SPE 163589

Solver Preconditioning Using the Combinatorial Multilevel Method

Yuhe Wang, SPE, John E. Killough, SPE, Texas A&M University

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in The Woodlands, Texas USA, 18–20 February 2013.

Abstract

The purpose of this paper is to report the first preliminary study of the recently introduced Combinatorial Multilevel (CML) method for solver preconditioning in large-scale reservoir simulation with coupled geomechanics. The CML method is a variant of the popular Algebraic Multigrid (AMG) method yet with essential differences. The basic idea of this new approach is to construct a hierarchy of matrices by viewing the underlying matrix as a graph and by using the discrete geometry of the graph such as graph separators and expansion. In this way, the CML method combines the merits of both geometric and algebraic multigrid methods. The resulting hybrid approach not only provides a simpler and faster set-up phase compared to AMG, but the method can be proved to exhibit strong convergence guarantees for arbitrary symmetric diagonally-dominant matrices. In addition, the underlying theoretical soundness of the CML method contrasts to the heuristic AMG approach, which often can show slow convergence for difficult problems.

This new approach is implemented in a reservoir simulator for both pressure and displacement preconditioners in the multi-stage preconditioning technique. We present results based on several known benchmark problems and provide a comparison of performance and complexity with the widespread preconditioning schemes used in large-scale reservoir simulation. An adaptation of CML for unsymmetric matrices is shown to exhibit excellent convergence properties for realistic cases.

Introduction

Reservoir simulation, which mimics or infers the behavior of fluid flow in a petroleum reservoir system through the use of mathematical models, is a practice that is widely used in petroleum upstream development and production. Reservoir simulation was born as an efficient tool for reservoir engineers to better understand and manage assets. However, like any numerical simulation tool, reservoir simulation is inherently computational intensive and easily becomes inefficient if more grids, coupled physics, and/or complex geometry are necessary to accurately describe the complex phenomena occurring in the subsurface. Mathematically speaking, reservoir simulation solves a system of discretized partial differential equations (PDEs) which describe the underlying physics. Due to stability constraints, an implicit formulation is required at least for the pressure system. Details about the numerical analysis for choosing an implicit formulation (or more specifically, the backward Euler method) can be found in the classic literature of Aziz and Settari (1979). However, as a recent exception, Piault and Ding (1993) attempted a fully explicit scheme in a reservoir simulation on a massively parallel computer and showed acceptable results. They adopted the Dufort and Frankel scheme which is unconditionally stable but numerically inconsistent (Dufort and Frankel 1953). This scheme is of order of $\Delta t^2/\Delta x^2$ accuracy, which clearly implies the truncation error can be significant if Δt does not approach *0* faster than Δx . In essence, implicit formulation is the only unconditionally stable and consistent scheme and is adopted by all commercial reservoir simulators. As a result, a linear solver is inevitable for reservoir simulation due to this implicit formulation.

There are four main streams of formulations applied in reservoir simulation: IMPES, fully-implicit, AIM, sequential implicit. Of these, fully-implicit is the most robust formulation but the resulting coupled system matrix is numerically challenging and computationally expensive. In the fully-implicit formulation, pressure, saturation/mass, and/or temperature are to be solved simultaneously. The generated system matrix is highly non-symmetric and not positive definite, which brings great challenges for applying robust and efficient preconditioners and liner solvers. This situation is further exacerbated for large-scale models with highly heterogeneous coefficients and unstructured gridding. Since, generally speaking, in black-oil simulation the solution of linear system ($Ax = b$) usually consumes up to 90% of the total execution time, linear solver performance enhancement means significant reservoir simulator speedup.

Many problems in petroleum extraction require understanding of fluid flow and its interaction with formation displacements. Fluid extraction and/or injection in deformable a reservoir formation modifies the *in-situ* stress field which may cause surface subsidence, fault activation, wellbore instability, thermal fracturing, etc. in geomechanically weak formations. To understand these problems one needs to perform coupled flow and displacement simulation. Generally, there are three approaches to couple flow simulation with poroelastic calculation: explicitly coupling, iterative coupling and fully-implicit coupling (Gai et al. 2003; Dean et al. 2006, Lu et al. 2007). In the explicit coupling scheme displacements are solved at selected time-steps. In iterative coupling flow and displacement are solved sequentially and then iteratively coupled at each time step. For fully-implicit coupling, flow and displacements are solved simultaneously through a full system matrix that contains flow and displacement contributions. Fully-implicit coupling is the most stable approach of these and can have second-order convergence for nonlinear iterations. However, the resulting coupled matrix becomes even more challenging to be solved efficiently compared with the fully-implicit flow simulation without rock deformation.

The quest for robust and efficient linear solvers in the fully-implicit formulation and fully-coupled flow and displacement is one of the main themes focused on by simulator developers in the petroleum industry. Matrix scaling and reordering and variants of the ILU method remain the state-of-the-art of most reservoir simulators. However, the convergence rate is not independent of problem size and can be slow for difficult matrices. Inspired by the different characteristics of pressure and saturation parts of the full system matrix, two-stage preconditioning method was developed as a 'divide-and-conquer' using the CPR (Constraint Pressure Residual) approach of Wallis (1983, 1985). Cao et al. (2005) extended CPR to a general multi-stage preconditioning framework for fully implicit flow simulation. Under this framework, the full matrix system is decomposed into different sub-blocks to deal effectively with the specific algebraic characteristic of each subset of equations. For instance, the pressure part is mainly elliptical and nearly symmetric while the saturation part is mainly hyperbolic and non-symmetric. As the result, it may be more efficient to customize a preconditioning method for different sub-blocks. The nearly symmetric pressure sub-block is usually diagonal dominant with positive diagonal and non-positive off-diagonal entries. Thus, it is generally positive (semi-) definite. This algebraic character enables us to apply the popular AMG approach for pressure preconditioning. Since the first application of the multigrid method in reservoir simulation by Behie and Forsyth (1983), AMG has become a very attractive option for the pressure solution. A number of implementations have been reported with promising performance. Generally, the convergence rate is independent of matrix size and scales linearly with matrix size. Stüben et al. (2007) developed efficient AMG implementations for fully-implicit and sequential implicit formulations. Klie et al. (2007) designed deflation AMG preconditioners for highly ill-conditioned reservoir simulation problems. The elliptic displacement sub-block resulted from coupled flow and geomechanics modeling is symmetric positive (semi-) definite, which makes multigrid applicable. White and Borja (2011) applied AMG as sub-preconditioner for fully coupled flow and geomechanics. Alpak and Wheeler (2012) implemented a supercoarsening multigrid solver for poroelasticity in 3D coupled flow and geomechanical modeling.

However, AMG is based on heuristics, especially for the classic Ruge-Stüben AMG (Ruge and Stüben 1987). Although it often exhibits impressive performance in practice, it does not offer guarantees on the speed of convergence especially for challenging matrices. In this paper, a newly developed Combinatorial Multilevel (CML) method (Koutis et al. 2007) is introduced to reservoir simulation problems. CML has provable convergence properties and sound theoretical machinery. It not only offers convergence guarantees for SDD (Symmetric Diagonally Dominant) matrices with arbitrary weights, but also has lower grid, operator and computational complexities comparing with other variants of AMG methods.

To the best knowledge of the authors, this is the first implementation of CML in reservoir simulation with coupled geomechanics. The contribution of this paper is that it adapts CML into the multistage preconditioning solution technique and provides performance comparisons with other popular preconditioners using challenging benchmarks. The paper is organized as follows. First, we briefly describe the multistage precondition framework and discuss the applicability of CML in this framework. Second, we present the CML algorithm. Third, we show comparisons using several case experiments. Finally, the conclusions and outlook are provided.

For the example cases CML results are compared with ILU(0) and two popular variants of AMG, Ruge-Stüben AMG and aggregation-based AMG (Notay 2010, 2012; Napov and Notay 2012). We convert and integrate a Matlab implementation of CML (Koutis et al. 2009) into our in-house multistage preconditioning framework of a comprehensive reservoir simulator. To compare with aggregation-based AMG, we choose the AGMG package (v3.2) (Notay 2012). For Ruge-Stüben AMG (RS_AMG hereafter), we use an implementation that is available as part of the PyAMG package (Bell et al. 2011). For all of the cases, the convergence criterion is set to $\|b - Ax\| / \|b\| < 1.0^{-6}$. All of the experiments were performed on a 64-bit Mac OS X 10.7 system with a 2.3 GHz dual-core Intel Core i5 processor and 8 GB DDR3 memory.

