LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-PROC-708970

Multigrid reduction for coupled flow problems with application to reservoir simulation

L. Wang, D. Osei-Kuffuor, R. Falgout, I. Mishev, J. Li

November 11, 2016

SPE reservoir simulation Conference
Montgomery, TX, United States
February 20, 2017 through February 22, 2017

Disclaimer

SPE Reservoir Simulation Conference 2017

SPE-182723-MS

Multigrid reduction for coupled flow problems with application to reservoir simulation

Lu Wang, Lawrence Livermore National Laboratory, Daniel Osei-Kuffuor, Lawrence Livermore National Laboratory, Rob Falgout, Lawrence Livermore National Laboratory, Ilya Mishev, ExxonMobil, and Jizhou Li, ExxonMobil

Abstract

Applications in geosciences, such as reservoir modeling, continue to grow in both size and complexity. Simulations are increasingly complex as they couple more physical phenomena over larger physical domains. As a result, the linear system that arises from the numerical solution of these problems can be challenging to solve by iterative methods. Simulators require advanced algebraic solvers that are robust enough to handle the anisotropies, heterogeneities and coupling between the physical variables, and scalable enough to handle large-scale solution on high performance parallel systems. Multigrid solvers are a class of iterative solvers that are scalable and efficient for solving linear systems that arise from Large-scale applications. However, applications with multiple physical unknowns pose a challenge for standard multigrid techniques, particularly when the coupling between the unknowns is strong. In this paper, we present our efforts to develop a multigrid- preconditioned Krylov solver, where the preconditioner is based on the multigrid reduction framework. This preconditioner is designed to represent the coupling between the physical variables of the reservoir modeling equations and account for the underlying physics of the system. Two-stage preconditioners, such as the well-known constrained pressure residual (CPR) approach and its variants like CPR-AMG, have been commonly used in reservoir simulation applications. We discuss how these current solver strategies may be interpreted within the multigrid reduction framework to better understand the different variations. Finally, we present techniques for improving the MGR approach and present results on solver performance on examples from reservoir modeling.

Introduction

Linear systems which originate from the equations of reservoir modeling are among the most difficult to solve by iterative methods. These linear systems typically represent coupling between multiple physical unknowns at different scales and are non-symmetric and very ill-conditioned. This makes it difficult to extract effective preconditioners that adequately account for the underlying physics of the problem. Solving these linear systems consumes most of the computational time in all modern reservoir simulators. Furthermore, the demand for more accurate simulations have led to larger and more complex reservoir models. This makes the linear systems even more difficult to solve by direct or iterative methods.

In the past, incomplete LU factorization (ILU) methods have been used for solving the equations of reservoir simulations, primarily due to their robust nature (Watts, 1981; Behie & Vinsome, 1982; Appleyard & Cheshire, 1983). These methods provide an approximation of the exact LU factorization by introducing sparsity in the L and U factors. However, as simulations cover larger and larger domains and are deployed over high performance parallel architectures, there is an apparent need for robust solvers that scale and the use of standard (single-level) ILU methods become less favorable.

In recent years, two-stage preconditioners have been widely used within the reservoir modeling community. See for instance (Wallis *et al.* , 1985; Scheichl *et al.* , 2003; Lacroix *et al.* , 2001, 2003; Zhou *et al.* , 2013; Al-Shaalan *et al.* , 2009; Liu *et al.* , 2015; Hu *et al.* , 2011, 2013b,a) and references therein. In this approach, the solution process for the coupled linear system is split into two main stages, based on the properties of the physical variables as prescribed by the dynamics of the underlying physics. This splitting enables an effective preconditioner to be constructed for each stage, allowing for a balance between robustness and scalability of the linear solver. It is worth noting that adaptations of the two-stage approach to deal with additional variables or contributions such as well equations have been explored in the context of a multi-scale approach (Zhou *et al.* , 2013; Liu *et al.* , 2015). The constrained pressure residual

(CPR) method (Wallis, 1983; Wallis *et al.* , 1985) is one variant of the two-stage preconditioning strategy. In this approach, the first stage consists of solving a reduced linear system for the variables that typically control the flow, such as the pressure variables. This linear system can be poorly conditioned, making it challenging to solve. As a result, an effective preconditioner is needed for this first stage. The global solution is then updated with the solution from the first stage. In the second stage of the CPR method, the solution for the remaining unknowns, describing the saturation or mole fraction related variables, are obtained by solving the linear system on the global problem. Assuming the first stage preconditioner was effective enough, a cheaper preconditioner may be used for the second stage solve to achieve an efficient global solution strategy. The effectiveness of the CPR method has led to various adaptations of the approach. For instance, it is common practice to perform an additional preconditioning step known as decoupling before or after the first stage in order to improve the performance and convergence of the global solution strategy. Decoupling after the first stage transforms the global problem so that cheap yet effective preconditioners can be used for an efficient solution of the second stage linear system (Zhou *et al.* , 2013). Decoupling prior to the first stage transforms the global system so that reduced linear system has algebraic properties that favor the construction of an efficient preconditioner for the first stage linear system. Since the convergence of the CPR method depends on the convergence of the first stage linear system solve, recent work in the area has focused on the development of effective and efficient preconditioners for the first stage linear system. Techniques such as the recent CPR-MS method (Cusini *et al.* , 2015) use multiscale finite volume or finite element methods to solve the reduced linear system. A more common alternative is the CPR-AMG method (Clees & Ganzer, 2007; Lacroix *et al.* , 2001, 2003; Stueben *et al.* , 2007; Al-Shaalan *et al.* , 2009; Scheichl *et al.* , 2003; Tchelepi & Jiang, 2009a; Dubois *et al.* , 2009; Gries *et al.* , 2013), which uses algebraic multigrid (AMG) solvers as preconditioners for the reduced linear system, in order to benefit from the robustness and scalability of AMG. Here, the work has focused primarily on constructing a reduced system such that the coefficient matrix has elliptic or M-matrix properties that are favorable for AMG, while yielding good convergence for the overall CPR strategy (Liu *et al.* , 2015; Gries *et al.* , 2013).

