SCIENCE CHINA
Mathematics

• ARTICLES •

December 2013 Vol. 56 No. 12: 2737–2751
doi: 10.1007/s11425-013-4737-3

Application of auxiliary space preconditioning in field-scale reservoir simulation

Dedicated to Professor Shi Zhong-Ci on the Occasion of his 80th Birthday

HU XiaoZhe[1], XU JinChao[1,*] & ZHANG ChenSong[2]

[1]*Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA;*
[2]*LSEC and NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China*
Email: hu_x@math.psu.edu, xu@math.psu.edu, zhangcs@lsec.cc.ac.cn

Received June 14, 2013; accepted August 13, 2013; published online October 16, 2013

Abstract We study a class of preconditioners to solve large-scale linear systems arising from fully implicit reservoir simulation. These methods are discussed in the framework of the auxiliary space preconditioning method for generality. Unlike in the case of classical algebraic preconditioning methods, we take several analytical and physical considerations into account. In addition, we choose appropriate auxiliary problems to design the robust solvers herein. More importantly, our methods are user-friendly and general enough to be easily ported to existing petroleum reservoir simulators. We test the efficiency and robustness of the proposed method by applying them to a couple of benchmark problems and real-world reservoir problems. The numerical results show that our methods are both efficient and robust for large reservoir models.

Keywords reservoir simulation, black-oil model, fully implicit method, auxiliary space preconditioning, algebraic multigrid method, Krylov subspace iterative method

MSC(2010) 65N22, 65F08, 65N55

Citation: Hu X Z, Xu J C, Zhang C S. Application of auxiliary space preconditioning in field-scale reservoir simulation. Sci China Math, 2013, 56: 2737–2751, doi: 10.1007/s11425-013-4737-3

1 Introduction

Petroleum reservoir simulation is a tool for predicting hydrocarbon reservoir performance under various operating conditions. It helps petroleum engineers to obtain information pertaining to the processes that take place within oil reservoirs—information that can be used to maximize recovery and minimize environmental damage. The fully implicit method (FIM), also referred to as simultaneous solution (SS) (see [11]), is a natural numerical discretization scheme that is widely-used in academic and commercial reservoir simulators at present. This method provides better stability than other often-used methods, such as implicit pressure explicit saturation (IMPES). However, FIM gives rise to coupled, non-symmetric, ill-conditioned Jacobian systems, which are usually difficult to solve.

The most time-consuming part of FIM is solving the Jacobian system $Ax = b$ at each Newton iteration. Very often, solving such linear systems with direct or iterative solvers takes more than 80% of the computational time in reservoir simulation. Furthermore, ever-increasing demand for more accurate numerical simulation has led to larger and more heterogenous discrete field-scale reservoir models. And

*Corresponding author

such models entail larger and more difficult linear systems. Thanks to their applicability and limited demands on memory, iterative methods are generally preferred over direct solvers for solving large-scale sparse linear systems. In fact, Krylov subspace iterative methods, such as orthogual minimization (ORTHOMIN) method, biconjugate gradient stabilized (BICGSTAB) method, and generalized minimal residual (GMRES) method, are widely employed in reservoir simulation.

Because iterative methods by themselves do not usually converge well for ill-conditioned problems; therefore, researchers have proposed many preconditioners in an effort to accelerate convergence. Over the last 50 years, the incomplete LU factorization (ILU) method, developed in the 1960s [7, 36]—a representative of purely algebraic methods—has become one of the most commonly used preconditioners in reservoir simulation. ILU provides an approximation of the exact LU factorization (computed via Gaussian elimination) by specifying the sparsity of the L and U factors. Other popular purely algebraic preconditioners, often referred to as black-box methods, include the block ILU method [10,24], the nested factorization method [2, 3], and the SVD-reduction method [39]. Due to their simplicity and generality, these methods have gained considerable interest from researchers in the field of reservoir simulation.

The convergence rate of purely algebraic preconditioners or solvers usually deteriorates as the size of the problem increases. More importantly, they are not robust with respect to heterogeneity and anisotropy. An alternative strategy is to construct preconditioners based on decoupling the primary unknowns according to their physical properties. Representative examples using this strategy include the combinative method [5], the constrained pressure residual (CPR) method [37, 38], and multi-stage preconditioners [1, 22, 23, 31, 34]. These methods are suitable for certain reservoir simulation problems and show great potential for field-scale reservoir simulation.

In this paper, we study a class of preconditioners for solving the Jacobian systems arising in FIM using the auxiliary space preconditioning (ASP) framework [41]. In the last few years, ASP has been successfully applied to several partial differential equations (PDEs), including the diffusion-convection-reaction equation, the linear elasticity model, the Stokes problem, the Navier-Stokes equation, complex fluids models, and magnetohydrodynamics; see [42] for a summary of these applications. Hu et al. [16] proposed an auxiliary space preconditioning method for petroleum reservoir simulation and the preliminary numerical results presented therein are very promising.

The ASP framework provides a theoretical framework for designing and analyzing efficient solvers or preconditioners for PDE-based problems: Using the special features and properties of each auxiliary space or problem, ASP constructs preconditioners by constructing efficient approximation to the solution to each auxiliary problem. One such example is the method of subspace corrections [40], where the auxiliary spaces selected are subspaces of the original solution space. For example, the well-known domain decomposition (DD) and multigrid (MG) methods can be analyzed using the method of subspace corrections.

In the ASP framework the auxiliary spaces are generally not restricted to subspaces. This is an important feature because it makes ASP very attractive for complicated real-world applications. In applications with complicated (possibly locally refined) grids, hierarchical grids are usually difficult if even possible to be obtained. Grasedyck et al. [13] developed an $O(N \log N)$ algorithm to construct a hierarchical structured auxiliary grid that has the same local density as the original grid with N vertices. This technique can be used to construct efficient auxiliary space preconditioners for practical problems such as the pressure equation in the reservoir simulation. However, for the sake of simplicity, we will choose auxiliary spaces to be the subspaces of the solution space in this paper.

Choosing appropriate auxiliary spaces, auxiliary problems, and their approximation schemes is the key to achieving satisfactory overall performance of auxiliary space preconditioning. We use the physical heuristics and mathematical properties of the black-oil model and its modifications to choose appropriate auxiliary problems:

• we choose auxiliary problems that decouple different primary unknowns without destroying the physical and mathematical properties;

• we apply efficient and robust solvers and/or smoothers for each auxiliary problem according to those properties;

- we use an algebraic method to solve the auxiliary problem for implicit wells; and
- we handle the strong coupling between different unknowns with an extra smoothing step.

We perform several numerical experiments, including million-cell black-oil test problems, to test the effectiveness and robustness of our method. To test the simulation results and speed of the proposed algorithm, we conduct preliminary comparison with benchmark results and simulation results produced by Schlumberger's[1)] Eclipse100 software. Preliminary comparisons on accuracy and efficiency between the proposed method and solvers commonly used in commercial software are reported. And, the numerical results indicate that the proposed method is efficient and robust.