Solution Technique – Multistage Preconditioning

The multistage preconditioning framework was introduced to fully-implicit reservoir simulation by Cao et al. (2005). To keep the presentation concise, we describe the key algorithmic steps of two-stage preconditioning. The extension to multistage is straightforward. To solve the following linear system

$$Ax = b$$

where A is the coupled system matrix that contains pressure and saturation sub-blocks, we perform the following steps:

1. Map total residual to the constraint decoupled pressure residual, r_P . Several possible mappings are available for

this stage, for example, an IMPES-like decoupling or a simple algebraic decoupling.

2. Solve the decoupled pressure system using a linear solver of selection to obtain $x_P = \tilde{A}_P^{-1} r_P$. This is the first-stage preconditioning.
3. Update the total residual $r_{updated}$ using newly computed pressure x_P, $r_{updated} = r - AWx_P$. W is a mapping matrix to map x_P to the total solution vector.
4. Solve the fully coupled system using a selected linear solver to obtain $x = M^{-1} r_{updated} + Wx_P$.

The 4 steps are repeated until convergence or stopping criterion is reached. Note that, in step 2, a preconditioned linear solver is applied to solve the decoupled pressure system. M^{-1} in step 4 acts as the second stage preconditioner. As a result, a nested iteration is formed such that a pressure sub-block is solved at the inter iteration while M^{-1} acts as a global smoother at the outer iteration. In practice, ILU, Gauss-Sediel, or block SOR is often an effective choice of M^{-1}. But these traditional preconditioners might not work well with the pressure sub-block that is mainly elliptic and cannot scale with the matrix size. Stüben et al. (2007) discussed the algebraic properties of the decoupled pressure sub-block and concluded that AMG is a favorable choice of preconditioner.

It is natural to appreciate that the multistage preconditioning methodology can also be applied to coupled flow and geomechanics simulation. We provide here a solution strategy that merges IMPES and fully coupled flow and poroelastic calculations. Indeed, the coupling between flow and poroelastic calculations is through pressure only. In this strategy, pressure and poroelastic calculations are coupled via GCR acceleration (Eisenstat et al. 1983). To solve

$$Ax = b$$

where A is now the full system containing pressure and displacement sub-blocks, we perform the following steps:

1. Map total residual to the constraint-decoupled pressure residual, r_P.
2. Solve the decoupled pressure system using a linear solver of choice to obtain $x_P = \tilde{A}_P^{-1} r_P$. This is the first stage preconditioning.
3. Update displacement residual r_D using newly computed pressure solution x_P.
4. Solve the displacement system using a selected linear solver to obtain $x_D = \tilde{A}_D^{-1} r_D$. This is the second stage preconditioning.
5. Map the constraint solution to the total estimate of pressure vector.
6. Update increment residual vector.
7. Make the increment residual vector orthogonal to previous increment direction.
8. Calculate step size.
9. Update solution and residual vectors.

The 9 steps are repeated until convergence or stopping criterion is reached. Using this approach a nested iteration is formed. There are two inner iterations for pressure and displacement. The outer iteration couples flow and displacement using GCR. AMG has been implemented as a sub-preconditioner for the mainly elliptic pressure sub-block and elliptic displacement sub-block. In the abovementioned places where AMG has been implemented, we now replace the solver with CML. The algebraic characteristics of both the pressure and displacement matrices should favor CML solution. In the following section, we will briefly describe the CML method.

The Combinatorial Multilevel Method

Before describing the algorithm of CML, we first show two examples in which aggregation based and classical AMG have convergence troubles. The first matrix comes from a maximum flow in network problem (Livne 2012). The resulting matrix is highly ill-conditioned with condition number about 10^{19}. Its sparse matrix plot is provided in **Fig. 1**. Note that all the sparse

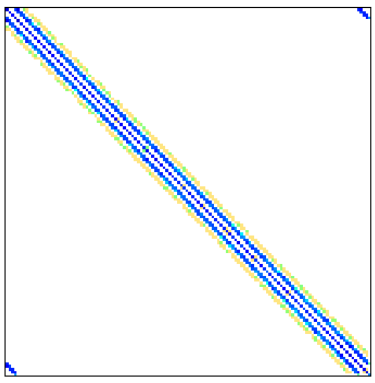

Fig. 1 – Sparse matrix plot

matrix plots in the paper are generated using the CSPY tool of CSparse package (Davis 2006). Zero entries are white. Entries with tiny absolute value are light orange and entries with large magnitude are black. In the midrange, it ranges from light green to deep blue.

Table 1 – Number of CG iterations

Method	Iterations
CML	28
AGMG	N/A
RS_AMG	N/A
ILU(0)	2726

The iteration counts of the CG accelerator are listed in **Table 1**. The relative residual reduction is plotted in **Fig. 2**. N/A denotes AGMG and RS_AMG do not converge in 10000 iterations. Clearly, it can be seen that CML can readily solve this problem as opposed to AGMG and RS_AMG. The reason for this is that unlike AGMG and RS_AMG, CML is not limited by indefinite matrices.

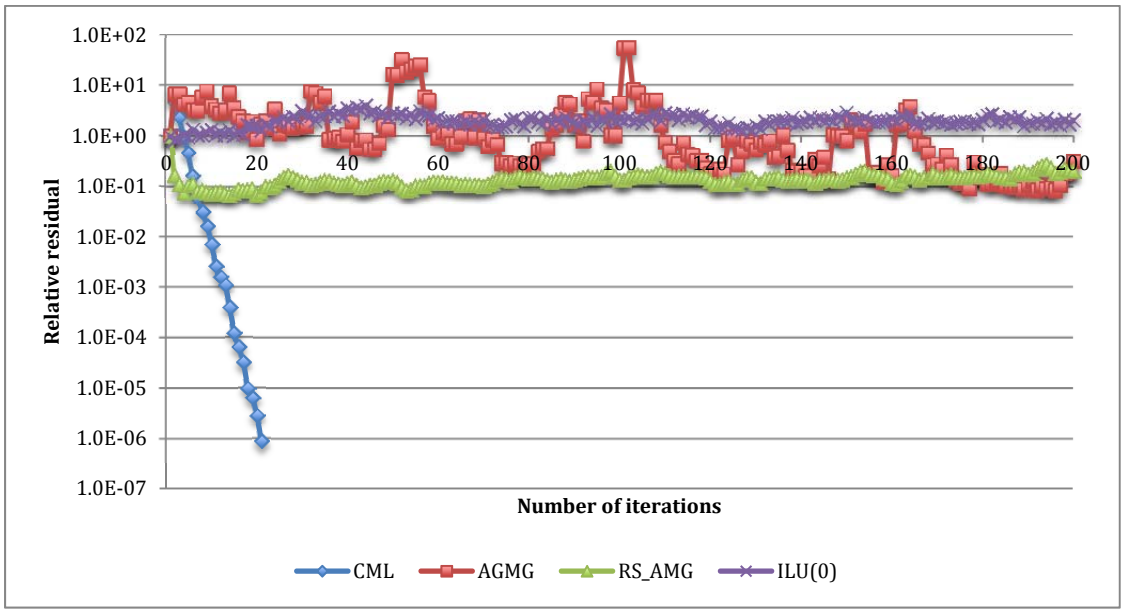

Fig. 2 – Relative residual reduction

The second example is extracted from a matrix of a reservoir simulation application (Beckner 2006). The original matrix is highly unsymmetric and describes a coupled system with more than one unknown per gridblock. We convert the matrix to a symmetric matrix by extracting a connected graph of the original matrix. The resultant matrix is not indefinite. Its sparse plot is provided in **Fig. 3**. As listed in **Table 2** and plotted in **Fig. 4**, for this SPD matrix, CML shows the fastest convergence while AGMG exhibits convergence difficulties.