In this paper, we present a general multigrid framework for constructing two-stage preconditioners, based on the multigrid reduction method (MGR) (Ries *et al.* , 1983). Multigrid reduction techniques have been around for many years and can be considered as a generalization of the CPR method in a standard multigrid framework. Adaptations of the reduction approach based on ILU strategies include the ARMS (Saad & Suchomel, 2002) and Multigraph (Bank & Smith, 2002) algorithms. One benefit of this generalization of two-stage preconditioners within the MGR framework is that we can derive a closed form of the error propagator for the approach, which is based on standard multigrid components such as the restriction, prolongation, and coarse grid operators. This enables us to show the effect of the different multigrid components on the convergence of the linear solver. Furthermore, the MGR framework provides a theoretical foundation to better understand and develop improved strategies, which can lead to more effective and scalable two-stage preconditioners.

The paper is organized as follows: In the Model Formulation section, we introduce the reservoir simulation equations and briefly discuss properties of the resulting linear system. We also describe the two-stage solver framework based on CPR. In the Multigrid Reduction section, we present the MGR approach and show how existing two-stage preconditioners may be represented in the general MGR framework. We also present different strategies for improving MGR for reservoir models. The Numerical Results section presents some numerical results on applications from standard industry tests like SPE 10 as well as simulation models from real fields. We conclude in section Discussion and Conclusion.

Model Formulation

Fully-implicit reservoir modeling equations We follow a general formulation of the fully-implicit modeling equations, in which each phase consists of several components. The components are transported within phases and exchanged across phase boundaries. In what follows, we present the equations that describe such phenomena in the isothermal case under the assumption of local thermodynamic equilibrium. Here, we assume N_c components are distributed between N_p phases that are in thermodynamic equilibrium. The unknowns are the mole fractions n_i, $i = 1,\ldots,N_c$ and a reference pressure P. Consequently, we have that

$$\sum_{i=1}^{N_c} n_i = 1 \tag{1}$$

The mass conservation equation for component i in volume V is given by:

$$\frac{\partial}{\partial t}\int_V W_i + \int_{\partial V} \boldsymbol{U}_i \cdot \boldsymbol{n} = \int_V R_i, \;\; i = 1,2,\cdots,N_c \tag{2}$$

where W_i is the total concentration of component i, $\boldsymbol{U}_i$ is the total flow rate of component i, and R_i represents the sources and sinks, which include productions and reactions. The total quantities in the above equation can be formally expressed as follows:

$$W_i = \phi\xi_T n_i, \;\; \text{where} \;\; \xi_T = \sum_{j=1}^{N_p} \xi_j S_j \;\; \text{for} \;\; i = 1,2,\cdots,N_c; \tag{3}$$

and

$$\boldsymbol{U}_i = \sum_{j=1}^{N_p} n_{ij}\xi_j \boldsymbol{v}_j, \;\; i = 1,2,\cdots,N_c \tag{4}$$

where ϕ is the porosity defined as pore volume per bulk volume; ξ_j is the molar density of phase j; S_j is the saturation of phase j defined as the volume of phase j per pore volume; n_{ij} is the mole fraction of component i in phase j; and $\boldsymbol{v}_j$ is the volumetric flow rate of phase j. Therefore, the mass conservation equation can be written with respect to the components as:

$$\frac{\partial}{\partial t}\int_V \phi\xi_T n_i dV + \sum_{j=1}^{N_p}\int_{\partial V} n_{ij}\xi_j \boldsymbol{v}_j \cdot \boldsymbol{n} dA = \bar{Q}_i, \;\; i = 1,2,\ldots,N_c. \tag{5}$$

The quantity $\boldsymbol{v}_j$ is given by multiphase Darcy's law

$$\boldsymbol{v}_j = -k\lambda_i \nabla\Phi_j, \;\; \nabla\Phi_j = \nabla(P + P_{cap,j}) - \rho_j g \nabla z, \;\; j = 1,2,\ldots,N_p, \tag{6}$$

where k is the absolute permeability, λ_i is the mobility of phase j, $P_{cap,j}$ is the capillary pressure of phase j, ρ_j is the mass density of phase j, g is the gravity constant, z is the vertical coordinate direction. In addition to (5), we note that the equation related to the pressure primary variable is given by (1).

To complete the set of equations we shall restrict ourselves to the isothermal setting and local thermodynamic equilibrium. This means that the flow is slow enough that the partitioning of a component i across the phases can be determined by thermodynamic equilibrium considerations. Without going into the details, this yields algebraic expressions of the form

$$\frac{n_{ij}}{n_{ik}} = Z_{ijk}(T, P, n_{ij}, n_{ik}) \tag{7}$$

for each phase $j \neq k$, and where T is the temperature and Z represents some algebraic function. See (Allen *et al.* , 1992; Falta, 1992; Helmig, 1997; Bastian, 1999) for a more detailed treatment of the thermodynamics.

Discretization The equations are discretized by the finite volume method, with the two-point flux approximation scheme, on an unstructured mesh. On a grid block V_m, the nonlinear equations satisfy:

$$\Psi_i = \left[(\phi\xi_T n_i)^{n+1} - (\phi\xi_T n_i)^n\right] m(V_m) - \Delta t \sum_{\ell\in\Pi(V_m)} T_{m\ell}\left[\sum_{j=1}^{N_p} n_{ij}\xi_j\lambda_j\Delta\Phi_j\right]_{m\ell}^{n+1} - \bar{Q}_i m(V_m), \quad i-1,2,\cdots,N_c \tag{8}$$

where $\Pi(V_m)$ is the set of the indices of the neighbors of V_m; $m(A_m)$ and $m(A_{m_\ell})$ are the volume of the grid block V_m and the area of the face $A_{m\ell}$ between V_m and V_ℓ correspondingly; and $T_{m\ell}$ is the transmissibility of connection ℓ given by:

$$T_{m\ell} = \frac{m(A_{m\ell})}{dist(x_m, x_\ell)}\frac{2k_m k_\ell}{k_m + k_\ell}.$$

Here, x_m and x_ℓ are the cell centers of the cells V_m and V_ℓ; and k_m and k_ℓ are the absolute permeabilities in these cells.

