SPE-193872-MS

# An Adaptive Preconditioning Strategy to Speed up Parallel Reservoir Simulations

Kun Wang and Hui Liu, University of Calgary; Lin Yan, Exploration and Development Research Institute, PetroChina; Jia Luo, University of Calgary; Keliu Wu, China University of Petroleum; Jing Li, University of Calgary; Fuli Chen, Exploration and Development Research Institute, PetroChina; Xiaohu Dong, China University of Petroleum; Zhangxin Chen, University of Calgary



This paper was prepared for presentation at the SPE Reservoir Simulation Conference held in Galveston, Texas, USA, 10–11 April 2019.



## Abstract

In reservoir simulation, an ILU preconditioner is the most widely used preconditioner for preconditioning linear systems due to its simplicity and low computational cost. However, an ILU preconditioner sometimes is not effective enough, especially for a large-scale parallel reservoir simulation problem with a highly heterogeneous geological model. A constrained pressure residual (CPR) preconditioner is considered a more efficient one, which employs two stages of a preconditioning process: the first stage uses the Algebraic Multi-grid (AMG) method to solve a pressure system, and the second stage uses the ILU method to solve the whole system. Its disadvantage is the high computational cost of the AMG method. In order to reduce the computation costs on preconditioners and the resulting linear solvers, we have developed an adaptive preconditioning strategy [16] to automatically select a preconditioner between an ILU preconditioner and a CPR preconditioner or switch the ILU preconditioner to the CPR preconditioner and vice versa during a linear solution process. In this paper, the adaptive strategy is further analyzed and studied to understand its numerical performance and to choose optimal switch criteria.

## Introduction

Recently, more high-resolution geological models are employed in reservoir simulation by the oil and gas industry. In order to reduce the computational time for large-scale simulations, parallel computing techniques are adopted in these simulations, and parallel reservoir simulators [1–5] have been successfully developed.

To obtain high parallel scalability, parallel preconditioning techniques have been developed for years. The point-wise and block-wise incomplete factorization (ILU) [6,7] preconditioners are the most frequently used ones in commercial simulators, and they are usually employed as a sub-domain solver with the Restricted Additive Schwarz (RAS) method [8] if parallel computing is applied. However, the effectiveness of the RAS method with ILU as a sub-domain solver is not effective enough, and the number of linear iterations significantly increases as the number of CPU cores (the number of sub-domains) increases.

The constrained pressure residual (CPR) method [9] employs two preconditioning stages and is considered a more effective preconditioner than ILU. The CPR method first uses the Algebraic Multi- grid method [10] to solve a pressure system, and then uses the ILU method to smooth the whole system. There are also some extensions [4,11,12,14] of a CPR preconditioner, such as the fast-auxiliary space preconditioner (FASP) [15] and a family of CPR-type preconditioners [13]. A CPR preconditioner has also been applied in parallel reservoir simulation and implemented in parallel simulators [4,5,11]. A CPR preconditioner can significantly reduce the number of linear iterations. The disadvantage of the CPR preconditioner is its huge computational cost spent on setting up the AMG solver.

In our previous work [16], we developed an adaptive preconditioning strategy, which allows a preconditioner switch between an ILU preconditioner and a CPR preconditioner according to each specific linear system and the convergence rate of a linear solver. A preconditioner switch criterion $e^{switch}$ is an important factor in the adaptive strategy, which determines when to use the ILU preconditioner and the CPR preconditioner. In this paper, we study the adaptive strategy by testing the preconditioner switch criterion $e^{switch}$ with different value. With the numerical results, we can recommend that the switch criterion $e^{switch} = 0.001$ is a safe value to obtain almost the optimal performance of the linear solvers.