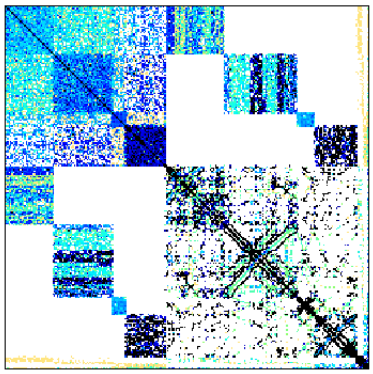

Fig. 3 – Sparse matrix plot

Table 2 – Number of CG iterations

Method	Iterations
CML	18
AGMG	634
RS_AMG	97
ILU(0)	1187

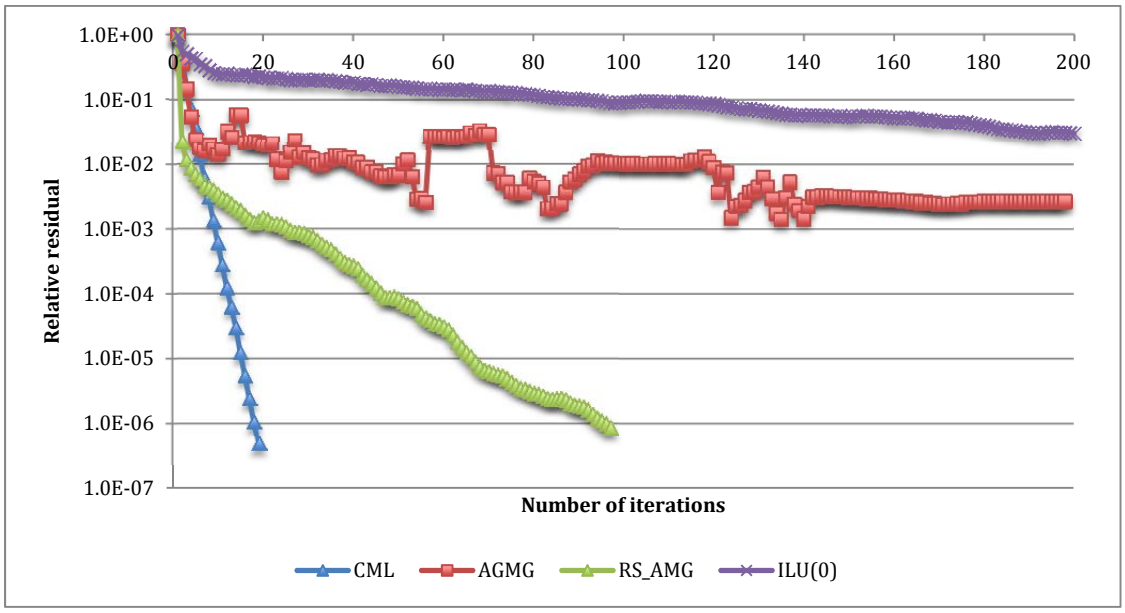

Fig. 3 – Relative residual reduction

The above excellent performance of CML can be attributed to its provable convergence properties and sound theoretical machinery. In this section, we describe the underlying principles of CML. As its name suggested, CML is inspired by the popular multilevel method, AMG, yet with two significantly different distinguishing features. CML features a uniquely different coarsening strategy that is faster than various AMG approaches and is easier to implement. The second feature is that CML is a truly black-box solver while AMG has several algorithmic input options that may be crucial for convergence, especially for the classic AMG. Note that aggregation-based AMG has a much better black-box feature than classic AMG. Although in practice the time spent in the set-up phase is generally negligible compared to the iteration phase, such timing can reflect the efficiency of the hierarchy construction and can also suggest an easy implementation. We focus on describing the two-level approach. Extension to multilevel is straightforward. To keep the presentation concise, we simplify the algebra and only present the key ingredients of CML. Algorithmic details with proofs can be found in Koutis (2007) and Koutis et al. (2009).

Table 3 lists the two-level CML algorithm to solve $Ax = b$, where A is the laplacian matrix. It should be noted that any SDD matrix can be converted to laplacian with light-weight transformation. It can be seen that this algorithm resembles the simple form of the two-level method with an exception that it works with the normalized laplacian. The reason for using normalization is described in Chung's monograph on spectral graph theory (Chung 1997). One reason is that the eigenvectors of the partition of vertices of a graph become orthogonal using the normalized laplacian. It is the restriction matrix R and the constructed hierarchy of matrices ($R^T AR$, which is called quotient graph in support theory) that distinguishes CML (**Table 3**) with other variants of AMG. A thorough description of the machinery that is used to build the hierarchy requires an extensive background of support theory for graph and Steiner preconditioners. We only point out the key ingredients in this paper and refer readers to Boman and Hendrickson (2003) for support theory for preconditioning and Gremban (1996) for Steiner preconditioners. The intuitive foundation of combinatorial preconditioning is that a preconditioner for the laplacian of a graph *A* should be the laplacian of a simpler graph *B*, which is derived from *A* in a principled fashion. Instead of providing algebraic formulations, we describe the Steiner preconditioner using an illustrative graph. As shown in **Fig. 4**, the Steiner preconditioner partitions the *n* vertices of the underlying laplacian graph *V* into *m* vertex-disjoint clusters C_i ($i = 1, \cdots, m$). Each C_i contains a star graph with leaves corresponding to the vertices in C_i. These leaves are rooted at a vertex r_i. The roots r_i ($i = 1, \cdots, m$) are connected to form the quotient graph. The restriction matrix R, which is of size $n \times m$, is constructed as $R_{i,j} = 1$ if vertex *i* is in cluster *j* and $R_{i,j} = 0$ if vertex *i* is not in cluster *j*.

In principle, CML combines the Steiner preconditioner with a multilevel method using a graph-theoretical approach. The basic idea of this new approach is to construct a hierarchy of matrices by viewing the underlying matrix as a graph and using

the discrete geometry of the graph such as graph separators and expansion. In this way, the CML method combines the merits of both geometric and algebraic multigrid methods, which provide strong convergence guarantees for SDD matrices.

Table 3 – Two-level CML algorithm

Two-level CML
Input: laplacian A, vector b, current solution x^k $n \times m$ restriction matrix R Output: Updated solution x^{k+1} 1. // Normalization $D = diag(A)$; $\hat{A} = D^{-1/2} A D^{-1/2}$; $\hat{b} = D^{-1/2} b$; $\hat{x}^k = D^{1/2} x^k$; 2. // Jacobi pre-smoothing $\hat{x}^k_{presmoothed} = \left(I - D^{-1}\hat{A}\right)\hat{x}^k + D^{-1}\hat{b}$; 3. // Restriction $r^k = \hat{b} - \hat{A}\hat{x}^k_{presmoothed}$; $r^k_{coarse} = R^T D^{1/2} r^k$; 4. // Solve using coarse level $A_{coarse} = R^T A R$; $\hat{x}_{coarse} = A^{-1}_{coarse} r^k_{coarse}$; 5. // Correction $\hat{x}^{k+1} = \hat{x}^k_{presmoothed} + D^{1/2} R \hat{x}_{coarse}$; 6. // Jacobi post-smoothing $\hat{x}^{k+1}_{postsmoothed} = \left(I - D^{-1}\hat{A}\right)\hat{x}^{k+1} + D^{-1}\hat{b}$ 7. // Variable transformation $x^{k+1} = D^{-1/2}\hat{x}^{k+1}_{postsmoothed}$

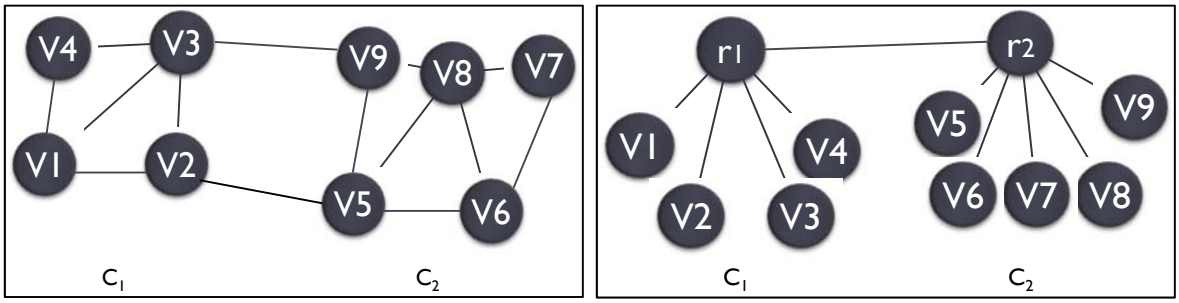

Fig. 4 – Illustration of Steiner preconditioner

Case Experiments

As it can be seen from the previous section, the core of the CML method is built for SDD matrices with general weights. In other words, theoretically, it only guarantees convergence for this class of matrices. In practice, we extend CML to also handle the nearly symmetric pressure sub-block matrix that is derived from black-oil simulation. In the case experiments, we first test on an incompressible system using the SPE 10th comparative project model (Christie and Blunt 2001) whose resulting pressure sub-block matrices are SDD and not indefinite. For this case, we provide the comparison of performance and complexity of CML, AGMG, RS_AMG and ILU(0). Next, based also on the SPE 10th comparative project model, we perform an experiment using a black-oil system that generates a nearly-symmetric pressure sub-block matrix. In addition, we

also provide tests on a series of matrices from unstructured gridding (Beckner et al. 2006; Diyankov et al. 2007). Finally, we test the performance of these methods on a finite element displacement sub-block matrix using a test instance from the University of Florida sparse matrix collection (Davis 1994). Since the time spent in the set-up phases of these preconditioners are negligible comparing to the iteration phases, the number of iterations of a chosen accelerator is a strong indicator of the performance of the respect preconditioner. We also introduce a notation of iteration cost by multiplying iteration counts by computational complexity to quantify the effective work consumed by each preconditioner.