The nonlinear system is solved by the fully implicit method with Newton-Raphson linearization. Recall that the equations corresponding to the pressure unknowns come from (1). The constraint in (1) suggests that the discrete system will have only $N_c - 1$ independent component unknowns. Thus, we can reduce the system by eliminating one of the component unknowns, yielding a consistent set of pressure equations and a non-defective linearized system. Using Ψ_i' to denote the reduced from of the discretized equations (Ψ_i' in (8)), the resulting Jacobian linear system that needs to be solved at each Newton step is

$$A\delta x = b. \tag{9}$$

Here, A is the Jacobian matrix, given by

$$A = \frac{\partial(\Psi_1', \cdots, \Psi_{N_c}')}{\partial(P, n_1, \cdots, n_{N_c-1})}; \tag{10}$$

$b = (\Psi_P, \Psi_1, \cdots, \Psi_{N_c})^T$ is the right-hand side; and $\delta x = (\delta(P, n_1, \cdots, n_{N_c}))^T$ is the Newton update solution.

Let N represent the total number of cells, and define $N_c' = N_c - 1$. Then the matrix A can be written in block form as:

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix}, \tag{11}$$

where each block A_{ij} corresponds to a matrix prescribed by the equations within one cell, and has the following structure

$$A_{ij} = \begin{pmatrix} a_{pp}^{ij} & a_{p1}^{ij} & a_{p2}^{ij} & \cdots & a_{pN_c'}^{ij} \\ a_{1p}^{ij} & a_{11}^{ij} & a_{12}^{ij} & \cdots & a_{1N_c'}^{ij} \\ a_{2p}^{ij} & a_{21}^{ij} & a_{22}^{ij} & \cdots & a_{2N_c'}^{ij} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N_c'p}^{ij} & a_{N_c'1}^{ij} & a_{N_c'2}^{ij} & \cdots & a_{N_c'N_c'}^{ij} \end{pmatrix} \tag{12}$$

Reordering the system by the unknowns, we have

$$A = \begin{pmatrix} A_{pp} & A_{p1} & A_{p2} & \cdots & A_{pN_c'} \\ A_{1p} & A_{11} & A_{12} & \cdots & A_{1N_c'} \\ A_{2p} & A_{21} & A_{22} & \cdots & A_{2N_c'} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{N_c'p} & A_{N_c'1} & A_{N_c'2} & \cdots & A_{N_c'N_c'} \end{pmatrix} = \begin{pmatrix} A_{pp} & A_{pn} \\ A_{np} & A_{nn} \end{pmatrix}. \tag{13}$$

We note that in this form, each submatrix A_{ip} represents separate discretized elliptic equations.

Two-stage Preconditioners To solve the linear system (9) from the reservoir simulations, regular Krylov subspace solvers are employed. In order to speed up these solvers, in practice, the following equivalent linear system is solved instead:

$$M^{-1}Ax = M^{-1}b, \tag{14}$$

where M is a left preconditioner.

A proper preconditioner can reduce the condition number of matrix $M^{-1}A$, which accelerates the convergence of the Krylov solver. For reservoir simulation, the ILU(k) and the ILU(p, tol) are widely applied methods, since they are simple to implement and effective on problems with moderate complexity. However, when the problem scale is large or the simulation models are more complex, more stable and robust preconditioners are needed.

Wallis *et al.* (1985) developed the constrained pressure residual (CPR) method as a preconditioner to speed up the convergence of the linear system solver. This method has been adopted by many researchers and commercial simulators, and various adaptations to the approach have been developed. The CPR preconditioner is a two-stage preconditioner that begins by first solving for the pressure variables from a reduced linear system corresponding to the A_{pp} block. The pressure solution is then used as an approximate solution to the global problem, which is then solved by the restricted additive Schwarz (RAS) method at the second stage. In the classical preconditioner developed by Wallis *et al.* (1985), the pressure problem is solved by the ILU method. Recently, algebraic multigrid (AMG) methods have been preferred to solve the pressure problem due to their scalability properties and effectiveness on elliptic problems. In practice, M_s is chosen as ILU or block SOR methods. To be efficient on parallel computers, the second stage preconditioner is prescribed by a RAS or block Jacobi method with an ILU solver for the subdomain problem on each processor. Details of the RAS method can be read in Cai & Sarkis (1999). The two-stage preconditioner has been demonstrated to be the most efficient preconditioner for reservoir simulation Cao (2002); Cao *et al.* (2005). For a detailed description of the two-stage methods, see Tchelepi & Jiang (2009b); Al-Shaalan *et al.* (2009).

Multigrid

Multigrid methods are so-called *scalable* or *optimal* methods because they can solve a linear system with N unknowns with only $O(N)$ work. This property gives them the potential to solve larger problems on proportionally larger parallel machines in constant time. Multigrid methods achieve this optimality by employing two complementary processes: *smoothing* and *coarse-grid correction*. In the classical setting of scalar elliptic problems, the smoother (or relaxation method) is a simple iterative algorithm like Gauss-Seidel that is effective at reducing high-frequency error. The remaining low-frequency error is then accurately represented and efficiently eliminated on coarser grids via the coarse-grid correction step. Applying this simple multigrid idea to get a scalable method can be challenging for some classes of problems, however. The "right" choice of smoother and intergrid transfer operators (restriction and interpolation) is not straightforward in general. This is especially true for systems of PDEs such as those that arise in reservoir simulation. When designed properly, a multigrid solver is algorithmically scalable, because it will uniformly damp all error frequencies with a computational cost that depends only linearly on the problem size.