The rest of the paper is organized as follows: The standard black-oil model and its modified version for polymer-flooding are reviewed in Section 2. In Section 3, we introduce the auxiliary space preconditioning framework, and based on this technique, we propose a preconditioner for the fully-implicit discretizations for the black-oil model. In Section 4, we discuss the details of the implementation of this preconditioner. Numerical results which show the effectiveness and robustness of our methods are presented in Section 5, and concluding remarks are given in Section 6.

2 Mathematical model

In this section, we will briefly review the mathematical formulation for multiphase flow in porous media; for details, please refer to Chen et al. [8]. Here, we focus on the classical black-oil model and its modified version for modeling polymer flooding. The nomenclature used in this section is listed at the end of the paper.

2.1 Black-oil model

The black-oil model is developed based on the assumptions: (1) the reservoir is isothermal; (2) the flow in the porous media has three phases (oil, gas, and water) and three components (oil, gas, and water); (3) mass transfer occurs between the oil and the gas phases; and (4) no mass transfer occurs between the water phase and either of the other two phases. The mass conservation equations of the black-oil model can be written as follows:

$$\frac{\partial}{\partial t}\left(\phi\frac{S_w}{B_w}\right) = -\nabla\cdot\left(\frac{1}{B_w}u_w\right) + \frac{Q_{Ws}}{B_w}, \tag{2.1}$$

$$\frac{\partial}{\partial t}\left(\phi\frac{S_o}{B_o}\right) = -\nabla\cdot\left(\frac{1}{B_o}u_o\right) + \frac{Q_{Os}}{B_o}, \tag{2.2}$$

$$\frac{\partial}{\partial t}\left[\phi\left(\frac{S_g}{B_g}+\frac{R_sS_o}{B_o}\right)\right] = -\nabla\cdot\left(\frac{1}{B_g}u_g+\frac{R_s}{B_o}u_o\right) + \frac{Q_{Gs}}{B_g}+\frac{R_sQ_{Os}}{B_o}, \tag{2.3}$$

where

$$u_\alpha = -\frac{kk_{r\alpha}}{\mu_\alpha}(\nabla P_\alpha - \rho_\alpha \mathfrak{g}\nabla z), \quad \alpha = o, g, w, \tag{2.4}$$

$$S_o + S_g + S_w = 1. \tag{2.5}$$

Among these equations, (2.5) represents the phase saturation balance. Note that for the sake of simplicity, we assume that the capillary pressures between the phases are always equal to zero in (2.4), i.e., $P_w = P_o = P_g$ and we use P to denote them.

We use the Peaceman model [28] for the volumetric well rates at the standard condition, i.e.,

$$Q_{Ws} = \sum_{j=1}^{N_w}\sum_{m=1}^{M_wj}\frac{2\pi\Delta L^{(j,m)}}{\ln(r_e^{(j,m)}/r_w^j)}\frac{kk_{rw}}{\mu_w}[P_{\rm bh}^{(j)} - P - \rho_w\mathfrak{g}(z_{\rm bh}^{(j)} - z)]\delta(x - x^{(j,m)}), \tag{2.6}$$

[1)] http://www.slb.com

$$Q_{Os} = \sum_{j=1}^{N_w} \sum_{m=1}^{M_w j} \frac{2\pi \Delta L^{(j,m)}}{\ln(r_e^{(j,m)}/r_w^j)} \frac{k k_{ro}}{\mu_o} [P_{\rm bh}^{(j)} - P - \rho_o \mathfrak{g}(z_{\rm bh}^{(j)} - z)] \delta(x - x^{(j,m)}), \tag{2.7}$$

$$Q_{Gs} = \sum_{j=1}^{N_w} \sum_{m=1}^{M_w j} \frac{2\pi \Delta L^{(j,m)}}{\ln(r_e^{(j,m)}/r_w^j)} \frac{k k_{rg}}{\mu_g} [P_{\rm bh}^{(j)} - P - \rho_g \mathfrak{g}(z_{\rm bh}^{(j)} - z)] \delta(x - x^{(j,m)}). \tag{2.8}$$

Here, if the wells are treated implicitly, the bottom hole pressures $P_{bh}^{(j)}$ are unknowns that need to be solved.

See the nomenclature at the end of the paper for the notations used in the above equations and the rest of this paper.

2.2 Modified black-oil model

The importance of improving oil/gas recovery rate lies in the fact that to a significant extent, China relies on oil (and gas) to meet its ever-growing need for energy. Furthermore, most of China's oil fields are located in continental basins and many are characterized by serious heterogeneity, low permeability, and high oil viscosity. Hence, we also consider a modified black-oil model, a variant of the classical black-oil model designed to model the polymer flooding in enhanced oil recovery (EOR) or improved oil recovery (IOR).

To this end, we add two more components, polymer and sodium chloride. The modified black-oil model for this case is based on the following assumptions: (1) the porous media is compressible, (2) the flow in porous media has three phases (oil, gas, and water) and five components (oil, gas, water, polymer, and sodium chloride), (3) there is thermal degradation in the polymer, (4) the decline of the residual oil as an effect of viscoelastic fluids is ignored, and (5) polymer and sodium chloride only exist in the water phase and there is no mass transfer between the water phase and either of the other two phases.

Based on these assumptions, the modified black-oil model consists of (2.1)–(2.3) from the black-oil model and the following two extra equations which govern the polymer and sodium chloride in the water phase:

$$\frac{\partial}{\partial t}\left(\phi \frac{S_w^* C_p}{B_w}\right) = -\nabla \cdot \left(\frac{C_p}{B_w} u_w\right) + \frac{Q_{Ws}}{B_w} C_p, \tag{2.9}$$

$$\frac{\partial}{\partial t}\left(\phi \frac{S_w C_s}{B_w}\right) = -\nabla \cdot \left(\frac{C_s}{B_w} u_w\right) + \frac{Q_{Ws}}{B_w} C_s, \tag{2.10}$$

where

$$u_w = -\frac{k k_{rw}}{\mu_{w,\rm eff} R_k} (\nabla P_w - \rho_w \mathfrak{g} \nabla z), \tag{2.11}$$

$$S_w^* = S_w - S_{\rm dpv}. \tag{2.12}$$

As mentioned above, we assume that the capillary pressures between phases are equal to zero. Hence $P_w = P$. More details about the effects of the polymer and brine will be given in the numerical experiments when needed.

2.3 Jacobian system

Among the many possible discretization methods for the above model, we consider only the fully implicit method (FIM) [11] in which the Newton linearization is combined with first-order upstream-weighting finite difference spatial discretization; for details, see [8, Chapter 8]. Basically, we eliminate S_g from (2.1)–(2.4) using (2.5) and plug (2.4) into (2.1)–(2.3). Moreover, we choose the main solution variables to be the increments δP, δS_w, and δS_o and give the rest of the variables in terms of these main solution variables.