## Governing equations

In the paper, we consider the black-oil model [17, 25], and the method proposed can be easily applied to the compositional model and the thermal model. The black-oil model assumes that (1) there are three phases (the oil, water, and gas phases) and three components (the oil, water, and gas components) co- existing in a reservoir; (2) the gas component can be in the oil and gas phases; (3) the oil and water components can only be in the oil and water phases, respectively. The mass conservation equations can be written as [18]

$$\frac{\partial}{\partial t}(\phi s_o \rho_o^o) = \nabla \cdot (\frac{KK_{ro}}{\mu_o} \rho_o^o \nabla \Phi_o) + q_o \, , \tag{1}$$

$$\frac{\partial}{\partial t}(\phi s_w \rho_w) = \nabla \cdot (\frac{KK_{rw}}{\mu_w} \rho_w \nabla \Phi_w) + q_w \, , \tag{2}$$

$$\frac{\partial(\phi \rho_o^g s_o + \phi \rho_g s_g)}{\partial t} = \nabla \cdot (\frac{KK_{ro}}{\mu_o} \rho_o^g \nabla \Phi_o) + \nabla \cdot (\frac{KK_{rg}}{\mu_g} \rho_g \nabla \Phi_g) + q_o^g + q_g \, , \tag{3}$$

where $\Phi_\alpha$, $s_\alpha$, $\mu_\alpha$, $\rho_\alpha$, $K_{r\alpha}$, and $q_\alpha$ are the potential, saturation, viscosity, density, relative permeability and production of phase $\alpha = o, w, g$, $\phi$ is porosity, and $K$ is permeability. Together, there are the following constraints:

$$\begin{aligned} s_o + s_w + s_g &= 1, \\ p_w &= p_o - p_{cow}, \\ p_g &= p_o + p_{cog}. \end{aligned}$$

Then a close system is established.

We choose the pressure $p$, the water saturation $s_w$ and the gas saturation $s_g$ as the primary variables. The fully implicit method (FIM) in time is used to discretize the nonlinear equations (1)–(3). At each nonlinear iteration in a time step, the increments of the primary variables can be obtained by solving the following linear equation:

$$\begin{pmatrix} A_{o,p} & A_{o,s_w} & A_{o,s_g} \\ A_{w,p} & A_{w,s_w} & A_{o,s_g} \\ A_{g,p} & A_{g,s_w} & A_{g,s_g} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta s_w \\ \delta s_g \end{pmatrix} = \begin{pmatrix} b_o \\ b_w \\ b_g \end{pmatrix}. \tag{4}$$

The linear equation (4) can be simplified as

$$Ax = b. \tag{5}$$

## Preconditioning technique for parallel reservoir simulation

To solve the linear equation (5), iterative linear solvers, such as GMRES and BiCGstab, are usually employed. Preconditioners are required to achieve fast convergence of linear iterations. The ILU preconditioners are the most widely used ones in the commercial reservoir simulators. For parallel reservoir simulation, the restricted additive Schwarz (RAS) method is usually used as a preconditioner, which employs the ILU preconditioners as sub-domain solvers. We use RAS-ILU to represent this preconditioning method in parallel computing. The computational cost of the ILU preconditioners is low, but it may not be very effective for problems with a large number of grid blocks and highly heterogeneous permeability, especially for parallel reservoir simulation. The constrained pressure residual (CPR) method is a more efficient preconditioner. As mentioned above, the CPR preconditioner employs two stages of a preconditioning process to separately handle the pressure system and the whole system. For the first stage, since the pressure system behaves with an elliptic property [24], the Algebraic Multi-grid (AMG) method is used as the first stage preconditioner to efficiently solve the pressure system. Usually, it is not necessary to accurately solve the pressure system, and only one V- cycle of the AMG method is enough for a preconditioning process. For the second stage, the ILU preconditioners are used to solve the whole system as a smoother. Some other extended CPR-type preconditioners can be found in the literature [13]. The CPR preconditioning process can be described as follows:

**CPR Preconditioning Process**

1. With the given initial vector $x_0$, calculate the residual $r = b - Ax_0$.
2. Get the pressure residual $r_p = \Pi_p r$.
3. Employ AMG V-cycle to solve the pressure system $A_{o,p} x_p = r_p$.
4. Map $x_p$ back to the whole vector $x_p^a = \Pi_p^T x_p$.
5. $x_1 = x_0 + x_p^a$.
6. Employ ILU to solve $x_2^a = R^{-1}(b - Ax_1)$.
7. $x_2 = x_1 + x_2^a$.
8. Output $x_2$.