1. Incompressible oil-water system

We build a 5-spot SPE 10 problem by defining one water injector at the center and four producers at the four corners. The compressibility of oil, water and rock is neglected to make an incompressible system. Note that the four preconditioners are written using different programming languages with unknown code optimization levels. We thus choose not to compare the elapsed time. Since the number of iterations taken to converge is strongly correlated to the time, we use iteration counts as a more fair comparison criterion. Listed in **Table 4** are the iteration counts taken by CML, AGMA, RS_AMG, and ILU(0). CG is used as the accelerator. Obviously, the three variants of AMG outperform ILU(0) by orders of magnitude. More importantly, CML is clearly the winner over AGMG and RS_AMG. The relative residual reductions of the full SPE 10 model (85 layers) are plotted in **Fig. 5**. It can be seen from **Fig. 5** that, the relative residual of CML decreases linearly in log scale while AGMG and RS_AMG deviate from log linear reduction to some extent. It is worth mentioning that there are a few algorithm knobs that can affect the performance of RS_AMG dramatically, such as the method used to determine the strength of connection between unknowns of the underlying matrix. A careless choice may even destroy the convergence of RS_AMG.

Table 4 – Iteration counts

Method	Iter #
CML	28
AGMG	43
RS_AMG	59
ILU(0)	2726

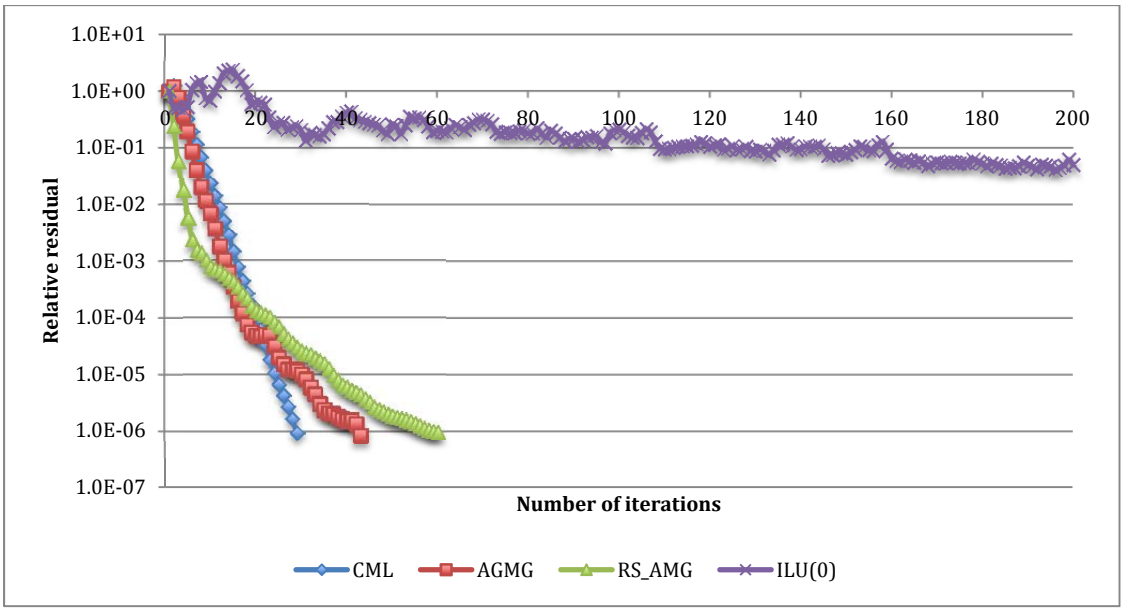

Fig. 5 – Relative residual reduction

As abovementioned, aggregation-based and classic AMG can scale linearly with matrix size. To compare the scalability of CML, AGMG and RS_AMG, we perform experiments by adding the layers of the SPE 10 model. To test how these methods scale with matrix size, we record elapsed time of each method and compare the normalized time that is taken to be the time taken by the multi-layer model divided by the time taken by a single layer model for each method respectively. The comparison results are shown in **Fig. 6**. Since ILU(0) scales badly with matrix size, the plots in **Fig. 6** are cut to better shown the differences among CML, AGMG and RS_AMG. It can be seen from **Fig. 6** that CML scales linearly as matrix size. Moreover, some supper linear scalability is exhibited, as the normalized time of 85 layers is 81.83. Neither AGMG nor RS_AMG shows strict linear scalability. In addition, for AGMG and RS_AMG the deviation from linear scalability seems to be enlarged when passing layer 35 (transition from non-fluvial Tarbert formation to channelized Upper Ness formation). This is believed to be attributable to the significant weight change of the underlying matrix graph. In contrast, thanks to the proved

ability to handle general weights, CML is shown to be insensitive to this change.

To make the comparison complete, complexities of these methods should be analyzed. Although these variants of multilevel preconditioners significantly enhance the convergence rate, they require extra cost in each iteration compared to ILU(0). We first provide the grid complexity and operator complexity of the three multilevel methods (**Fig. 7**). Grid complexity is

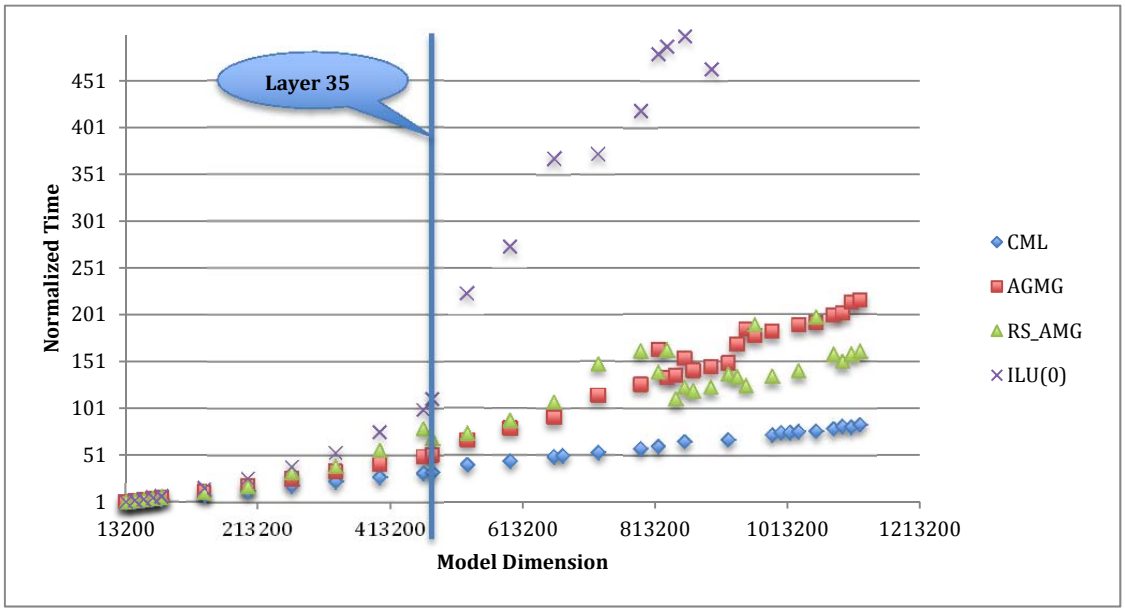

Fig. 6 – Normalized time vs. model dimension

defined as the total number of grid points of all levels divided by the number of original fine grid points. Operator complexity is similarly defined as the total number of nonzero entries of all levels divided by the number of nonzero entries of original fine grid. Clearly, smaller grid and operator complexities are favored. As it can be seen from **Fig. 7** and **Fig. 8**, CML preserves the smallest grid and operator complexities among the three approaches. In addition, the grid and operator complexities of AGMG and RS_AMG show variations as matrix dimension increases while the variations are almost flat for CML. RS_AMG has the worst grid and operator complexities. RS_AMG applies a heuristic approach to mimic the grid coarsening of geometric multigrid using the connection strengths of matrix entries, while CML and AGMG use an agglomerative clustering technique. Since the number of nonzero entries determines the number of floating point operations in preconditioning, the computational complexity is directly related to grid and operator complexity.