In each Newton step, we need to solve a nonsymmetric and indefinite Jacobian system that has the following two-by-two block[2)] form:

[2)] We denote the solution variable as $x_{\rm R} := [\delta P, \delta S_w, \delta S_o]^t$ and $x_{\rm W} := \delta P_{bh}$.

$$Ax = b, \quad \text{i.e.,} \quad \begin{bmatrix} A_{\mathrm{RR}} & A_{\mathrm{RW}} \\ A_{\mathrm{WR}} & A_{\mathrm{WW}} \end{bmatrix} \begin{bmatrix} x_{\mathrm{R}} \\ x_{\mathrm{W}} \end{bmatrix} = \begin{bmatrix} b_{\mathrm{R}} \\ b_{\mathrm{W}} \end{bmatrix}, \tag{2.13}$$

where A_{RR} is the derivatives of the reservoir equations with respect to the reservoir variables, A_{RW} is the derivatives of the reservoir equations with respective to the well variables, A_{WR} is the derivatives of the well equations with respect to the reservoir variables, and A_{WW} is the derivative of the well equations with respect to the well variables.

Furthermore, if we divide the reservoir variables into the pressure variable part and the non-pressure variable part (saturations or concentrations), we can write the reservoir block A_{RR} as a two-by-two block matrix:

$$A_{\mathrm{RR}} = \begin{bmatrix} A_{\mathrm{PP}} & A_{\mathrm{PS}} \\ A_{\mathrm{SP}} & A_{\mathrm{SS}} \end{bmatrix}, \tag{2.14}$$

where A_{PP} corresponds to the derivatives with respect to the pressure unknowns, A_{SS} corresponds to the derivatives with respect to the saturation/concentration unknowns, and A_{PS} and A_{SP} are the blocks that couple the pressure and non-pressure variables.

In such cases, the linear system of equations (2.13) is often large, sparse, nonsymmetric, and indefinite. Different blocks have different properties due to the analytic nature of their corresponding continuous differential equations. The equation corresponding to the pressure unknowns is elliptic, the equation corresponding to the saturation unknowns is hyperbolic, and the equation corresponding to the well unknowns is algebraic. All these facts make the resulting Jacobian systems very challenging to solve. Moreover, the complicated geometric and physical properties of the reservoir, such as the anisotropy and heterogeneity, make the Jacobian systems more difficult to solve by transitional methods. In this paper, we solve (2.13) using the preconditioned GMRES method [30] with an efficient and robust preconditioning strategy based on auxiliary space preconditioning.

3 Auxiliary space preconditioning

In this section, we will briefly recall the auxiliary space preconditioning framework introduced in [41], which was developed based on the fictitious spaces preconditioning techniques by Nepomnyaschikh [26].

3.1 An abstract theoretical framework for SPD systems

Consider a symmetric positive definite (SPD) system $Ax = b$ on a vector space V equipped with an inner product $a(\cdot,\cdot) = (A\cdot,\cdot)$. The auxiliary space preconditioning method constructs a preconditioner by using approximate solvers on the following product of auxiliary spaces:

$$\overline{V} = V \times W_1 \times \cdots \times W_J\ , \tag{3.1}$$

where $W_1, \ldots, W_J$ are auxiliary (Hilbert) spaces endowed with inner products $\bar{a}_j(\cdot,\cdot) = (\cdot,\cdot)_{A_j}$, $j = 1, \ldots, J$. With appropriate transformation operators $\Pi_j : W_j \mapsto V$, we can define an additive preconditioner as follows:

$$B = S + \sum_{j=1}^{J} \Pi_j A_j^{-1} \Pi_j^t. \tag{3.2}$$

A distinctive feature of the auxiliary space method is the presence of V in (3.1) as a component of $\bar{V}$ together with the presence of the operator $S : V \mapsto V$, which is usually called the smoother. It can be proved that the preconditioner (3.2) admits the following estimate:

Theorem 3.1 (See [41, Theorem 2.1]). *Assume that*

$$(\Pi_j w_j, \Pi_j w_j)_A \leqslant c_j^2 (w_j, w_j)_{A_j}, \quad \forall\, w_j \in W_j, \tag{3.3}$$

$$(SAv, v)_A \leqslant c_s^2 (v, v)_A, \quad \forall\, v \in V, \tag{3.4}$$

and, for each $v \in V$, there are $v_0 \in V$ and $w_j \in W_j$ such that

$$v = v_0 + \sum_{j=1}^{J} \Pi_j w_j \quad \textit{and} \quad (SAv_0, v_0)_A + \sum_{j=1}^{J} (w_j, w_j)_{A_j} \leqslant c_0^2 (v, v)_A.$$

Then the condition number of the preconditioned system is estimated as follows:

$$\kappa(BA) \leqslant c_0^2 (c_s^2 + c_1^2 + \cdots + c_J^2). \tag{3.5}$$

Naturally, we can give a multiplicative version of (3.2) and obtain the following iterative method:

$$x \leftarrow x + S(b - Ax) \quad \text{and} \quad x \leftarrow x + \Pi_j A_j^{-1} \Pi_j^t (b - Ax), \quad j = 1 : J. \tag{3.6}$$

Under the same assumptions of Theorem 3.1, we have the following theory (see [15]).

Remark 3.2 (See [15, Remark 1]). Under the same assumptions as those in Theorem 3.1, the convergence rate of (3.6) depends only on c_0, c_s, c_j $(1 \leqslant j \leqslant J)$, and J.

Remark 3.3. The method of subspace corrections (MSC) is a special case of the auxiliary space preconditioning technique, in which W_j is a subspace of V. Another special case is the fictitious space lemma [26] developed by Nepomnyashikh, which differs from the auxiliary space preconditioning techniques in that it does not use the smoother S. The auxiliary space preconditioning method has been applied to many other problems, such as the Maxwell equation, the $H(\mathrm{div})$ problem [15], and fourth-order elliptic problems [43]. We refer to [42] for a comprehensive introduction to the auxiliary space preconditioning framework. The preconditioner (3.2) introduced in this section can be viewed as an additive version preconditioner, and its multiplicative variant can be found in [17].

3.2 Auxiliary space preconditioning for the fully implicit method

Now, we discuss how to use the auxiliary space preconditioning technique to solve (2.13) arising from the discretization of the (modified) black-oil model. As stated earlier, different blocks in the Jacobian system have different properties and the couplings between different unknowns are strong. In order to design a robust and efficient preconditioner, we should take full advantage of all the analytical, mathematical, and physical properties of the (modified) black-oil model. Our preconditioner consists of three key steps:

- apply simple decoupling strategies to weaken the coupling between different unknowns;
- transform the complicated Jacobian system into several simpler auxiliary problems and design robust and efficient solvers for each; and
- combine all the auxiliary problems together with a smoother for the original Jacobian system in order to define a preconditioner.