The CPR preconditioner is able to significantly reduce the number of linear iterations. However, the computational cost of each CPR preconditioning process is huge. Compared with an ILU preconditioner, if the CPR preconditioner is employed for the problems whose linear systems are relatively easy to solve, the number of linear iterations is fewer but more computation time may be required, as addressed above.

## An adaptive preconditioning strategy

An ILU preconditioner and a CPR preconditioner both have their own advantage. There are two kinds of situations in which the ILU preconditioner can take less computation time than the CPR preconditioner: (1) when linear systems are easy to solve, the ILU preconditioner is effective enough; (2) when the stopping criterion of a linear system is easy to achieve by the ILU preconditioned linear solver (even if the linear system is difficult to solve).

The first kind of situation mostly happens when the number of grid blocks in a reservoir model is small or a geological model is isotropic or parallel computing is not employed. The second kind of situation happens when a large-scale heterogeneous model is used in parallel simulations and an inexact Newton method [4] is applied to change the stopping criterion of a linear system for each nonlinear iteration. In Fig. 1, we can see the convergence history of the RAS-ILU preconditioned GMRES solver when we use 128 CPU cores to solve a black-oil SPE10 [16,19] benchmark problem. As we can see, the residual drops very fast at the beginning of iterations but stays almost unchanged when the residual reaches a certain value. If the required stopping criterion of a linear iteration is larger than that value, the ILU preconditioner will be effective enough and efficiently finish this job. From this figure, we can observe one more phenomenon, which is that the linear system becomes more and more difficult to solve during the processes of nonlinear iterations. That is because the initial absolute residual of the linear system (this absolute residual is also the residual of the nonlinear equation since the initial guesses for the linear system are zero) becomes smaller and smaller. It can also be seen from Table 1, where we employ the CPR preconditioner. To reduce the relative residual of the linear system (shown as "Relative residual achieved" in Table 1), the linear solver employs more iterations as the nonlinear absolute residual becomes smaller.