Multigrid Reduction

Multigrid Reduction (MGR) is a type of multigrid method that is derived by making certain approximations to an underlying direct method called total reduction to yield an efficient iterative solver (Ries *et al.* , 1983). MGR and its variants have been shown to be

effective for Poisson-like equations and anisotropic problems. They have also been used to do non-intrusive parallel time integration (Falgout *et al.* , 2014). To describe the MGR method, we start by decomposing the unknowns into two sets conventionally known as C-points and F-points. In the context of this paper, the C-points are the pressure unknowns (denoted by 'p') and the F-points are the mole fraction unknowns (denoted by 'n'). This gives the following two-by-two matrix decomposition:

$$A = \begin{pmatrix} A_{pp} & A_{pn} \\ A_{np} & A_{nn} \end{pmatrix} = \begin{pmatrix} I_p & A_{pn}A_{nn}^{-1} \\ 0 & I_n \end{pmatrix} \begin{pmatrix} A_{pp} - A_{pn}A_{nn}^{-1}A_{np} & 0 \\ 0 & A_{nn} \end{pmatrix} \begin{pmatrix} I_p & 0 \\ A_{nn}^{-1}A_{np} & I_n \end{pmatrix}, \tag{15}$$

where I_s and I_p are identity operators. We define the so-called ideal restriction R_*, ideal prolongation P_*, and injection operator S by

$$R_* = \begin{pmatrix} I_p & -A_{pn}A_{nn}^{-1} \end{pmatrix}, \quad P_* = \begin{pmatrix} I_p \\ -A_{nn}^{-1}A_{np} \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ I_n \end{pmatrix}. \tag{16}$$

Then, since $A_{nn} = S^T AS$ and $A_{pp} - A_{pn}A_{nn}^{-1}A_{np} = R_*AP_*$, it is easy to see that

$$A^{-1} = P_*(R_*AP_*)^{-1}R_* + S(S^TAS)^{-1}S^T$$

and

$$0 = I - A^{-1}A = I - P_*(R_*AP_*)^{-1}R_*A - S(S^TAS)^{-1}S^TA \tag{17}$$

$$= (I - P_*(R_*AP_*)^{-1}R_*A)(I - S(S^TAS)^{-1}S^TA) \tag{18}$$

$$= (I - S(S^TAS)^{-1}S^TA)(I - P_*(R_*AP_*)^{-1}R_*A), \tag{19}$$

where the equivalence occurs since $R_*AS = S^TAP_* = 0$. This identity defines the two-level multigrid method with the ideal Petrov-Galerkin coarse-grid operator R_*AP_* and the F-relaxation $S(S^TAS)^{-1}S^T$. Equation (17) is the additive MGR identity and (18) and (19) are multiplicative identities with pre-smoothing and post-smoothing. In this paper, we only consider the multiplicative variants of the algorithm. In practice, MGR methods replace ideal restriction and prolongation with approximations R and P respectively, where

$$R = \begin{pmatrix} I_p & W_R \end{pmatrix}, \quad P = \begin{pmatrix} I_p \\ W_P \end{pmatrix}, \tag{20}$$

$W_R \approx -A_{pn}A_{nn}^{-1}$, and $W_P \approx -A_{nn}^{-1}A_{np}$. Consequently, RAP is an approximation to the Schur complement R_*AP_*:

$$RAP = A_{pp} + W_RA_{np} + A_{pn}W_P + W_RA_{nn}W_P \tag{21}$$

$$\approx A_{pp} - A_{pn}A_{nn}^{-1}A_{np}.$$

The F-relaxation in (19) is also generally replaced with a more efficient method and often extended to all unknowns, not just F-points.

In general, we define the MGR operator in either pre-smoothing or post-smoothing form by

$$I - M_{MGR}^{-1}A = (I - PM_C^{-1}RA)(I - M_S^{-1}A), \tag{22}$$

$$I - M_{MGR}^{-1}A = (I - M_S^{-1}A)(I - PM_C^{-1}RA), \tag{23}$$

where $M_C^{-1} \approx (RAP)^{-1}$ is the coarse grid correction and M_S^{-1} is the smoother. The smoother can be (approximate) F-relaxation, such that $M_S^{-1} = SM_n^{-1}S^T$ with $M_n^{-1} \approx (S^TAS)^{-1}$. It can also be extended to a global fine-grid relaxation method such as Jacobi, Gauss-Seidel, ILU, etc. Previous (optimal) variants of MGR replace F-relaxation with FCF-relaxation and apply the two-level method recursively to achieve a multilevel method (Foerster *et al.* , 1981; Ries *et al.* , 1983; Parter, 1987; MacLachlan *et al.* , 2006; Falgout *et al.* , 2014). Other methods in the literature that are closely related to MGR, such as ARMS (Saad & Suchomel, 2002; Li *et al.* , 2003) and multigraph (Bank & Smith, 2002, 1999), often use ILU factorizations to approximate A_{nn} and the Schur complement.