Next, we discuss these key steps in the preconditioning algorithm in detail.

3.2.1 *Decoupling*

The decoupling technique is a preprocessing step designed to weaken the coupling between different unknowns. It is usually applied to the reservoir block A_{RR} to decouple the pressure unknowns from the non-pressure (saturation/concentration) unknowns. This technique has been studied by many researchers (e.g., see [1, 4, 21, 23]) successfully. We apply the decoupling technique to the entire Jacobian system A using what is referred to as the alternative block factorization (ABF) strategy introduced by Bank et al. [4]. Solve the following preprocessed equation:

$$\tilde{A}_{\mathrm{RR}} x_{\mathrm{R}} = \tilde{b}_{\mathrm{R}}, \quad \text{with} \quad \tilde{A}_{\mathrm{RR}} = D_{\mathrm{RR}}^{-1} A_{\mathrm{RR}} \ \text{ and } \ \tilde{b}_{\mathrm{R}} = D_{\mathrm{RR}}^{-1} b_{\mathrm{R}}, \tag{3.7}$$

where

$$D = \begin{bmatrix} \mathrm{diag}(A_{\mathrm{PP}}) & \mathrm{diag}(A_{\mathrm{PS}}) \\ \mathrm{diag}(A_{\mathrm{SP}}) & \mathrm{diag}(A_{\mathrm{SS}}) \end{bmatrix}. \tag{3.8}$$

This decoupling strategy is effective in closeting the eigenvalues of matrix $\tilde{A}$, which can, in turn, reduce the convergence rate of GMRES. Moreover, Klie [21] also noted that $D_{\mathrm{RR}}^{-1} A_{\mathrm{RR}}$ is preferred in practice over $A_{\mathrm{RR}} D_{\mathrm{RR}}^{-1}$, because the latter might spoil the diagonal-dominance of A_{RR}.

Remark 3.4. There are many other options for decoupling, such as Householder transformations, the implicit pressure explicit saturation (IMPES)-type method, and the balanced decoupling strategy (BSD) method based on the least square method. Their performance and comparison can be found in [1,21,23]. For the present study, we chose the ABF approach due to its simplicity and reasonable decoupling effects. Investigating efficient and robust decoupling strategies beyond the scope of this paper.

Remark 3.5. In practice, we construct the preprocessed linear system $\tilde{A}$ explicitly and then apply the preconditioned GMRES method on $\tilde{A}$. Therefore, we need to compute the inverse of the decoupling matrix D. This is done by reordering D according to the cells, which will lead to a block diagonal matrix in which each block on the diagonal is a (3×3)-matrix (black-oil model) or a (5×5)-matrix (modified black-oil model).

Note that, in the rest of this section, we apply our preconditioner on $\tilde{A}$ instead of A. For the sake of simplicity, we use A to denote the preprocessed linear system $D^{-1}A$. The same convention applies to b.

3.2.2 *A multiplicative preconditioner*

The primary unknowns that we are solving in the (modified) black-oil model are the pressure, saturation/concentration, and bottom hole pressure (BHP) of the implicit wells. Therefore, it is natural to introduce auxiliary problems for each primary unknown, and to use their properties to construct an auxiliary space preconditioner. Assume that we have already been given the auxiliary problems A_{P}, A_{S}, and A_{W} for the pressure, saturation/concentration, and BHP of the implicit wells, respectively, and the corresponding transfer operators Π_{P}, Π_{S}, and Π_{W}, together with a smoother S for A. We will discuss how to construct these auxiliary problems and their solvers in the next section, Section 4.

Now, an additive version of the auxiliary space preconditioner can be defined as

$$B_a := S + \Pi_{\mathrm{P}} A_{\mathrm{P}}^{-1} \Pi_{\mathrm{P}}^t + \Pi_{\mathrm{S}} A_{\mathrm{S}}^{-1} \Pi_{\mathrm{S}}^t + \Pi_{\mathrm{W}} A_{\mathrm{W}}^{-1} \Pi_{\mathrm{W}}^t.$$

In practice, instead of computing the inverse operators exactly, we use approximate solvers for each auxiliary problem, i.e., A_{P}^{-1}, A_{S}^{-1}, and A_{W}^{-1} are approximated by B_{P}, B_{S}, and B_{W}, respectively. This leads to the following additive preconditioner, which is still denoted by B_a,

$$B_a := S + \Pi_{\mathrm{P}} B_{\mathrm{P}} \Pi_{\mathrm{P}}^t + \Pi_{\mathrm{S}} B_{\mathrm{S}} \Pi_{\mathrm{S}}^t + \Pi_{\mathrm{W}} B_{\mathrm{W}} \Pi_{\mathrm{W}}^t. \tag{3.9}$$

Moreover, in order to achieve better efficiency and robustness, we use the following multiplicative version of the preconditioner (Algorithm 1) in practice.

It is easy to see that Algorithm 1 defines a preconditioner B_m such that

$$I - B_m A = (I - SA)(I - \Pi_{\mathrm{P}} B_{\mathrm{P}} \Pi_{\mathrm{P}}^t A)(I - \Pi_{\mathrm{S}} B_{\mathrm{S}} \Pi_{\mathrm{S}}^t A)(I - \Pi_{\mathrm{W}} B_{\mathrm{W}} \Pi_{\mathrm{W}}^t A), \tag{3.10}$$

and in this paper, we use B_m as a preconditioner for the GMRES method to solve the Jacobian system (3.7).

4 Construction of auxiliary problems

The choice of auxiliary problems is crucial to the overall performance of the preconditioner B_m in Algorithm 1. We want the auxiliary problems to preserve the properties of the governing equations so that

Algorithm 1 A multiplicative preconditioner for the (modified) black-oil model

Given an initial guess x
$x \leftarrow x + \Pi_{\mathrm{W}} B_{\mathrm{W}} \Pi_{\mathrm{W}}^t (b - Ax)$,
$x \leftarrow x + \Pi_{\mathrm{S}} B_{\mathrm{S}} \Pi_{\mathrm{S}}^t (b - Ax)$,
$x \leftarrow x + \Pi_{\mathrm{P}} B_{\mathrm{P}} \Pi_{\mathrm{P}}^t (b - Ax)$,
$x \leftarrow x + S(b - Ax)$.

it will be possible to construct an efficient solver for each auxiliary problem. It is worthy to point out that, in this paper, we construct auxiliary problems algebraically for simplicity. In practice, we have more freedom to adjust these auxiliary problems.

4.1 Implicit well constraint auxiliary problem

Implicit wells and the reservoir are usually strongly coupled together, and this strong interaction may cause the traditional linear solvers to converge slowly or even to fail to converge [25]. The Peaceman equation, basically an algebraic equation, describes the constraint on the flow rate. Its non-PDE character suggests that the implicit well constraints should be treated separately. However, the strong coupling suggests that they should be handled together, and the coupled strategy is preferred in general. We choose A_{W} such that both facts are taken into account. First of all, the A_{WW} part is included in A_{W} and decoupled from the A_{RR} part. Secondly, notice that the coupling between the reservoir and the implicit wells happens at the perforated cells; therefore, the choice of A_{W} should also involve those perforated cells.