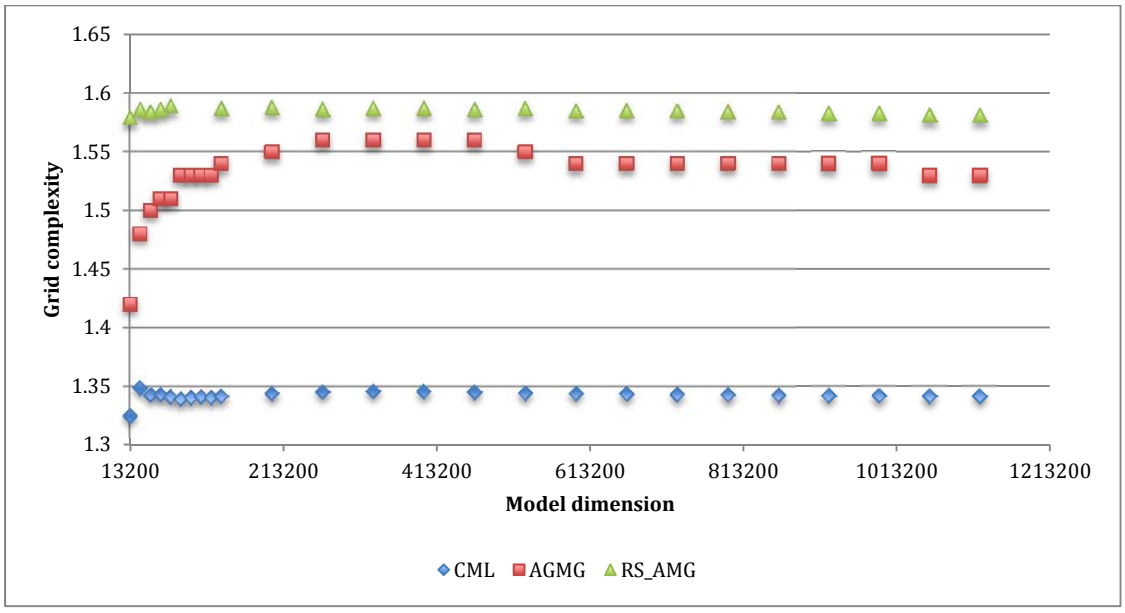

Fig. 7 – Grid complexity

Fig. 9 shows the estimated computational complexity of CML, AGMG, RS_AMG, and ILU(0). Computational complexity is defined as the number of floating point operations a preconditioner consumes per iteration (normalized by the number of nonzero entries of the original fine matrix). As expected, ILU(0) has the lowest computational complexity among the 4 methods. The computational complexity of CML is lower than AGMG except when the grid dimension is smaller than about 100,000. RS_AMG has the most expensive computational complexity. Since the computational complexity is the extra work per iteration, we introduce the effective iteration counts as the performance indicator. The iteration costs of the 85-layer model are listed in **Table 5**. It can be seen that, the advantage of CML over AGMG and RS_AMG is further increased when these factors are taken into account.

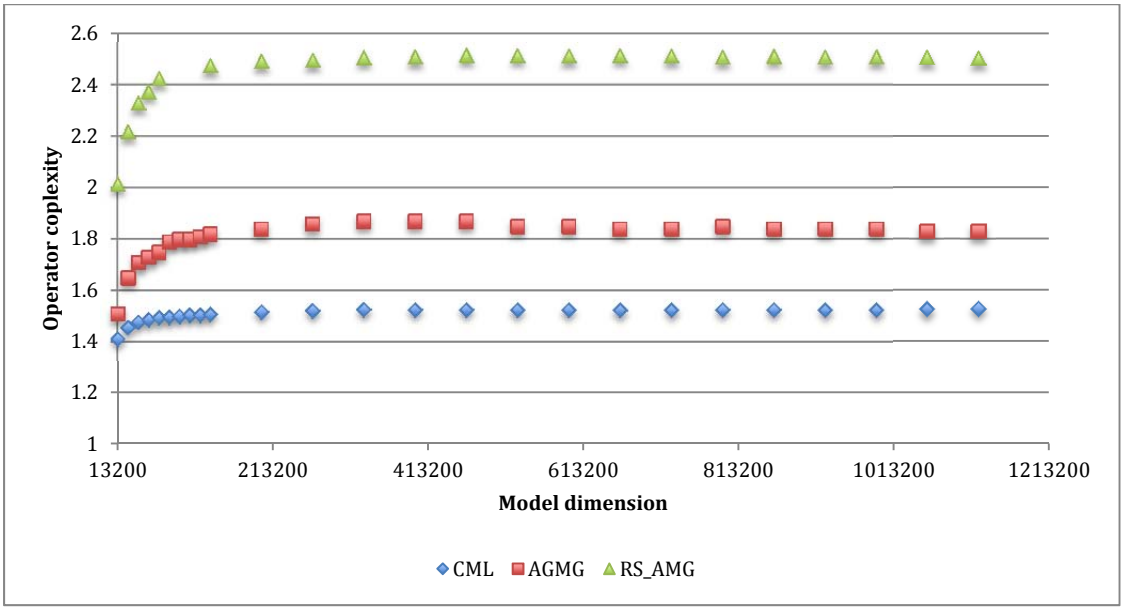

Fig. 8 – Operator complexity

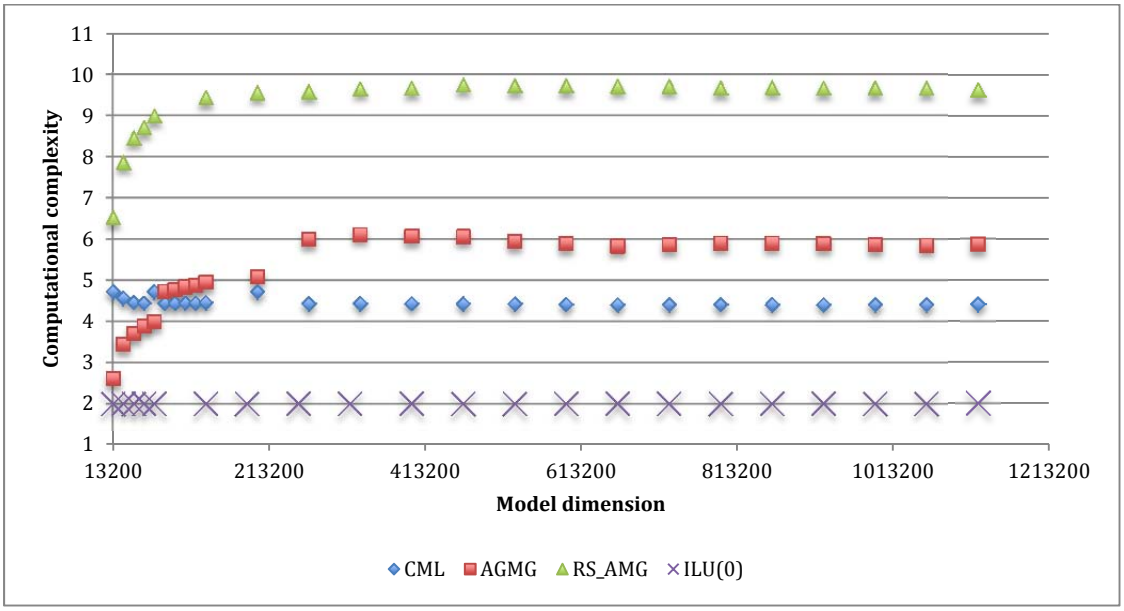

Fig. 9 – Computational complexity

Table 5 – Iteration costs

Method	Iter cost
CML	123.39
AGMG	252.41
RS_AMG	568.76
ILU(0)	5452.0

2. Black-oil system

Since the pressure sub-block matrix is generally nearly symmetric, it is more interesting to test the performance of CML on this class of matrices. We extend the incompressible system to a black-oil system. Watts (1981) proposed an approach to build a symmetric approximation of the resulting nearly symmetric pressure matrix. He then reformulated the solution process by adding an outer iteration such that (to solve $Ax = b$):

$$
\begin{aligned}
& r = b - Ax^k \\
& \textit{while (not converged)} \\
& \quad \Delta x = A_S^{-1}; \\
& \quad x^{k+1} = x^k + \Delta x; \\
& \quad r = b - Ax^{k+1}; \\
& \textit{end}
\end{aligned}
$$