Interpretation of preconditioners in MGR MGR is a general framework. The two-stage preconditioner can be rewritten as a special case of MGR. In particular, consider the case

$$\text{CPR:} \quad \begin{cases} W_R = -\text{blockdiag}(A_{pn})\ \text{blockdiag}(A_{nn})^{-1}, \\ W_P = 0, \end{cases} \tag{24}$$

where $\text{blockdiag}(A_{nn})$ is the matrix of nonzeros of A_{nn} that are on the main block diagonal of A (i.e., only nonzero couplings between unknowns in the same grid cell) and similarly for $\text{blockdiag}(A_{pn})$. With this, the MGR method in (23) is the same as the quasi-IMPES

variant of CPR (Lacroix *et al.* , 2003; Hammersley & Ponting, 2008). Here, by (21) and (24), the coarse-grid (pressure) system M_C approximates the Schur complement as follows:

$$M_C^{(CPR)} = A_{pp} - \text{blockdiag}(A_{pn})\ \text{blockdiag}(A_{nn})^{-1} A_{np}. \tag{25}$$

Note that an alternative approach for using (25) in CPR is to transform the system in (15) by first left-muliplying it with the matrix

$$\begin{pmatrix} I_p & -W_R \\ 0 & I_n \end{pmatrix}. \tag{26}$$

This makes the upper-right block of the matrix "nearly zero" (it is exactly zero when W_R is ideal), effectively decoupling the pressure system from the mole fraction equations. Once the system is transformed, the CPR algorithm can again be interpretted as an MGR method, but with trivial injection operators for R and P. We do not consider this variant of MGR here.

Improving MGR for reservoir models The CPR operators R and P given by (20)/(24) approximate the ideal operators R_* and P_*. Although CPR has proven to be a highly effective solver for reservoir simulation, in the framework of MGR, it is easy to imagine alternative approximations that can be more robustness for difficult classes of problems. One simple approach is to first note that the W_R and W_P for the ideal case satisfy the following linear systems:

$$A_{nn}^T W_R^T = -A_{pn}^T; \qquad A_{nn} W_P = -A_{np}. \tag{27}$$

Hence, one good approach for computing W_R and W_P is to apply an iterative method to these systems as in Algorithm 1.

Algorithm 1 Prolongation Relaxation algorithm

Let $W_R^0 = 0$ and $W_P^0 = 0$
For $k = 1, \ldots, k_{\max}$:
$W_R^k \leftarrow W_R^{k-1} + (-A_{pn} - W_R^{k-1} A_{nn}) M_{nn}^{-1}$
$W_P^k \leftarrow W_P^{k-1} + M_{nn}^{-1}(-A_{np} - A_{nn} W_P^{k-1})$

Any smoother can be used as the relaxation method M_{nn}. In practice, just one step of block Jacobi is often good enough and produces nearly the same operators as the CPR method in (24):

$$\text{Block Jacobi:} \quad \begin{cases} W_R = -A_{pn}\ \text{blockdiag}(A_{nn})^{-1}, \\ W_P = -\text{blockdiag}(A_{nn})^{-1}\ A_{np}. \end{cases} \tag{28}$$

Other methods such as ILU and larger values of $k_{\max}$ can also be used. This typically improves the robustness of MGR, but at the cost of a higher computational complexity. Finding the right balance between these competing factors is important.

One way to reduce operator complexity is to notice that the reduction method in (17)–(19) does not require both R_* and P_*. For example, we can define restriction to be the injection operator $R_I = \begin{pmatrix} I_p & 0 \end{pmatrix}$ and we have

$$\begin{aligned} &(I - P_*(R_I A P_*)^{-1} R_I)(I - S(S^T A S)^{-1} S^T A) \\ &= \begin{pmatrix} I - (A_{pp} - A_{pn}A_{nn}^{-1}A_{np})^{-1} A_{pp} & -(A_{pp} - A_{pn}A_{nn}^{-1}A_{np})^{-1} A_{pn} \\ A_{nn}^{-1}A_{np}(A_{pp} - A_{pn}A_{nn}^{-1}A_{np})^{-1} A_{pp} & I + A_{nn}^{-1}A_{np}(A_{pp} - A_{pn}A_{nn}^{-1}A_{np})^{-1} A_{pn} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{nn}^{-1}A_{np} & 0 \end{pmatrix} \\ &= 0 \end{aligned}$$

The same result holds if we keep R_* and define interpolation to be the injection operator P_I instead. With this version of the reduction method, we can choose to set either W_R or W_P to be zero and use Algorithm 1 to compute the other one.

For the MGR method studied in the numerical results section, we set $W_R = 0$ and define W_P as in (28). We use the pre-smoothing variant of MGR in (22) with smoother given by global block Jacobi relaxation followed by block Jacobi F-relaxation. We use the MGR preconditioner in Algorithm 2 to precondition GMRES.

Algorithm 2 MGR preconditioner for Reservoir Simulation

Let $r = b$ and $e_0 = 0$
Global Relaxation: $e_1 \leftarrow e_0 + M^{-1} r$, where $M = \text{blockdiag}(A)$
F-Relaxation: $e_2 \leftarrow e_1 + S M_{nn}^{-1} S^T (r - A e_1)$ where $M_{nn} = \text{blockdiag}(A_{nn})$
Coarse Grid Correction: $e_3 \leftarrow e_2 + P M_C^{-1} R_I (r - A e_2)$ where M_C^{-1} is approximated by a classical AMG method

Numerical Results

In this section, numerical experiments are performed for the MGR preconditioner. The MGR preconditioner is implemented as a part of system solver in HYPRE (Falgout *et al.* , 2006). The HYPRE software library is a collection of high performance preconditioners and solvers for large sparse linear systems of equations on massively parallel machines. All of the experiments are preformed on a multi-socket Linux cluster at LLNL with 324 compute nodes connected by InfiniBand QDR (QLogic). All compute nodes have dual socket 8-core Intel Xeon E5-2670 (2.6 GHz) processors for 16 cores per node and 64 GB memory per node. The code is compiled using gcc-4.9.2 as the compiler and openmpi-1.6.5 for the MPI implementation.