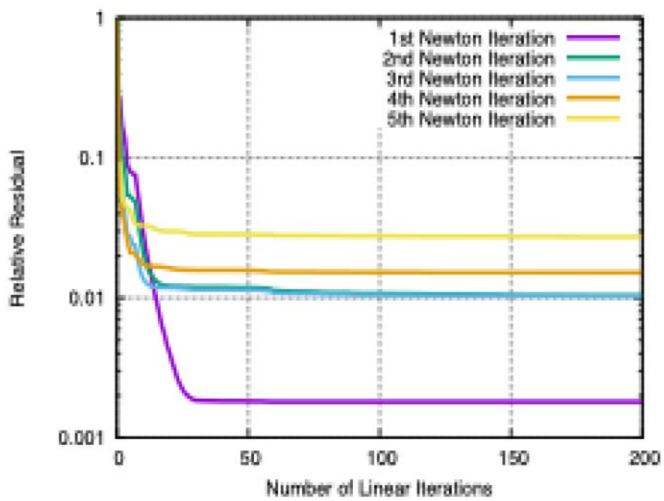


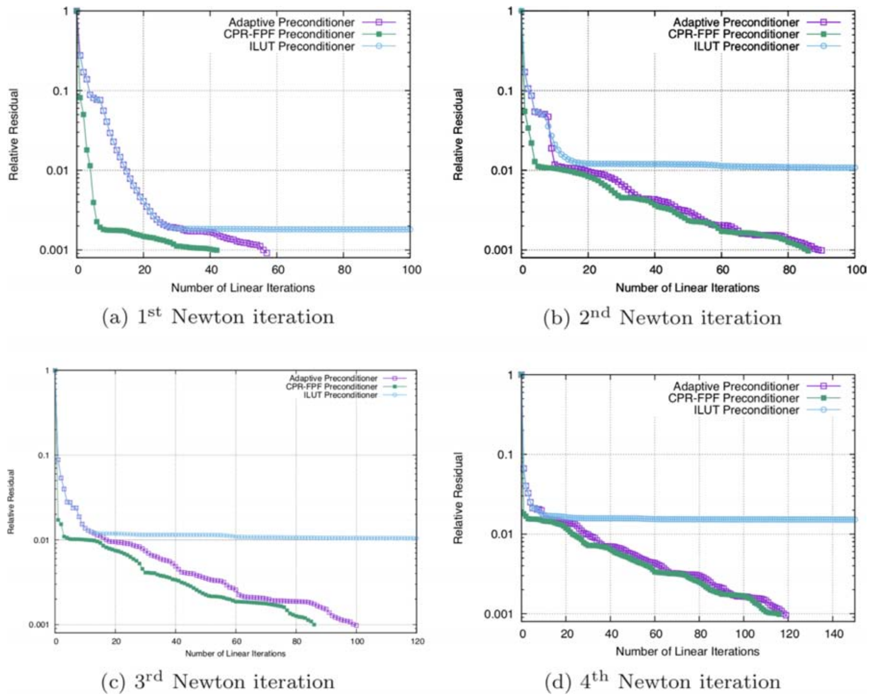

**Figure 1—RAS-ILU preconditioned GMRES convergence history**

**Table 1—Numerical performance of a time step, [16]**

| Nonlinear | Absolute residual | Linear | Relative residual achieved |
|---|---|---|---|
| 1st | 0.422 | 1 | 0.0356 |
| 2nd | 0.0215 | 15 | 0.00663 |
| 3rd | 0.000349 | 50 | 0.0111 |
| 4th | 4.93e-6 | 73 | 0.0231 |

Based on the above observation, we introduce an adaptive preconditioning strategy. Basically, an ILU preconditioner is preferred if it works well; otherwise, it is necessary to bring a CPR preconditioner to the linear solver.

To guarantee a fast convergence of linear iterations, the convergence rate is monitored if the ILU preconditioner is used. If the convergence rate is not satisfied, the preconditioner will switch to the CPR preconditioner. This preconditioner switching should not affect the convergence of linear iterations. In Fig. 2, we set the linear iteration stopping criterion to be 0.01 and see the performance of the ILU preconditioner, the CPR preconditioner, and the adaptive (first-ILU-then-CPR) preconditioner. We can see that the switching between the ILU preconditioner and the CPR preconditioner does not affect the convergence rate of the linear iterations and the computational efficiency.




(a) $1^{\text{st}}$ Newton iteration

(b) $2^{\text{nd}}$ Newton iteration

(c) $3^{\text{rd}}$ Newton iteration

(d) $4^{\text{th}}$ Newton iteration

**Figure 2—Performance of the preconditioners.**

In summary, for a linear system resulted from the *k*-th nonlinear iteration,

$$A_k x = b_k, \tag{6}$$

we propose the following algorithm [16]:

---

**Adaptive Preconditioning Strategy**

---

1. Set up the preconditioner switch criterion $e^{switch}$.
2. Calculate the absolute residual of the linear system $e_k = ||A_k x - b_k||_2$.
3. Calculate the preconditioner switch indicator $e_k^r = \frac{e_k}{e_1}$, where $e_0$ is the absolute residual of the linear system at the first nonlinear iteration.
4. If $e_k^r > e^{switch}$, use the ILU preconditioner.
    - Monitor the convergence rate of the linear solver.
        - If the convergence rate is satisfied, continue to use the ILU preconditioner;
        - Else, switch the ILU preconditioner to the CPR preconditioner.

   Else, use the CPR preconditioner.

---

## Numerical examples

The SPE9 problem is tested on the cluster Niagara from SciNet, and the black oil SPE10 problem is tested on the cluster Cedar from WestGrid. The computing node of Cedar used in our work has a total of 32 CPU cores (2.1GHz) and 128 GB memory. The computing node of Niagara has a total of 40 CPU cores (2.4 GHz) and 202 GB memory.