where A_S is the symmetric approximation of A. Results of Watts' work indicated that 2 or 3 iterations is generally enough for convergence. The original work was attempting to use the conjugate gradient method as the accelerator for the linear solver. With the development of non-symmetric accelerators such as GMRES or BICGSTAB a few years later (Saad and Schultz 1986; Van der Vorst 1992), the requirement for matrix symmetry was eliminated. We instead build a hierarchy of matrices using A_s and use this hierarchy as a preconditioner for GMRES. The hope is that the slight non-symmetry does not change much of the spectrum of A. Moreover, we attempt to directly build a preconditioner using the nearly symmetric A with CML. Similarly, we compare the convergence of CML, AGMG, RS_AMG and ILU(0) using full SPE 10 model. **Table 6** lists the iteration counts and iteration costs of each method. **Fig. 10** provides the relative residual reduction history. CML_unsymm denotes that we apply CML directly to A while CML_symm means we apply CML to A_S. GMRES(10) is applied as the accelerator. CML still outperform AGMG, RS_AMG and ILU(0). But surprisingly, CML_unsymm has the best performance. Its residual tends more to decrease log linearly than others. The mathematical justification could not be provided at this point. A hypothesis of CML_unsymm's excellent performance is that the nearly symmetric A is still (semi-) positive definite. In addition, support theory for preconditioning, which is the foundation of CML, might be able to be extended to more general matrices. Boman and Hendrickson (2003) discussed the extension of support theory to general matrices and Huang (2012) generalized support theory for preconditioning to non-symmetric matrices. Understanding of the performance of CML on nearly symmetric matrices and the development of a multilevel method based on support theory for slightly non-symmetric matrices clearly requires further research.

Table 6 – Iteration counts and costs

Method	Iter #	Iter cost
CML	18	77.22
AGMG	28	106.40
RS_AMG	42	455.70
ILU(0)	539	1078.0

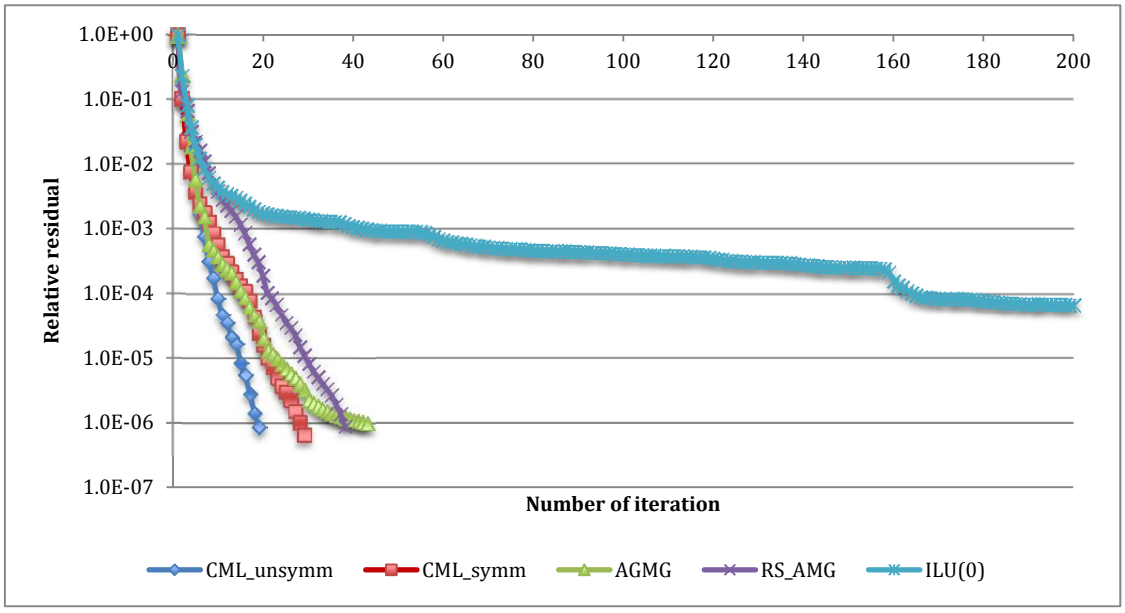

Fig. 10 – Relative residual reduction

In addition, compared to the incompressible system, the performance of these non-symmetric preconditioners seems to become better in this black-oil case. This can be understood by realizing that the underlying matrices become strictly diagonal dominant due to the effect of compressibility.

To assess the applicability of the solvers in unstructured reservoir simulation, we perform further experiments using a series of matrices from unstructured reservoir models (Beckner et al. 2006; Diyankov et al. 2007). Unfortunately, there is no publicly available information of these models except the system matrices alone. We can only infer the underlying information by viewing the matrices. To apply CML, AGMG and RS_AMG, we extract the nearly symmetric pressure-like matrices since the original matrices appear to be from coupled systems. **Fig. 11** lists the sparse matrix plots of the extracted matrices. Clearly, these were derived from unstructured gridding. **Table 7** shows the performance of the various solvers. It can be seen from **Table 7** that CML performs better for larger matrices (SBO-3). For small-sized matrices, it seems that CML is not as efficient as AGMG and is even worse than ILU(0) for SEO-1, CI-1 and CIT-1. For SBO-4 and CI-1, the iteration counts of CML and AGMG are close. The reason why CML has worse efficiency than AGMG is that the computational complexity of CML is about two times higher than AGMG. As we have seen in **Fig. 9**, the computational complexity of AGMG is lower than CML when the matrix dimension is small and the computational complexity of CML is flat as the matrix dimension increases. This also implies that CML tends to perform better for large-scale matrices. It should be noted that for the 6 test instances, CML is applied directly to the slightly unsymmetric matrices. Hence, these results also suggest that CML is applicable for this type problem making it a promising alternative method for large-scale flow simulation.