The details of the MGR preconditioner for the tests are as follows. The pressure unknowns are marked as the C-points and other mole fractions unknowns are marked as F-points. The block Jacobi smoother is already efficient for using as the F-relaxation and global relaxation. Each block for global relaxation and F-relaxation corresponds to one cell. Since the coarse grid matrix is mostly elliptic in nature and AMG methods are efficient for solving this problem, BoomerAMG has been used as the solver for the coarse grid. The coarsening scheme is HIMS, the interpolation is with extended+i interpolation truncated to at most 4 elements per row. The smoothers is one step of lexicographic ordered hybrid Gauss-Seidel relaxation. For the full simulations, Newton method is applied and the termination tolerance is 10^{-6}. The number of maximal Newton iterations is 20. The linear solver is preconditioned GMRES(50) and its maximal inner iterations is 100.

The first test is a 2-phase oil-water model. It has a simple domain of size $20m \times 20m \times 20m$. The permeability and porosity are homogenous. The porosity is 0.2 and the permeability is set to $k_x = k_y = k_z = 100\ mD$. There is one injection well at one corner and one production well at the opposite corner. The total simulation time is 200 days. We use this test to study the convergence and scalability of the proposed MGR preconditioners. In order to test the convergence, we use a sequence of different structured grids. There sizes are $10 \times 10 \times 10$, $20 \times 20 \times 20$, $40 \times 40 \times 40$, $80 \times 80 \times 40$, and $160 \times 160 \times 40$. Note that although the grid size decreases as we refine it, this does not lead to higher heterogeneity. However for the problem on real field, higher grid resolution usually leads to higher heterogeneity and hence to more difficult Jacobian systems to solve. In Table 1 shows the total number of Newton iterations, total number of linear iterations, total wall time, and the average linear solver time. From this table, we can see that the CPU time for solving one Jacobian system is proportional to the size of the test problem. This means that the MGR precondiitoner has optimal computational complexity for this test problem.

Problem Size	# Newton steps	#linear iterations	total wall time	Average solver time
$10 \times 10 \times 10$	40	520	13.03s	0.025s
$20 \times 20 \times 20$	39	624	109.83s	0.176s
$40 \times 40 \times 40$	39	683	1029.93s	1.508s
$80 \times 80 \times 40$	40	741	4503.50s	6.077s
$160 \times 160 \times 40$	39	758	19507.37s	25.735s

Table 1: Convergence test for the MGR preconditioner

Then, we test this model to investigate the scalability of MGR preconditioner. For the weak scalability test, we fix the local problem size as 64000 cells on each core. Table 2 shows the number of Newton iterations, linear iterations, and total wall time of the numerical test. As shown in the table, the number of iterations for the linear solver is stable as we increase the problem size and number of cores. Almost linear scalability was achieved up to 64 cores. The strong scalability is tested on multiple nodes for the test problem too. Table 3 shows the results of strong scalability with up to 128 cores. From the table, we observe that the total number of linear iterations does not increase as the number of processes increases. It shows that the preconditioner is stable even with multiple cores. However, as shown in Figure 1 the strong scalability of the total wall time of the solver deteriorates as number of processes increases. The main reason is because of the low linear solver parallel efficiency. This is mainly due to the dominant cost of communication of the AMG solver on the coarse grid.

#nodes	# cores	# Problem size	# Newton steps	#linear iterations	total wall time
1	4	$40 \times 40 \times 40$	40	725	1643.05s
1	8	$80 \times 40 \times 40$	39	686	1757.35s
2	16	$80 \times 80 \times 40$	39	697	1802.53s
4	32	$80 \times 80 \times 80$	39	706	1926.63s
4	64	$160 \times 80 \times 80$	38	656	2041.30s

Table 2: Weak scalability tests for the MGR preconditioner

The second test is to evaluate the performance and the scalability of the MGR preconditioner for a large and more complex model, the SPE10 benchmark problem. This test problem uses the dataset from the Tenth SPE Comparative Solution Project, which was designed to compare the ability of upscaling approaches used by various participants to predict the performance of a water-flooding in a

#nodes	# cores	# Newton steps	#linear iterations	total wall time	speed up
1	8	39	685	196.98s	5.228
2	16	39	683	109.22s	9.430
3	24	39	683	80.63s	12.774
4	32	39	674	62.19s	16.561
5	40	39	683	60.25s	17.094
6	48	39	691	55.74s	18.470
7	56	39	674	50.92s	20.226
8	64	39	674	46.49s	22.154
16	128	39	677	41.06s	25.083

Table 3: Strong scalability tests

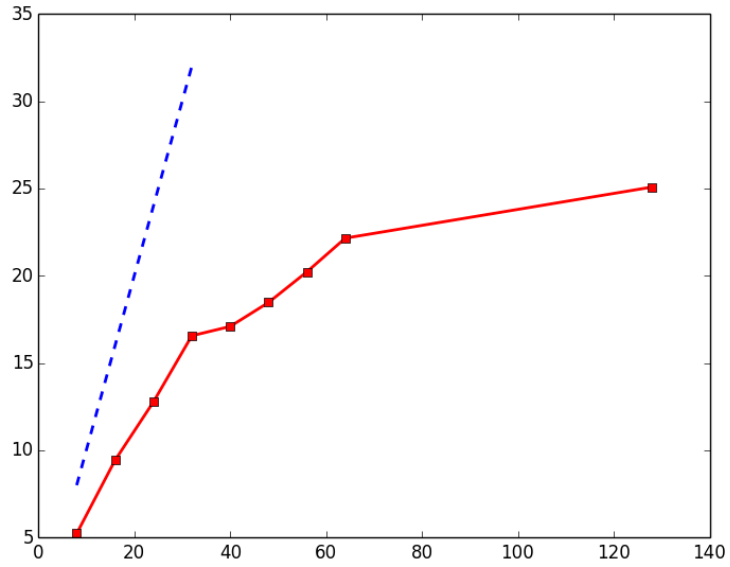

Figure 1: Scalability of MGR preconditioner

simple but highly heterogeneous reservoir. The model is generated on a $60 \times 220 \times 85$ regular Cartesian grid. The model dimensions are $1200 \times 2200 \times 170(ft)$. The top 70 ft (35 layers) represents the Tarbert format and the bottom 100 ft (50 layers) represents Upper Ness. There is one injector in the center of the field and four produces at each of the four corners. The simulation time is 100 days. Table 4 shows the total wall time and the total number of linear iterations. By using more processors, the number of linear iterations slightly increases. However the scalability of the computing time is still good.