More precisely, we can rearrange the Jacobian system in (2.13) as follows:

$$A \Longrightarrow \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & \bar{A}_{\mathrm{RW}} \\ 0 & \bar{A}_{\mathrm{WR}} & A_{\mathrm{WW}} \end{bmatrix}, \tag{4.1}$$

where A_{11} is derivative of the reservoir equations on the imperforated grids with respect to the reservoir variables on the imperforated grids, A_{12} is the derivative of the reservoir equations on the imperforated grids with respect to the reservoir variables on the perforated grid, A_{21} is the derivative of the reservoir equations on the perforated grid with respect to the reservoir variables on the imperforated grid, and A_{22} is the derivative of the reservoir equations on the perforated grids with respect to the reservoir variables on the perforated grid. Now we choose the auxiliary problem

$$A_{\mathrm{W}} = \begin{bmatrix} A_{22} & \bar{A}_{\mathrm{RW}} \\ \bar{A}_{\mathrm{WR}} & A_{\mathrm{WW}} \end{bmatrix}. \tag{4.2}$$

As an example, we show a 3×3 grid (see Figure 1(a)) with an injection well at the center (perforated at cells 2 and 5). The Jacobian matrix A is a bordered system (in grey); see Figure 1(b). After the reordering, we get a new matrix as in Figure 1(c).

Remark 4.1. Note that, the coupling between the reservoir and the well equations is fully covered in A_{W}. In general, the size of A_{W} is relatively small, such that we can solve it exactly by a direct method, i.e., $B_{\mathrm{W}} = A_{\mathrm{W}}^{-1}$.

Remark 4.2. In practice, on the cells that are close to the perforated cells, the coupling between the reservoir and the wells may also be strong. We can, therefore, choose A_{W} such that they are also included. This can be done in the same way as above.

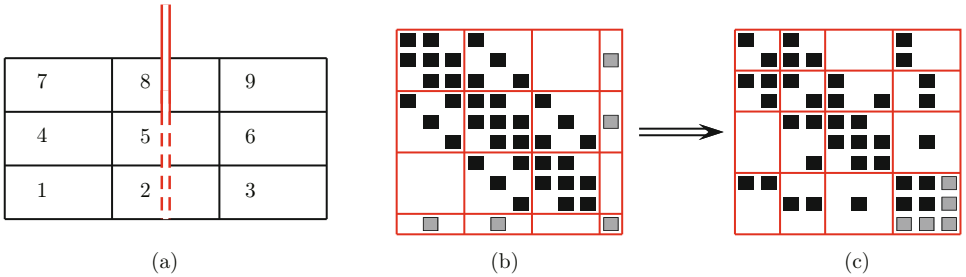

Figure 1 Reordering for well variables and perforated cells

4.2 Saturation/concentration auxiliary problem

For the auxiliary problem for saturation/concentration, we set $A_{\mathrm{S}} = A_{\mathrm{SS}}$. As stated earlier, the equation describing the saturation unknown is mainly hyperbolic. Moreover, the upstream-weighting technique [18] is used in our discretization. For such problems, the block Gauss-Seidel (BGS) method with downwind ordering and crosswind blocks is an efficient method; see [19, 20] for details. Therefore, we adopt the method here and use it to approximate A_{S}^{-1}. In general, such downwind ordering can be obtained by the Tarjan algorithm [14, 35]. Note the fact that the multiphase flow in the porous media flows from high-pressure cells to low-pressure cells. For convenience, we can simply obtain the downwind ordering by ordering the pressure unknowns. Theoretically, after reordering, A_{S} is close to a lower triangular matrix. Hence, the Gauss-Seidel method can be applied to solve such kind of matrices efficiently.

Remark 4.3. In some models proposed for polymer flooding, the diffusion effect of the polymer is also considered. When this diffusion effect is relatively strong, we can adjust the saturation (or concentration) smoother to account for the elliptic nature of the auxiliary problem.

4.3 Pressure auxiliary problem

It is known that A_{P} preserves the ellipticity of the pressure equation, and we expect it to be possible to apply efficient solvers like the algebraic multigrid (AMG) method [6, 12, 29, 32, 33] to define B_{P}. Based on this expectation, we choose completely decoupled auxiliary problems in which the coupling is handled by the smoother S. One such example is the well-known constrained pressure residual (CPR) preconditioner [37].

We can include the well unknowns (bottom hole pressure for rate-constrained wells) to construct A_{P} as well, i.e.,

$$A_{\mathrm{P}} = \begin{bmatrix} A_{\mathrm{PP}} & A_{\mathrm{PW}} \\ A_{\mathrm{WP}} & A_{\mathrm{WW}} \end{bmatrix}.$$

Clearly, this choice helps resolve the coupling between the wells and the reservoir. Usually, B_{P} is an approximate inverse of A_{P} achieved via a few AMG cycles. In practice, the performance of the AMG method may degenerate due to the anisotropy and heterogeneity of the reservoir. Moreover, the preprocessing step (3.7) and the extra well unknowns may destroy the elliptic property of this auxiliary problem. In this case, we can apply the combined preconditioner technique introduced in [17], in which the AMG method and the ILU preconditioner are combined to define an effective solver B_{P} for A_{P}.

Remark 4.4. The CPR method, originally proposed in [37], has been applied in many petroleum reservoir simulators. It can be presented for the reservoir block A_{RR} in the following form:

$$B_{\mathrm{CPR}} := S(I - A_{\mathrm{RR}}M) + M, \tag{4.3}$$

where S is usually line successive over-relaxation (LSOR) or Incomplete LU factorization and where

$$M = \begin{bmatrix} B_{\mathrm{P}} & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{with } B_{\mathrm{P}} \approx A_{\mathrm{PP}}^{-1}.$$

Remark 4.5. There are several multi-stage methods, including the CPR-type preconditioners, which are defined by a different choice of M. Other possible choices of M are

$$\begin{bmatrix} B_{\mathrm{P}} & 0 \\ 0 & B_{\mathrm{S}} \end{bmatrix}, \quad \begin{bmatrix} B_{\mathrm{P}}^{-1} & 0 \\ A_{\mathrm{SP}} & B_{\mathrm{S}}^{-1} \end{bmatrix}^{-1} \quad \text{or} \quad \begin{bmatrix} B_{\mathrm{P}}^{-1} & A_{\mathrm{PS}} \\ 0 & B_{\mathrm{S}}^{-1} \end{bmatrix}^{-1},$$

where $B_{\mathrm{P}} \approx A_{\mathrm{PP}}^{-1}$ and $B_{\mathrm{S}} \approx A_{\mathrm{SS}}^{-1}$.