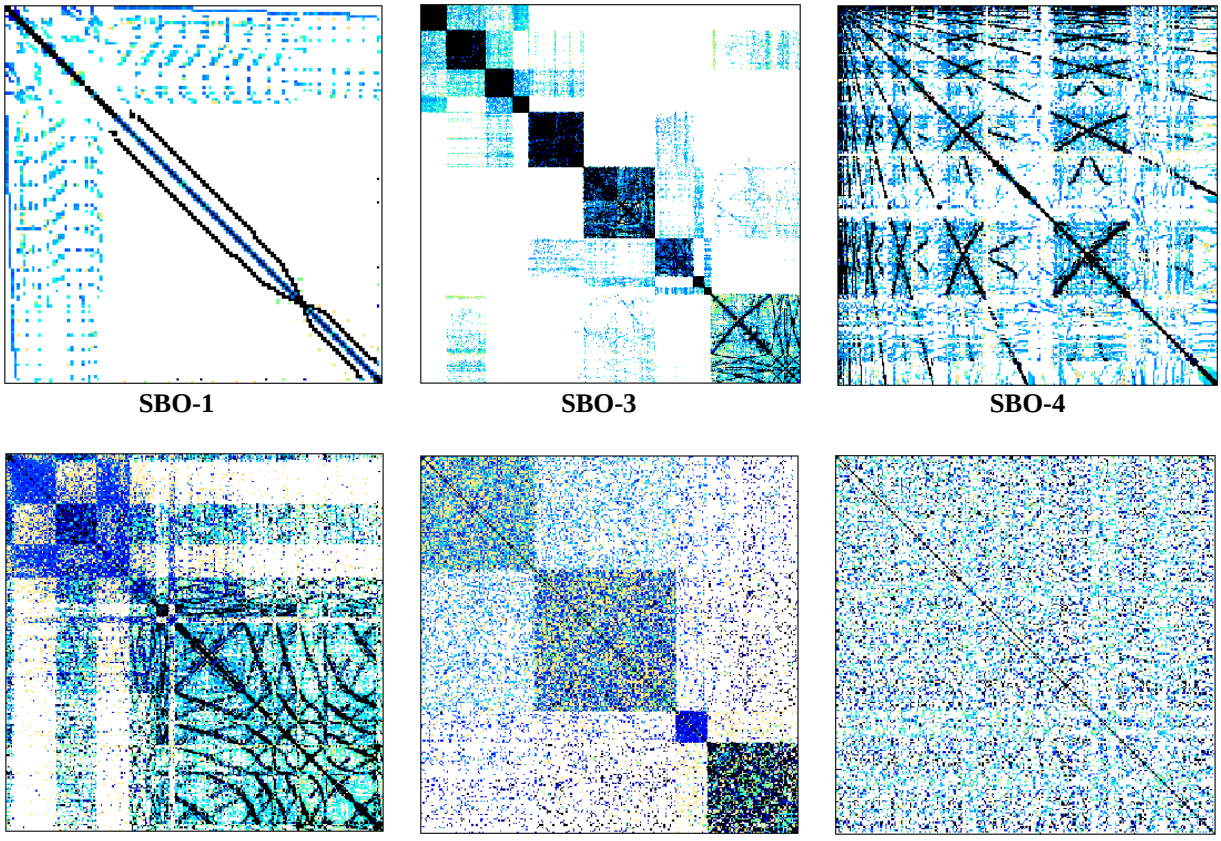

Fig. 11 – Sparse matrix plot

Table 7 – Iteration counts and costs

	Size (unknown#, nonzero#)	CML		AGMG		RS_AMG		ILU(0)	
		Iter #	Iter cost	Iter #	Iter cost	Iter #	Iter cost	Iter #	Iter cost
SBO-1	(21692, 144986)	14	60.20	24	78.96	48	350.4	81	162
SBO-3	(2165051, 1849317)	14	56.20	40	136.90	49	401.31	326	652
SBO-4	(61956, 486010)	28	114.24	25	67.5	68	685.44	175	350
SEO-1	(21498, 185700)	10	28.4	6	9.72	4	35.44	6	12
CI-1	(13500, 88860)	17	65.96	13	16.25	5	52.75	11	22
CIT-1	(4359, 28041)	26	112.06	24	85.44	18	144.72	32	64

3. Displacement computation in coupled flow and geomechanics

The underlying matrix for displacement computation is symmetric. Hence, we can directly apply CML as preconditioner if the displacement matrix is diagonal dominant. If diagonal dominance cannot be preserved, neither CML nor AMG guarantees convergence. Since there is no well-documented benchmark problem for coupled flow and geomechanics, we instead test directly on a benchmark matrix from the University of Florida sparse matrix collection (Davis 1994). The test case comes from a coupled flow and geomechanical study of CO_2 sequestration in a depleted gas reservoir (Ferronato et al. 2010).

The 3D view of the depleted gas reservoir and its 2D planar view are shown in **Fig. 12**. The finite element discretization has about 250,000 nodes and more than 1,250,000 elements. A number of local and regional faults exist in this reservoir, which is shown as solid line the **Fig. 12**. The data results from the application of commercial reservoir simulator for flow simulation while the study the fault activation and ground deformation is derived from a geomechanical simulation. The resulting sparse matrix is plotted in **Fig. 13**. Note that it has been reordered by the reverse Cuthill-McKee (RCM) algorithm.

Table 8 – Iteration counts and costs

Method	Iter #	Iter cost
CML	26	78.52
AGMG	243	634.23
RS_AMG	75	318.75
ILU(0)	121	242.00

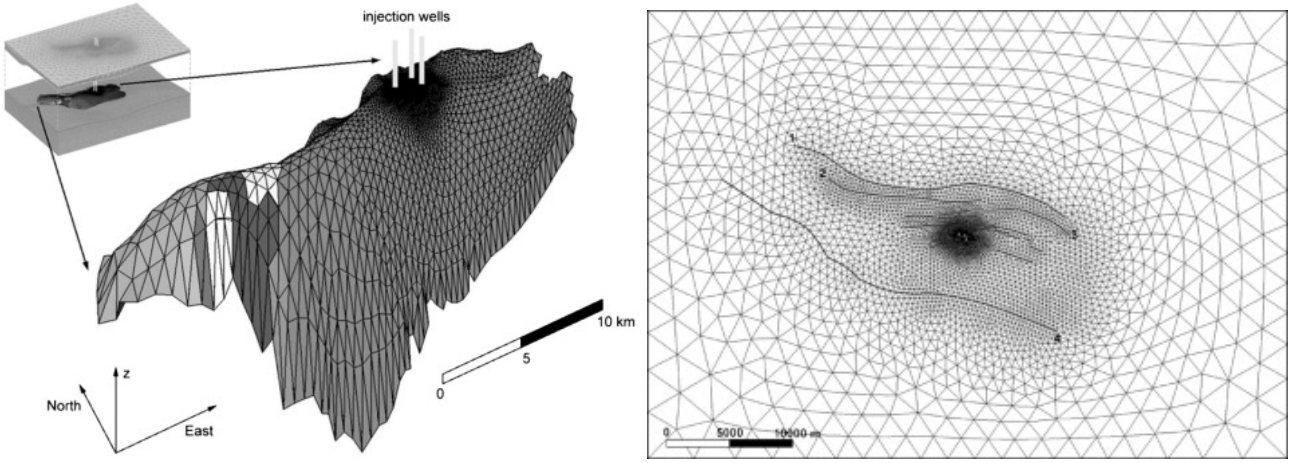

Fig. 12 –3D (left) and 2D (right) view of the depleted gas reservoir for CO_2 sequestration (From Ferronato et al. 2010)

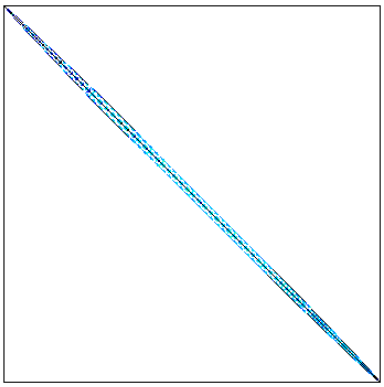

Fig. 13 – Sparse matrix plot of displacement matrix

The iteration counts and costs are listed in **Table 8** and iteration reduction histories are plotted in **Fig. 14**. Obviously, CML is again the winner among the four approaches. In addition, similar to the incompressible system case, the residual reduction of CML decreases log linearly. For this case, AGMG exhibits a poor performance and is even worse than the plain ILU(0).

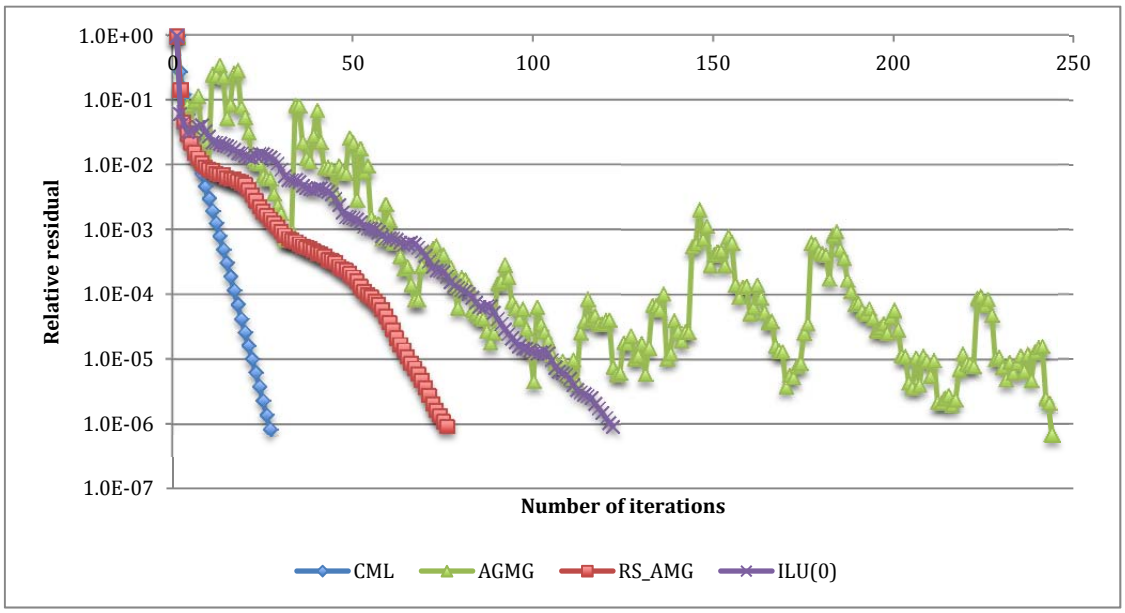

Fig. 14 – Relative residual reduction

Conclusions and Outlook

In summary, this paper introduces a new multilevel preconditioner, CML, to reservoir simulation and coupled geomechanics. CML is rooted in support theory and Steiner preconditioner and is integrated with the popular multilevel approach. The resulting algorithm has a unique matrix hierarchy building machine that tends to bring geometry information back into the algebraic operations thus introduces proven strong convergence guarantees for SDD matrices with general weights.