# nodes	# cores	Total time	# linear Iterations
8	64	404.64s	1474
16	128	198.42s	1499
32	256	132.58s	1539

Table 4: Scaling tests for SPE10 model

Lastly, we investigate the robustness of the proposed MGR preconditioners with real field problem. The problem is a 3-phase (water, gas, oil) model. The fine grid contains 3 million cells in total. It has 53 wells, 25 injection wells, and 28 production wells. The matrices we are testing is the Jacobian systems of the second Newton step at $t = 100$. Table 5 presents the numerical results for the Jacobian system of the model problem. It shows that the linear solver is robust. Even using more processors, the convergence rate is similar. It has good scalability for the solve time. Although the speedup of the sequential MGR preconditioner over ILU(5) is only 1.13, MGR precondiitoner is almost scalable with problem size the processors, which is very important for large-scale problems. Due to the complexity of the problem, ILU preconditioner with smaller fill-in number are not converging for these problems. Although ILU(5)

works well for these two models because the problem size is relatively small. It is very inefficient for large size of problem and difficult to develop a parallel ILU(5). However in the test our new algorithm still performs more efficiently and robustly. The numerical results shows the efficiency of our new approach for practical reservoir simulation problems.

# nodes	# cores	method	# iterations	solve time	speed up
1	1	MGR	9	24.67	1.13
1	2	MGR	8	11.02	2.53
1	4	MGR	9	7.05	3.95
1	8	MGR	9	4.58	6.08
4	32	MGR	10	1.45	19.19
1	1	ILU(5)	8	27.83	1

Table 5: Convergence test for the MGR preconditioner

Discussion and Conclusion

We presented the MGR framework and applied it to reservoir modeling. Lots of the existing preconditioners for reservoir modeling, like The two stage preconditioners, can be rewritten as a special case of MGR. The general framework provides Based on the framework we also proposed an effective and robust preconditioner for solving the large-scale Jacobian systems arising from a fully implicit discretization.

We tested the efficiency and robustness of the new MGR preconditioner by applying it to several reservoir models. We also tested the scalability of the MGR preconditioner with both simple benchmark models and real field problem with more complicated features. 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.

However, more testing is certainly needed to draw any final conclusions for the proposed MGR preconditioner. In particular, the influence of wells especially the hydraulic wells, models with fractures or other geophysical properties, or multi-phase models should be carefully investigated. In all of the numerical tests, a two level MGR with the block Jacobi relaxation is already good enough. However this cannot always be expected to be true. some more complex cases should be considered to design a more robust multilevel MGR preconditioner with stronger relaxation.

Acknowledgements

The authors appreciate Bui, Quan Minh in University of Maryland for preparing the simulator and data for the numerical tests. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

Al-Shaalan, Tareq Mutlaq, Klie, Hector Manuel, Dogru, Ali H., & Wheeler, Mary Fanett. 2009. Studies of Robust Two Stage Preconditioners for the Solution of Fully Implicit Multiphase Flow Problems. *Soc. Pet. Engrs. J.*

Allen, Myron B. III, Behie, Grace A., & Trangenstein, John A. 1992. *Multiphase Flow in Porous Media*. Vol. Volume 34 of Lecture Notes in Engineering. Springer-Verlag.

Appleyard, J., & Cheshire, I. 1983. Nested factorization. *Reservoir Simulation Symposium, SPE*.

Bank, Randolph E, & Smith, R Kent. 1999. The incomplete factorization multigraph algorithm. *SIAM Journal on Scientific Computing*, **20**(4), 1349 – 1364.

Bank, Randolph E, & Smith, R Kent. 2002. An Algebraic Multilevel Multigraph Algorithm. *SIAM J. on Scientific Computing*, **23**(5), 1572–1592.

Bastian, Peter. 1999. *Numerical computation of multiphase flow in porous media*. Ph.D. thesis, Habilitationsschrift.

Behie, A., & Vinsome, P. K. W. 1982. Block iterative methods for fully implicit reservoir simulation. *Soc. Pet. Engrs. J.*, 658 – 668.

Cai, Xiao-Chuan, & Sarkis, Marcus. 1999. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, **21**(2), 792 – 797.

Cao, Hui. 2002. *Development of techniques for general purpose simulation*. Ph.D. thesis, Standord university.

Cao, Hui, Tchelepi, Hamdi A, Wallis, John Richard, & Yardumian, Hrant E. 2005. Parallel scalable unstructured CPR-type linear solver for reservoir simulation. *In: SPE Annual Technical Conference and Exhibition*.

Christie, MA, Blunt, MJ, *et al.* . 2001. Tenth SPE comparative solution project: A comparison of upscaling techniques. *In: SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Clees, Tanja, & Ganzer, Leonhard. 2007. An Efficient Algebraic Multigrid Solver Strategy for Adaptive Implicit Methods in Oil Reservoir Simulation. *Reservoir Simulation Symposium, SPE*.

Cusini, Matteo, Lukyanov, Alexander A., Natvig, Jostein, & Hajibeygi, Hadi. 2015. Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. *Journal of Computational Physics*, **299**(10), 472–486.

Dubois, Olivier, Mishev, Ilya D., & Zikatanov, Ludmil. 2009. Energy Minimizing Bases for Efficient Multiscale Modeling and Linear Solvers in Reservoir Simulation. *Reservoir Simulation Symposium, SPE*.

