. E. Jordan, H. Deng, J. Luo, A parallel framework for reservoir simulators on distributed-memory supercomputers, in: SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition, 2015.
[14] K. Stüben, Algebraic multigrid (amg): an introduction with applications, in: GMD-Forschungszentrum Informationstechnik, 1999.
[15] K. Stüben, An introduction to algebraic multigrid, multigrid, Multigrid (2001) 413532.
[16] K. Stüben, A review of algebraic multigrid, Journal of Computational and Applied Mathematics 128 (1) (2001) 281309.
[17] J. R. Wallis, R. P. Kendall, T. E. Little, Constrained residual acceleration of conjugate residual methods, in: SPE Reservoir Simulation Symposium, 1985.
[18] K. Wang, H. Liu, Z. Chen, A scalable parallel black oil simulator on distributed memory parallel computers, Journal of Computational Physics 301 (2015) 1934.
[19] H. Liu, Dynamic load balancing on adaptive unstructured meshes, 10th IEEE International Conference on High Performance Computing and Communications, 2008.
[20] J. M. Rutledge, D. R. Jones, W. H. Chen and E. Y Chung, The use of massively parallel SIMD computer for reservoir simulation, SPE Paper 21213 presented at the eleventh SPE Symposium on Reservoir Simulation, Anaheim, 1991.
[21] J. E. Killough, R. Bhogeswara, Simulation of compositional reservoir phenomena on a distributed-memory parallel computer, Journal of Petroleum Technology 43(11) 1368-1374, 1991.
[22] Ali H. Dogru, Larry Siu Kuen Fung, Usuf Middya, Tareq Al-Shaalan and Jorge Alberto Pita, A next-generation parallel reservoir simulator for giant reservoirs, SPE/EAGE Reservoir Characterization & Simulation Conference, 2009.
[23] J. Luo, Z. Chen, K. Wang, H. Deng, and H. Liu. An Efficient and Parallel Scalable Geomechanics Simulator for Reservoir Simulation, SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition. Society of Petroleum Engineers, 2015.
[24] C. Loken et al., SciNet: lessons learned from building a power-efficient top-20 system and data centre, Journal of Physics: Conference Series, 256(1), 012-016, 2010.
[25] M. Qiu, Z. Ming, J. Li, K. Gai, and Z. Zong, Phase-change memory optimization for green cloud with genetic algorithm, Computers, IEEE Transactions on 64.12 (2015): 3528-3540.