We implement CML into the multistage preconditioning framework for reservoir simulation and coupled geomechanics. Specifically, CML is applied for pressure and poroelastic displacement preconditioning. We perform experiments on a series of examples and compare the performance of CML with AGMG, RS_AMG, and ILU(0). From the results, we have the following findings:

1. CML has better scalability than AGMG and RS_AMG. Through the incompressible system example, we show only CML can scale strictly linearly using the SPE 10 model.
2. CML has lower grid and operator complexity than AGMG and RS_AMG, which reveals it has better hierarchy building machinery.
3. Although without theoretical justification yet, it is shown that CML can be directly applied to nearly symmetric pressure-like matrices. Its performance is superior to AGMG and RS_AMG for large-scale matrices. Handling nearly symmetric matrices robustly and efficiently is a prerequisite for pressure preconditioning in reservoir simulation application. CML is shown to be capable in this aspect through our experiments.

This preliminary study shows CML is a promising alternative for pressure and displacement preconditioning in reservoir simulation and coupled geomechanics, especially for large-scale models. A relative unpleasant aspect of CML is, however, the theoretical support for nearly symmetric matrices is not available yet. Although the current algorithm is shown to work with pressure-like matrices in reservoir simulation and has better performance than AGMG and RS_AMG for large models, we need to justify or develop new algorithms based on CML. Indeed, one of the purposes of this paper is to bring attention to this new way of pressure and displacement preconditioning and to serve as an introduction for further research in this area.

References

Aziz, K. and Settari, A. 1979. *Petroleum Reservoir Simulation*. Applied Science Publishers Ltd, London.

Alpak, F.O. and Wheeler, M.F. 2012. A Suppercoarsening Multigrid Method for Poroelasticity in 3D Coupled Flow and Geomechanics Modeling. Computational Geoscience 16:953-974. http://dx.doi.org/10.1007/s10596-012-9297-z.

Beckner, B.L., Usadi, A.K., Ray, M.B., Diyankov, O.V. 2006. Next Generation Reservoir Simulation Using Russian Linear Solvers. Paper SPE 103578 presented at the SPE Russian Oil and Gas Technical Conference and Exhibition, Moscow, Russia, 3-6 October. http://dx.doi.org/10.2118/103578-MS.

Behie, G.A. and Gorsyth, P.A. 1983. Multigrid Solution of the Pressure Equation in Reservoir Simulation. *SPE J* **23** (4): 623-632. SPE-10492-PA. http://dx.doi.org/10.2118/10492-PA.

Bell, W.N., Olson, L.N., Schroder, J.B. PyAMG: Algebraic Multigrid Solvers in Python v2.0, 2011.

Boman, E.G., Hendrickson, B. 2003. Support Theory for Preconditioning. SIAM J. Matrix Anal. Appl. 25(3): 694-717.

Cao, H., Tchelepi, H.A., Wallis, J.R., Yardumian, H. 2005. Parallel Scalable Unstructured CPR-Type Linear Solver for

Reservoir Simulation. Paper SPE 96809 presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 9-12 October. http://dx.doi.org/ 10.2118/96809-MS.

Chung, F.R.K. 1997. *Spectral Graph Theory*, American Mathematical Society.

Christie, M.A. and Blunt, M.J. 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. Paper SPE 66599 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 11-14 February. http://dx.doi.org/10.2118/66599-MS.

Davis, T.A. 1994. University of Florida Sparse Matrix Collection, NA Digest.

Davis, T.A. 2006. *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia.

Dean, R.H., Gai, X., Stone, C.M., Minkoff, S.E. 2003. A Comparison of Techniques for Coupling Porous Flow and Geomechanics. Paper SPE 79709 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 3-5 February. http://dx.doi.org/10.2118/79709-MS.

Diyankov, O.V., Koshelev, S.V., Kotegov, S.S., Krasnogorov, I.V., Kuznetsova, N.N., Pravilnikov, V.Y., Beckner, B.L., Maliassov, S.Y., Mishev, I.D., Usadi, A.K. 2007. Sparsol – Sparse Linear Systems Solver. *J Numer Math* **0**(0): 1-16.

Dufort, E.C., Frankel, S.P. 1953. Stability conditions in the numerical treatment of parabolic equations. *Math Tables and other Aids to Computation*, **7**(43):135-152.

Eisenstat, S.C., Elman, H.C., Schultz, M.H. 1983. Vibrational Iterative Methods for Nonsymmetric Systems of Linear Equations. *SIAM J Numer Anal*, **20**:345-357.

Ferronato, M., Gambolati, G., Janna, C., Teatini, P. 2010. Geomechanical Issues of Anthropogenic CO_2 Sequestration in Exploited Gas Fields. *Energy Conversion and Management*, **51**(10): 1918-1928. http://dx.doi.org/10/1016/j.enconman.2010.02.024.

Gai, X., Dean, R.H., Wheeler, M.F., Liu, R. 2003. Coupled Geomechanical and Reservoir Modeling on Parallel Computers. Paper SPE 79700 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 3-5 February. http://dx.doi.org/10.2118/79700-MS.

Gremban, K. 1996. Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems. PhD dissertation, Carnegie Mellon University.

Huang, Y. 2012. Generalization of Support Theory for Preconditioning. Fifth International Conference on Information and Computing Science. July 24-25.

Klie, H., Wheeler, M.F., Clees, T., Stüben, K. 2007. Deflation AMG Solvers for Highly Ill-Conditioned Reservoir Simulation Problems. Paper SPE 10582 presented at the SPE Reservoir Simulation Symposium, Houston, USA, February 26-28.

Koutis, I. 2007. Combinatorial and Algebraic Tools for Optimal Multilevel Algorithms. PhD Dissertation, Carnegie Mellon University (May 2007).

Koutis, I., Miller, G., Tolliver, D. 2009. Combinatorial Preconditioners and Multilevel Solvers for Problems in Computer Vision and Image Processing. In: Proceeding ISVC '09 Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I

Liven, O.E. Lean Algebraic Multigrid MATLAB Software, Release 2.1.1, 2012.

Lu, B., Alshaalan, T.M., Wheeler, M.F. 2007. Iteratively Coupled Reservoir Simulation for Multiphase Flow. Paper SPE 110114 presented at SPE Annual Technical Conference and Exhibition, Anaheim, USA, November 11-14.

Notay, Y. 2010. An Aggregation-based Algebraic Multigrid Method. *Electronic Transactions on Numerical Analysis*, **37**:123-146.

Napov, A., Notay, Y. 2012. An Algebraic Multigrid Method with Guaranteed Convergence Rate. *SIAM J Sci Comput* Vol. **34**(2):A1079-A1109. http://dx.doi.org/10.1137/100818509.

Notay, Y. 2012. Aggregation-based Algebraic Multigrid for Convection-diffusion Equations. *SIAM J Sci Comput* **34**: A2288-A2316.

Notay, Y. AGMG Software and Documentation; see http://homepages.ulb.ac.be/~ynotay/AGMG.

Piault, E., Ding, Y. 1993. A fully explicit scheme in reservoir simulation on a massively parallel computer. Paper SPE 25274 presented at the SPE Symposium on Reservoir Simulation, New Orleans, Louisiana, 28 February–3 March. http://dx.doi.org/10.2118/25274-MS.

Ruge, J.W. and Stüben, K. 1987. Algebraic Multigrid (AMG). In S.F. McCormick, editor, Multigrid Methods, volume 3 of Frontiers in Applied Mathematics, pages 73-130. SIAM, Philadelphia, PA.

Saad, Y., Schultz, M.H. 1986. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J Sci Stat Comput* **7**(3): 856-869. http://dx.doi.org/10.1137/0907058.

Stüben, K., Clees, T., Klie, H., Lu, B., Wheeler, M.F. 2007. Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation. Paper SPE 105832 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 26-28. February. http://dx.doi.org/10.2118/105832-MS.

Van der Vorst, H.A. 1992. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J Sci Stat Comput* **13** (2): 631–644

Wallis, J.R. 1983. Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration. Paper SPE 12265 presented at the SPE Reservoir Simulation Symposium, San Francisco, California, 15-18 November. http://dx.doi.org/10.2118/12265-MS.

Wallis, J.R., Kendall, R.P., Little, T.E. 1985. Constrained Residual Acceleration of Conjugate Residual Methods. Paper SPE

13563 presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, 10-13 February. http://dx.doi.org/10.2118/13536-MS.

White, J.A., Borja, R.I. 2011. Block-preconditioned Newton-Krylov Solvers for Fully Coupled Flow and Geomechanics. *Comput Geosci* **15**(4): 647-659. http://dx.doi.org/10.1007/s10596-011-9233-7.