Falgout, R. D., Friedhoff, S., Kolev, Tz. V., MacLachlan, S. P., & Schroder, J. B. 2014. Parallel Time Integration with Multigrid. *SIAM J. Sci. Comput.*, **36**(6), C635–C661. LLNL-JRNL-645325.

Falgout, Robert D, Jones, Jim E, & Yang, Ulrike Meier. 2006. The design and implementation of HYPRE, a library of parallel high performance preconditioners. *Pages 267 – 294 of: Numerical solution of partial differential equations on parallel computers*. Springer.

Falta, Ronald William. 1992. *Multiphase Transport of Organic Chemical Contaminants in the Subsurface*. Ph.D. thesis, Department of Material Sciences and Mineral Engineering, University of California, Berkeley.

Foerster, Hartmut, Stüben, K, & Trottenberg, Ulrich. 1981. Nonstandard multigrid techniques using checkered relaxation and intermediate grids. *Pages 285 – 300 of: Elliptic Problem Solvers,*. Academic Press, New York.

Gries, Sebastian, Stuben, Klaus, Brown, Geoffrey L., Chen, Dingjun, & Collins, David A. 2013. Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations. *Reservoir Simulation Symposium, SPE*.

Hammersley, R.P., & Ponting, D.K. 2008. Solving Linear Equations in Reservoir Simulation Using Multigrid Methods. *In: SPE Russian Oil and Gas Technical Conference and Exhibition*. Society of Petroleum Engineers.

Helmig, Rainer. 1997. *Multiphase Flow and Transport Processes in the Subsurface: A Contribution to the Modeling of Hydrosystems*. Springer Berlin Heidelberg.

Hu, Xiaozhe, Liu, Wei, Qin, Guan, Xu, Jinchao, & Zhang, Zhensong. 2011. Development of a fast auxiliary subspace pre-conditioner for numerical reservoir simulators. *In: SPE Reservoir Characterisation and Simulation Conference and Exhibition*. Society of Petroleum Engineers.

Hu, XiaoZhe, Xu, JinChao, & Zhang, ChenSong. 2013a. Application of auxiliary space preconditioning in field-scale reservoir simulation. *Science China Mathematics*, **56**(12), 2737 –2751.

Hu, Xiaozhe, Wu, Shuhong, Wu, Xiao-Hui, Xu, Jinchao, Zhang, Chen-Song, Zhang, Shiquan, & Zikatanov, Ludmil. 2013b. Combined preconditioning with applications in reservoir simulation. *Multiscale Modeling & Simulation*, **11**(2), 507 – 521.

Lacroix, S., Vassilevski, Y., Wheeler, J., & Wheeler, M. 2003. ITERATIVE SOLUTION METHODS FOR MODELING MULTIPHASE FLOW IN POROUS MEDIA FULLY IMPLICITLY. *SIAM J. Sci. Comput.*, **25**(3), 905 – 926.

Lacroix, Sébastien, Vassilevski, Yuri V., & Wheeler, Mary F. 2001. Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS). *Numerical Linear Algebra with Applications*, **8**(8), 537–549.

Li, Zhongze, Saad, Yousef, & Sosonkina, Masha. 2003. pARMS: a parallel version of the algebraic recursive multilevel solver. *Numerical linear algebra with applications*, **10**(5-6), 485 – 509.

Liu, Hui, Wang, Kun, Chen, Zhangxin, & Jordan, Kirk E. 2015. Efficient Multi-stage Preconditioners for Highly Heterogeneous Reservoir Simulations on Parallel Distributed Systems. *Reservoir Simulation Symposium, SPE*.

MacLachlan, Scott, Manteuffel, Tom, & McCormick, Steve. 2006. Adaptive reduction-based AMG. *Numerical Linear Algebra with Applications*, **13**(8), 599 – 620.

Parter, Seymour V. 1987. On an estimate for the three-grid MGR multigrid method. *SIAM Journal on Numerical Analysis*, **24**(5), 1032 – 1045.

Ries, Manfred, Trottenberg, Ulrich, & Winter, Gerd. 1983. A note on MGR methods. *Linear Algebra and its Applications*, **49**, 1–26.

Saad, Yousef, & Suchomel, Brian. 2002. ARMS: an algebraic recursive multilevel solver for general sparse linear systems. *Numerical Linear Algebra with Applications*, **9**(5), 359–378.

Scheichl, R., Masson, R., & Wendebourg, J. 2003. Decoupling and Block Preconditioning for Sedimentary Basin Simulations. *Computational Geosciences*, **7**(4), 295–318.

Stueben, Klaus, Clees, Tanja, Klie, Hector, Lu, Bo, & Wheeler, Mary Fanett. 2007. Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation. *Reservoir Simulation Symposium, SPE*.

Tchelepi, Hamdi A., & Jiang, Yuanlin. 2009a. Scalable Multistage Linear Solver for Coupled Systems of Multisegment Wells and Unstructured Reservoir Models. *Reservoir Simulation Symposium, SPE*.

Tchelepi, Hamdi A, & Jiang, Yuanlin. 2009b. Scalable Multistage Linear Solver for Coupled Systems of Multisegment Wells and Unstructured Reservoir Models. *In: SPE reservoir simulation symposium*. Society of Petroleum Engineers.

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

Wallis, J. R., Kendall, R. P., Little, T. E., & Nolen, J. S. 1985. Constrained residual acceleration of conjugate residual methods. *SPE Reservoir Simulation Symposium*.

Watts, J. W. 1981. A conjugate gradient truncated direct method for the iterative solution of the reservoir simulation pressure equation. *Soc. Petrol. Engin. J.*, **21**, 345–353.

Zhou, Yifan, Jiang, Yuanlin, & Tchelepi, Hamdi A. 2013. A scalable multistage linear solver for reservoir models with multisegment wells. *Computational Geosciences*, **17**(2), 197–216.