Remark 4.6. In the CPR-type methods, the coupling is handled by S besides the standard techniques discussed in Subsection 3.2.1. This choice may not be effective or robust, especially for strongly coupled

problems. The coupling between the reservoir and the wells is usually strong, suggesting that we should choose coupled auxiliary problems yet maintain the property of each unknown. It is on this basis that it is possible for us to design a good solver for each auxiliary problem.

4.4 Smoother for the coupled system

While the CPR preconditioner and its variants are chosen such that the auxiliary problems are completely decoupled, the smoother S handles the coupling and must, therefore, be strong. Hence, it is necessary to use an expensive smoother like LSOR or ILU. However, we choose coupled auxiliary problems, which allows us to use computationally cheaper smoothers. In this paper, we simply employ the block Gauss-Seidel method as S and apply downwind ordering and crosswind blocks to improve the method's performance without introducing much extra work (we use the same downwind ordering and crosswind blocks as those used for solving A_{S}).

The auxiliary problems and the smoother are combined as shown in Algorithm 1, which defines the auxiliary space preconditioner B_m for the (modified) black-oil model. B_m then serves as a preconditioner for the GMRES method. In the next section, we will present several numerical examples to demonstrate the effectiveness and robustness of the preconditioner B_m.

5 Numerical experiments

In this section, we describe several numerical experiments and analyze the performance of the proposed auxiliary space preconditioner B_m. All the experiments were performed on a desktop with Intel i7 CPU and 8GB DDR3 memory. The code was compiled using Microsoft Visual Studio 2008 and Intel C/C++ compiler 12.0.1.

The developed auxiliary space preconditioning method has already been implemented in the fast auxiliary space preconditioning (FASP) solver package[3]. Therefore, in this section, we will just call it FASP for simplicity. For testing purposes, we plugged the FASP solver into the Monix simulator[4], which can solve reservoir problems involving up to five components (three pseudo-hydrocarbon components, inert gas, and water) flowing in two or three dimensions in a single- and/or dual-porosity mode by the fully implicit method. At each Newton-Raphson iteration, the GMRES method with the auxiliary space preconditioner was applied in order to solve the resulting Jacobian matrix.

We compare the simulation results of Monix with the industry-standard commercial software, Eclipse100 (2009 version). The Eclipse family of reservoir simulation software offers the industry most complete and robust set of numerical solutions to fast and accurate prediction of dynamic behavior, for all types of reservoirs and degrees of complexity—structure, geology, fluids, and development schemes.

5.1 Modified SPE1 benchmark

The first test problem is based on the SPE1[5] (Case B[6]) comparative test [27]. The permeability and the porosity are homogenous. The porosity is 0.3 and the permeability is set to $k_x = k_y = 500$ mD and $k_z = 100$ mD. There is one gas injector and one producer, which are located at two opposite corners. The total simulation time is 10 years. We use this test problem to study the algorithmic scalability of the proposed auxiliary space preconditioning method for the black-oil model. For this purpose, we refine the original grid in order to obtain a sequence of different structured grids—their sizes are $20 \times 20 \times 10$, $80 \times 80 \times 20$, $100 \times 100 \times 20$, $160 \times 160 \times 20$, and $200 \times 200 \times 30$. Note that although the grid size decreases as we refine it, this does not lead to higher heterogeneity. In real applications, higher grid resolution usually leads to higher heterogeneity and hence to more difficult Jacobian systems to solve.

[3] http://fasp.sourceforge.net

[4] Monix is a reservoir simulator developed by Monix energy solutions, Inc. http://www.monixe.com

[5] Society of petroleum engineers (SPE)

[6] Variable bubble-point pressure case

Table 1 Performance of the FASP preconditioner for the refined SPE1 benchmark (Case B)

Problem size	# Newton steps	# Linear iterations	Total wall time	Linear solver time	Average linear solver time
$20 \times 20 \times 10$	333	4,820	91 s	88 s	0.25 s
$80 \times 80 \times 20$	1,262	11,411	2,246 s	1,887 s	1.50 s
$100 \times 100 \times 20$	1,409	12,677	3,537 s	2,821 s	2.00 s
$160 \times 160 \times 20$	2,223	19,275	13,067 s	9,826 s	4.42 s
$200 \times 200 \times 30$	2,372	19,195	29,260 s	22,140 s	9.33 s

We designed this test problem to test only the scalability of our preconditioner. In Table 1, we report the total number of Newton steps, the total number of iterations of the linear solver, the total wall time, the total linear solver time, and the average linear solver time for each Newton iteration. From this table, we can see that the CPU time for solving one Jacobian matrix is proportional to the size of the test problem. The table shows that the FASP method has optimal computational complexity for this black-oil test problem and is potentially scalable.

5.2 SPE10 benchmark

This test problem uses the second dataset from the Tenth SPE Comparative Solution Project [9], which was designed to compare the ability of upscaling approaches used by various participants to predict the performance of a water-flooding in a simple but highly heterogeneous black-oil reservoir described by a fine-scale ($60 \times 220 \times 85$) regular Cartesian geological model. The model has a simple geometry, with no top structure or faults. The model dimensions are $1200 \times 2200 \times 170$ (ft). The top 70 ft (35 layers) represents the Tarbert formation, and the bottom 100 ft (50 layers) represents Upper Ness. There is one injector in the center of the field and four producers, one at each of the four corners. The total simulation time is 2,000 days.

The problem statement specified that the intent and the basis for competition was to compare the accuracy of the solution. We compared our numerical results with the benchmark results from Landmark, Geoquest, Chevron, and Streamsim; see Figure 2. We find that our results are in good agreement with the reported results based on other simulators. The total wall time for a single simulation run for the SPE10 problem (1.1M grid cells) using Monix with the proposed preconditioner is less than 40 minutes. The CPU times for the linear solvers and other parts of the Monix simulator are reported in Table 2. We note that, if using Eclipse100, non converging linear solutions will force the simulator to use extremely small time steps, which results in long turnaround time (more than 120 hours).