We use the mass error as the nonlinear iteration stopping indicator, and it can be written as follows:

$$e_o = \frac{r_o}{\frac{V}{\Delta t}[(\phi \rho_o^o s_o)^{n+1} - (\phi \rho_o^o s_o)^n]} < \epsilon, \tag{6}$$

$$e_w = \frac{r_w}{\frac{V}{\Delta t}[(\phi \rho_w s_w)^{n+1} - (\phi \rho_w s_w)^n]} < \epsilon, \tag{7}$$

$$e_o = \frac{r_g}{\frac{V}{\Delta t}[(\phi \rho_o^g s_o)^{n+1} - (\phi \rho_o^g s_o)^n + (\phi \rho_g s_g)^{n+1} - (\phi \rho_g s_g)^n]} < \epsilon. \tag{8}$$

In Tables 2-6, "Nonlinear" is the total number of nonlinear iterations, "Linear" is the total number of linear iterations, "ILU" is the number of the ILU preconditioner used as the preconditioner, "CPR" is the number of the CPR preconditioner used as the preconditioner, "L. Time" is the linear solver time, and "T. Time" is the total simulation time.

**Table 2—Preconditioner performance, 2 CPU cores, SPE9.**

| PC | $e^{switch}$ | Time steps | Nonlinear | Linear | ILU/CPR | L. time seconds | T. time seconds |
|---|---|---|---|---|---|---|---|
| CPR-FPF | - | 94 | 448 | 771 | - | 38.2 | 49.1 |
| Adaptive | 0.02 | 94 | 443 | 1,515 | 1,417/98 | 32.0 | 42.2 |
| | 0.01 | 94 | 457 | 1,754 | 1,699/55 | 31.8 | 42.3 |
| | 0.005 | 94 | 449 | 1,789 | 1,748/41 | 31.2 | 41.7 |
| | 0.001 | 94 | 458 | 1,872 | 1,872/0 | 31.6 | 42.1 |

**Table 3—Preconditioner performance, 4 CPU cores, SPE9.**

| PC | $e^{switch}$ | Time steps | Nonlinear | Linear | ILU/CPR | L. time seconds | T. time seconds |
|---|---|---|---|---|---|---|---|
| CPR-FPF | - | 94 | 457 | 865 | - | 18.3 | 24.0 |
| Adaptive | 0.02 | 94 | 452 | 1,933 | 1,774/159 | 14.4 | 19.8 |
| | 0.01 | 94 | 455 | 2,025 | 1,935/90 | 14.1 | 19.4 |
| | 0.005 | 94 | 463 | 2,145 | 2,134/11 | 14.0 | 19.2 |
| | 0.001 | 94 | 463 | 2,072 | 2,072/0 | 13.9 | 19.1 |

**Table 4—Preconditioner performance, 8 CPU cores, SPE9.**

| PC | $e^{switch}$ | Time steps | Nonlinear | Linear | ILU/CPR | L. time seconds | T. time seconds |
|---|---|---|---|---|---|---|---|
| CPR-FPF | - | 97 | 534 | 972 | - | 8.7 | 12.8 |
| Adaptive | 0.02 | 94 | 442 | 2,018 | 1,870/148 | 6.6 | 10.3 |
| | 0.01 | 94 | 444 | 1,957 | 1,857/100 | 6.4 | 10.1 |
| | 0.005 | 92 | 382 | 1,969 | 1,907/62 | 6.3 | 9.7 |
| | 0.001 | 94 | 443 | 2,273 | 2,272/1 | 6.2 | 9.9 |

**Table 5—Preconditioner performance, 64 CPU cores, black-oil SPE10.**

| PC | $e^{switch}$ | Time steps | Nonlinear | Linear | ILU/CPR | L. time seconds | T. time seconds |
|---|---|---|---|---|---|---|---|
| CPR-FPF | - | 398 | 1,848 | 26,216 | - | 2340.1 | 3425.3 |
| Adaptive | 0.02 | 398 | 1,799 | 31,876 | 9,018/22,858 | 2108.7 | 3172.2 |
| | 0.01 | 398 | 1,800 | 33,929 | 11,112/22,817 | 2138.3 | 3193.1 |
| | 0.005 | 398 | 1,809 | 35,701 | 12,870/22,831 | 2160.3 | 3229.9 |
| | 0.001 | 398 | 1,848 | 38,267 | 15,939/22,328 | 2179.0 | 3264.9 |