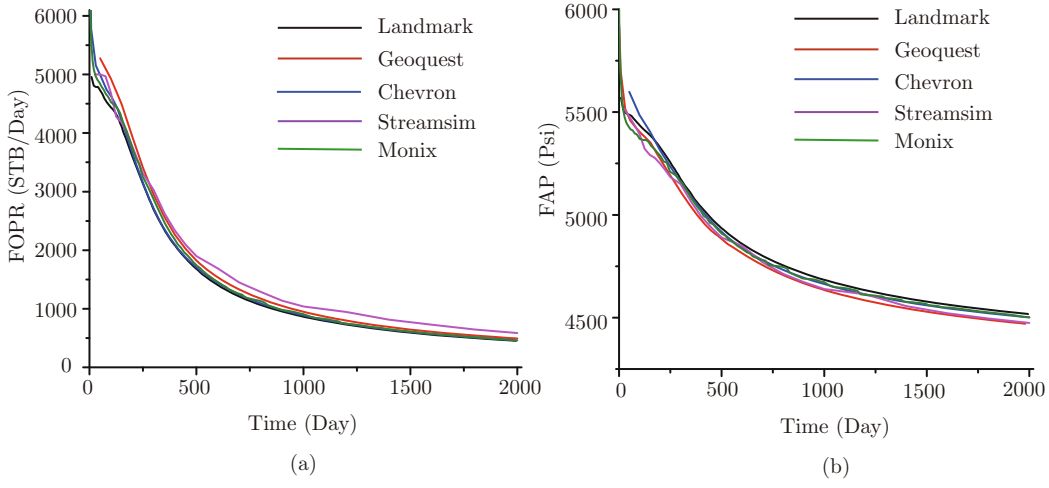

Figure 2 Comparisons of simulation results using different simulators. (a) Field oil production rate (FOPR); (b) Field average pressure (FAP)

Table 2 The CPU time cost (in minutes) of each component in simulating Model 2 in SPE10. The linear solver (FASP) counts for 78% of the total wall time and the time cost of each linear solver is 5.9 seconds (9.9 linear iteration) on average

Components	Wall time	Comments
Reading input	1.5	
Initialization	1.5	Including initializing and scheduling
Linear solver	29.0	295 Newton iterations in total
Others	5.0	Including assembling, time-stepping, logging, ...
Total	37.0	

Table 3 The CPU time for the entire field simulation and the average wall time for each Newton iteration

Simulator	# Time steps	# Newton iter.	Wall time	Avg. Wall time
Eclipse100	606	2136	136 minutes	3.8 seconds
Monix	1425	6012	87 minutes	0.87 seconds

5.3 Black-oil field test

This test example is based on a field problem in eastern part of China. The reservoir is characterized by a complex fault block-oil reservoir model with six small fault blocks. The problem has very complicated oil–water interfaces. The numerical simulation begins with a non-equilibrium initial state and has multiple pressure-volume-temperature relation (PVT) and relative permeability partitions. Due to the lack of natural energy, the development process mainly relies on water injection. The simulation history is 30 years with a total of 242 wells during peak time of oil development. We use a $71 \times 140 \times 42$ corner-point grid with several fault planes. The resolution on the horizontal direction is about 30 meters. A summary of the numerical results is given in Table 3. In particular, the proposed preconditioner is very robust for this field test problem.

5.4 Polymer-flooding field test

This test problem is a $157 \times 53 \times 57$ (corner-point grid) five-component modified black-oil model for polymer-flooding (see Section 2) geological model based on a structural-lithological reservoir. The geological model, which is based on an offshore reservoir in China, has complicated top structural and several faults[7]. In addition, more than 80% of the grid blocks are inactive. There are 37 working wells in peak, and the simulation period lasts 10 years. In the first seven years, all the injectors inject water; after that, eight start to inject some polymer solution and continue to do so for the next three years. The non-Newtonian effect of polymer-flooding is modeled by non-constant effective viscosity

$$\mu_{w,\text{eff}} := \mu_w[1 + (\beta(C_p, C_b) - 1)M(u_w)],$$

where

$$\beta(C_p, C_b) > 1 \quad \text{and} \quad 0 \leqslant M(u_w) \leqslant 1$$

are the given functions[8]. Using this real field test, we investigate the robustness of the proposed preconditioners for both the black-oil model (three primary variables in each grid cell) and the modified black-oil model (five primary variables in each grid cell), water flooding and polymer flooding, complex well conditions, geological faults, etc. A summary of the numerical results is given in Table 4; the proposed preconditioner is very robust for this field test problem. The linear solver (FASP) counts for 67% of the total wall time in Monix and the time cost of each linear solver is 0.58 seconds (5.7 linear iteration) on average.

[7] Geological faults are modeled using non-neighboring connections.

[8] Their explicit expressions are determined by lab experiments and omitted here.

Table 4 The CPU time for the entire field simulation and the average wall time for each Newton iteration

Simulator	# Time steps	# Newton iter.	Wall time	Avg. Wall time
Eclipse100	628	1562	81 minutes	3.11 seconds
Monix	728	2650	38 minutes	0.86 seconds

6 Conclusions

We applied the auxiliary space preconditioning framework to field-scale reservoir simulation and proposed an effective and robust preconditioner for solving large-scale Jacobian systems arising from a fully implicit discretization scheme. It is crucial to choose the auxiliary problems according to the mathematical and physical properties of the unknowns:

• The pressure equation is mostly elliptic in nature, and AMG methods are efficient for solving this auxiliary problem;

• The ordered block Gauss-Seidel (GS) method is used as an efficient smoother for the saturation (and/or concentration) auxiliary problem;

• Direct solvers can be applied to solve the well constraints together with the flows at the perforated cells due to their small size.

Most importantly, we have to take into account the coupling between different variables. And, as our choice of auxiliary problems take the coupling into account, we can use a relatively cheap global smoother, such as the ordered block GS smother.

We tested the efficiency and robustness of the new method by applying it to several numerical examples, including million-cell black-oil models and a polymer-flooding model. The numerical results show that our method is quite robust and also has good scalability, both of which are important for field-scale reservoir simulation.

Acknowledgements This work was supported by Petro-China Joint Research Funding (Grant No. 12HT105 0002654), National Science Foundation of USA (Grant No. DMS-1217142), the Dean's Startup Fund, Academy of Mathematics and System Sciences and the State High Tech Development Plan of China (863 Program) (Grant No. 2012AA01A309). The authors would like to thank RIPED, Petro. China, for providing data for the numerical tests.

References

1 Al-Shaalan T, Klie H, Dogru A, et al. Studies of robust two stage preconditioners for the solution of fully implicit multiphase flow problems. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 2009, 1–12

2 Appleyard J, Cheshire I M. Nested factorization. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 1983, 315–324

3 Appleyard J, Cheshire I, Pollard R. Special techniques for fully implicit simulators. In: Proceedings of the European Symposium on Enhanced Oil Recovery. Harwell: Atomic Energy Research Establishment, Computer Science and Systems Division, 1981, 395–408

4 Bank R E, Chan T F, Coughran Jr W M, et al. The alternate-block-factorization procedure for systems of partial differential equations. BIT, 1989, 29: 938–954

5 Behie A, Vinsome P. Block iterative methods for fully implicit reservoir simulation. Old SPE J, 1982, 22: 658–668

6 Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. In: Sparsity and its Applications. Cambridge: Cambridge University Press, 1985, 257–284

7 Buleev N I. A numerical method for the solution of two-dimensional and three-dimensional equations of diffusion. Mat Sb N S, 1960, 51: 227–238

8 Chen Z, Huan G, Ma Y. Computational Methods for Multiphase Flows in Porous Media. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006

9 Christie M, Blunt M. Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE Reservoir Evaluation Eng, 2001, 4: 308–317

10 Concus P, Golub G, Meurant G. Block preconditioning for the conjugate gradient method. SIAM J Sci Statist Comput, 1985, 6: 220–252

11 Douglas Jr J, Peaceman D W, Rachford D. A method for calculating multi-dimensional displacement. Trans Amer Inst Min Metallurgical Petroleum Eng, 1959, 216: 297–306

12 Falgout R. An Introduction to Algebraic Multigrid. Comput Sci Eng, 2006, 8: 24–33

13 Grasedyck L, Wang L, Xu J. A nearly optimal multigrid method for general unstructured grids. Technical report, preprint, 2012

14 Gustavson F. Finding the block lower triangular form of a sparse matrix. In: Sparse matrix computations. New York: Academic Press, 1976, 275–289

15 Hiptmair R, Xu J. Nodal auxiliary space preconditioning in $H(\mathrm{curl})$ and $H(\mathrm{div})$ spaces. SIAM J Numer Anal, 2007, 45: 2483–2509

16 Hu X, Liu W, Qin G, et al. Development of a fast auxiliary subspace preconditioner for numerical reservoir simulators. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 2011, 1–10

17 Hu X, Wu S, Wu X H, et al. Combined preconditioning with applications in reservoir simulation. Multiscale Model Simul, 2013, arXiv:1205.6113v3

18 Huyakorn P, Nilkuha K. Solution of transient transport equation using an upstream finite element scheme. Appl Math Model, 1979, 3: 7–17

19 Kim H, Xu J, Zikatanov L. A multigrid method based on graph matching for convection-diffusion equations. Numer Linear Algebra Appl, 2003, 10: 181–195

20 Kim H, Xu J, Zikatanov L. Uniformly convergent multigrid methods for convection-diffusion problems without any constraint on coarse grids. Adv Comput Math, 2004, 20: 385–399

21 Klie H. Krylovsecant methods for solving large-scale systems of coupled nonlinear parabolic equations. PhD thesis. Houston, TX: Rice University, 1997

22 Lacroix S, Vassilevski Y, Wheeler J, et al. Iterative solution methods for modeling multiphase flow in porous media fully implicitly. SIAM J Sci Comput, 2003, 25: 905–926

23 Lacroix S, Vassilevski Y V, Wheeler M F. Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS). Numer Linear Algebra Appl, 2001, 8: 537–549

24 Meyerink J. Iterative methods for the solution of linear equations based on incomplete block factorization of the matrix. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 1983, 1–8

25 Mishev I, Shaw J, Lu P. Numerical experiments with AMG solver in reservoir simulation. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 2011, 1–8

26 Nepomnyaschikh S V. Decomposition and fictitious domains methods for elliptic boundary value problems. In: Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations. Philadelphia, PA: SIAM, 1992, 62–72

27 Odeh A S. Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. J Petroleum Technol, 1981, 33: 13–25

28 Peaceman D W. Reproseutation of a horizontal well in numerical reservoir simulation. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 1991, 7–16

29 Ruge J W, Stüben K. Algebraic multigrid. In: Multigrid methods, Frontiers Appl Math, vol. 3. Philadelphia, PA: SIAM, 1987, 73–130

30 Saad Y, Schultz M H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Statist Comput, 1986, 7: 856–869

31 Scheichl R, Masson R, Wendebourg J. Decoupling and block preconditioning for sedimentary basin simulations. Comput Geosci, 2003, 7: 295–318

32 Stüben K. An introduction to algebraic multigrid. In: Trottenberg U, Oosterlee C W, Schüller A, eds. Multigrid. London: Academic Press, 2001, 413–532

33 Stüben K. Solving Reservoir Simulation Equations. In: 9th International Forum on Reservoir Simulation. Abu Dhabi: UAE, 2007, 1–53

34 Stüben K, Clees T, Klie H, et al. Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 2007, 1–11

35 Tarjan R E. Depth-first search and linear graph algorithms. SIAM J Comput, 1972, 1: 146–160

36 Varga R S. Matrix Iterative Analysis. Englewood Cliffs, NJ: Prentice-Hall Inc., 1962

37 Wallis J. Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 1983, 1–10

38 Wallis J, Kendall R, Little T. Constrained residual acceleration of conjugate residual methods. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 1985, 1–14

39 Watts J, Shaw J. A new method for solving the implicit reservoir simulation matrix equation. In: SPE Reservoir Simulation Symposium. Oklahoma: Society of Petroleum Engineers, 2005, 1–7

40 Xu J. Iterative methods by space decomposition and subspace correction. SIAM Rev, 1992, 34: 581–613
41 Xu J. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. Computing, 1996, 56: 215–235
42 Xu J. Fast Poisson-based solvers for linear and nonlinear PDEs. In: Bhatia R, ed. Proceedings of the International Congress of Mathematics, vol. 4. Singapore: World Scientific, 2010, 2886–2912
43 Zhang S, Xu J. Optimal solvers for fourth-order PDEs discretized on unstructured grids. SIAM J Numer Anal, 2012, submitted

Nomenclature

- ϕ: Porosity
- k: Permeability
- S_α: Saturation of the α-phase
- $S_{\rm dpv}$: Percentage of inaccessible pore volume for polymer
- S_w^*: Saturation of the polymer
- μ_α: Viscosity of the α-phase
- $\mu_{\gamma,\rm eff}$: Effective viscosity for the γ-phase/component
- P_α: Pressure of the α-phase
- $P_{\rm cow}, P_{\rm cgo}$: Capillary pressure
- u_γ: Velocity of the γ-phase/component
- B_α: Formation volume factor of the α-phase
- $k_{r\alpha}$: Relative permeability of the α-phase
- ρ_α: Mass density of the α-phase
- ρ_r: Mass density of rock
- R_s: Gas solubility
- R_v: Oil volatility
- R_k: Relative permeability-reduction factor
- $C_{\rm ab}$: Percentage of polymer adsorption
- C_p, C_s: Concentration of polymer and sodium chloride in the water phase.
- $\tilde{q}_\gamma$: Source/sink for the γ-phase/component
- $Q_{\beta s}$: Volumetric flow rates in standard conditions for the β-component
- $\mathfrak{g}$: Gravitational acceleration
- z: Depth
- Subscript s: Indicator for standard conditions
- $\delta(x)$: Dirac delta function
- N_w: Total number of wells
- $M_{w,j}$: Total number of perforated zones of the j-th well
- $\Delta L^{(j,m)}$: Segment length of the m-th perforated zone of the j-th well
- $x^{(j,m)}$: Central location of the m-th perforated zone of the j-th well
- $r_w^{(j)}$: Well bore radius of the j-th well
- $r_e^{(j,m)}$: Drainage radius of the j-th well at the gridlock in which $x^{(j,m)}$ is located
- $p_{\rm bh}^{(j)}$: Bottom hole pressure of the j-th well at the well datum $z_{\rm bh}^{(j)}$