**Table 6—Preconditioner performance, 96 CPU cores, black-oil SPE10.**

| PC | $e^{switch}$ | Time steps | Nonlinear | Linear | ILU/CPR | L. time seconds | T. time seconds |
|---|---|---|---|---|---|---|---|
| CPR-FPF | - | 398 | 1,830 | 25,010 | - | 1593.9 | 2327.0 |
| Adaptive | 0.02 | 398 | 1,788 | 31,993 | 9,330/22,663 | 1473.6 | 2208.6 |
| | 0.01 | 398 | 1,771 | 33,286 | 11,299/21,987 | 1462.3 | 2170.6 |
| | 0.005 | 398 | 1,783 | 35,181 | 13,238/21,943 | 1451.2 | 2147.3 |
| | 0.001 | 398 | 1,766 | 37,247 | 16,055/21,192 | 1453.2 | 2156.8 |

## SPE9 problem

The ninth SPE benchmark [23] has $24 \times 25 \times 15 = 9000$ grid blocks and heterogenous permeability, as shown in Fig. 3. There are 25 production wells and one water injection well. The simulation time is 900 days.

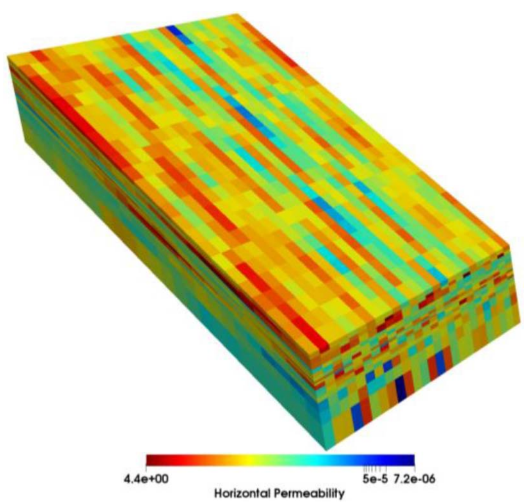



**Figure 3—Horizontal permeability of SPE9 problem.**

For this example, we use 2, 4, and 8 CPU cores to exam the adaptive preconditioning strategy. The numerical results of using different CPU cores have very similar phenomena. If we use the CPR preconditioner, 865 linear iterations are employed in the entire simulation. We set $e^{switch}$ be 0.01, 0.005, and 0.001 to see the numerical performance of the adaptive preconditioner. Since this example is an easy one for linear solvers, we expect that more ILU preconditioners can be used to save the computational time. As we can see from Table 2, the adaptive preconditioning strategy uses the ILU method as the preconditioner in most time. Compared with the CPR preconditioner, the adaptive preconditioner is able to save about 20% time on the linear solvers. We can also see that the number of the ILU preconditioners increases as reducing the value of $e^{switch}$, and when $e^{switch} = 0.001$, all the CPR preconditioners are replaced by the ILU preconditioner. However, increasing the number of the ILU preconditioners does not improve the efficiency of linear solvers, and the difference on computation time is not obvious. The preconditioner switch criterion affects the computation time very little. The number of the linear iterations and the computation cost on the linear solvers reach a balance, and it is difficult to further improve the performance of the preconditioners through the proposed strategy.

**Black-oil SPE10 problem**

The SPE10 benchmark [19] has 60 × 220 × 85 cells and highly heterogeneous geological model; see Fig. 4. There are four producers with fixed bottom-hole pressure and one water injector with a fixed injection rate. The original problem is an oil-water two-phase problem, but we consider a black-oil three-phase problem here. The stopping criterion $\in$ in Eqns. (6)-(8) is 0.01.

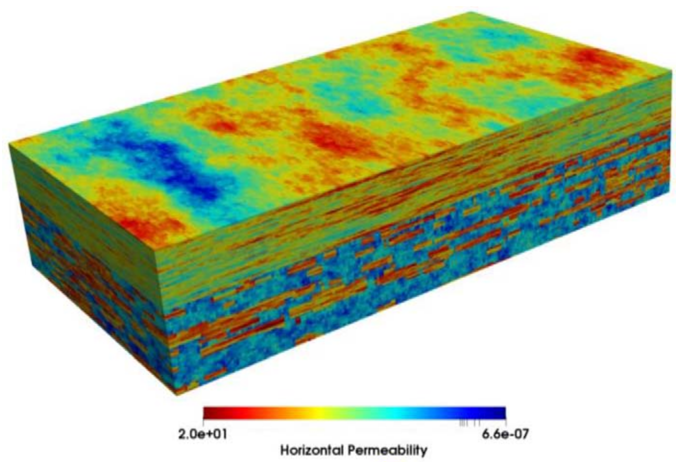



Figure 4—Horizontal permeability of SPE10 problem.

In this example, we test the adaptive preconditioning strategy by using 64 and 96 CPU cores. We first test the CPR preconditioner and use the results as a reference to exam the adaptive preconditioning strategy. In Tables 5-7, we have the numerical performance of the adaptive preconditioner with different switch criteria. To exam the adaptive preconditioning strategy, we also set the preconditioner switch criterion $e^{switch}$ to be 0.02, 0.01, 0.005, and 0.001. This problem is a big challenge for linear solvers due to its highly heterogeneous permeability and the number of grid blocks. What we can expect is that more CPR preconditioners should be used to effectively solve the linear systems. The numerical results in Tables 5 and 6 give the performance of the adaptive preconditioning strategy, and the results agree with what we expect. The CPR preconditioners take a big portion of the total preconditioners, and only 1/6 of the CPR preconditioners are replaced by the ILU preconditioners. Even so, benefiting from the adaptive preconditioning strategy, about 10% of the computational cost on the linear solvers is saved. Comparing the numerical performance of the adaptive strategies with the different preconditioner switch criterion $e^{switch}$, we do not see obvious difference on computation time. Based on the SPE9 problem and the black-oil SPE10 problem, we can conclude that it is safe to select a small value for the preconditioner switch criterion, such as 0.001, in practical application of the adaptive strategy.

## Conclusions

In this paper, the adaptive preconditioning strategy for parallel reservoir simulation is further studied and analyzed. This adaptive preconditioning strategy takes advantage of both an ILU preconditioner and a CPR preconditioner. The ILU preconditioner is preferred for simple problems due to its low computational cost, while the CPR preconditioner is employed for difficult problems because of its effectiveness. When the ILU preconditioner is employed, the convergence rate is monitored. If the convergence rate reaches out of expectation, the preconditioner switches to the CPR preconditioner.

According to the numerical examples, the adaptive preconditioner is able to optimize the utilization of a preconditioner: for the SPE9 problem, most of the preconditioners are the ILU preconditioner, while the CPR preconditioner is preferred for the more difficult problem, SPE10 problem. The computational cost spent on linear solvers can be saved from 10% to 20%. For practical application of the adaptive preconditioning strategy, we can choose a relative small switch criterion $e^{switch}$ (such as 0.001) to achieve good performance of the linear solvers.

## Acknowledgements

The support of Department of Chemical and Petroleum Engineering, University of Calgary and Reservoir Simulation Group is gratefully acknowledged. The research is partly supported by NSERC/Energi Simulation, Alberta Innovates (iCore), IBM Thomas J. Watson Research Center, and the Energi Simulation/ Frank and Sarah Meyer 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).

The supports of Beijing Natural Science Foundation (2184120) and National Program on Key Basic Research Project (Grant 2015CB250901) are also acknowledged.

## References

1. DeBaun, D., Byer, T., Childs, P., Chen, J., Saaf, F., Wells, M., Liu, J., Cao, H., Pianelo, L., Tilakraj, V., et al, 2005. An extensible architecture for next generation scalable parallel reservoir simulation. In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.
2. Dogru, A.H., Sunaidi, H.A., Fung, L.S., Habiballah, W.A., Al-Zamel, N., Li, K.G., 2002. A parallel reservoir simulator for large-scale reservoir simulation. *SPE Reserv. Eval. Eng.* **5** (1), 11e23.
3. Dogru, A.H., Fung, L.S.K., Middya, U., Al-Shaalan, T., Pita, J.A., 2009. A next-generation parallel reservoir simulator for giant reservoirs. In: *SPE/EAGE Reservoir Characterization & Simulation Conference*.
4. Wang, K., Liu H., Chen Z., A scalable parallel black oil simulator on distributed memory parallel computers, *J. Comput. Phys.* **301** (2015) 19–34.
5. Wang, K., Liu, H., Luo, J., Chen, Z. A multi-continuum multi-phase parallel simulator for large-scale conventional and unconventional reservoirs. *J. Nat. Gas Sci. Eng.* 2016, **33**, 483-496.
6. Wallis J.R., Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration, in: *SPE Reservoir Simulation Symposium*, 1983.
7. Behie G.A., P.A.F. Jr., Incomplete factorization methods for fully implicit simulation of enhanced oil recovery, *SIAM J. Sci. Stat. Comput.* **5** (3) (1984) 541–561.
8. Cai X.C., Sarkis M., A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM J. Sci. Comput.* **21** (2) (1999) 792–797.
9. Wallis J.R., Kendall R.P., Little T.E., Constrained residual acceleration of conjugate residual methods, in: *SPE Reservoir Simulation Symposium*, 1985.

10. K. Stüben, A review of algebraic multigrid, *J. Comput. Appl. Math.***128**(1)(2001)281–309.
11. Cao H., Tchelepi H.A., Wallis J.R., Yardumian H.E., Parallel scalable unstructured CPR-type linear solver for reservoir simulation, in: *SPE Annual Technical Conference and Exhibition*, 2005.
12. 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.
13. H. Liu, K. Wang, Z. Chen, A family of constrained pressure residual preconditioners for parallel reservoir simulations, *Numer. Linear Algebra Appl.* (2015) 1–32.
14. 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.
15. 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.
16. Wang, K., Liu, H., Luo, J., Chen, Z. (2018). Efficient CPR-type preconditioner and its adaptive strategies for large-scale parallel reservoir simulations. *Journal of Computational and Applied Mathematics*, **328**, 443–468.
17. Z. Chen, G. Huan, Y. Ma, Computational Methods for Multiphase Flows in Porous Media, Vol.**2**, *Siam*, 2006.
18. Z. Chen, Reservoir Simulation: Mathematical Techniques in Oil Recovery. *CBMS-NSF Regional Conference Series in Applied Mathematics*, Vol. **77**, SIAM, Philadelphia, 2007.
19. M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques., *SPE Reser. Eval. Eng.* **4** (4) (2001) 308–317 4.4 (2001): 308-317.
20. H.D. Sterck, U. Yang, J. Heys, Reducing complexity in parallel algebraic multigrid preconditioners, *SIAM J. Matrix Anal. Appl.* **27**(4) (2006) 1019–1039.
21. H.D. Sterck, R.D. Falgout, J.W. Nolting, U.M. Yang, Distance two interpolation for parallel algebraic multigrid, Numer. *Linear Algebra Appl.* **15** (2–3) (2008) 115–139.
22. M. Adams, M. Brezina, J. Hu, R. Tuminaro, Parallel multigrid smoothing: polynomial versus Gauss Seidel, *J. Comput. Phys.* **182**(2) (2003) 593–610.
23. Killough, J. E. "Ninth SPE comparative solution project: a reexamination of black-oil simulation." In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1995.
24. Z. Chen, BDM mixed methods for a nonlinear elliptic problem, *J. Comp. Appl. Math.* **53** (1994), 207–223.
25. Z. Chen, M. Espedal, and R. E. Ewing, Continuous-time finite element analysis of multiphase flow in groundwater hydrology, *Applications of Mathematics* **40** (1995), 203–226.