SPE 106237

Parallel Unstructured Solver Methods for Complex Giant Reservoir Simulation

Larry S.K. Fung and Ali H. Dogru, Saudi Aramco

This paper was prepared for presentation at the 2007 SPE Reservoir Simulation Symposium held in Houston, Texas, U.S.A., 26–28 February 2007.

Abstract

The major issues for parallel solver in a modern reservoir simulator are robustness, scalability, efficiency, and flexibility. There is significant interest in running fast field-scale simulation for complex giant Middle-East reservoirs which will require tens to hundreds of millions of grid cells to give reasonable resolution. At the same time, significant geologic complexity will require the treatment of dual permeability regions, faulting and fractures, and high variations of reservoir and fluid properties. Of course, the methods should also work well for extracted sector simulation with local grid refinements in both the structured and unstructured discretization. The preconditioning methods considered in this work include both the single-stage and multistage frameworks. In the single-stage framework, a novel method in addition to the well-known variants of incomplete LU factorizations [ILU0, ILU(k), and ILUT] is considered. The new method is a highly parallel method which in this paper will be referred to as the unstructured line solve power series (LSPS) method. The method will be discussed and contrasted in light of key issues for parallel linear solvers. The unstructured LSPS has certain interesting properties in the parallel construct which makes it a highly effective component.

The multistage method researched in this work is of the constraint pressure residual (CPR) framework. The method uses approximate pressure solve as the first stage preconditioning to the full system preconditioning. A number of original adaptations based on this concept were researched. Here, the use of the parallel algebraic multigrid (AMG) method and other single-level methods mentioned above in combinations within the multistage CPR framework were explored. Certain methods constructed in this way are found to be highly efficient, scalable, and robust. The methods developed will be discussed and several test problems included in this paper.

Introduction

Parallel reservoir simulation involving millions of grid cells is now common practice and is an essential component for the management of many giant carbonate complexes in the Middle East. The recent advances are aided in part by the computational power offered by inexpensive Linux PC clusters. Many of today's parallel machines are built with mass-produced commodity-based components. At the same time, research and development on parallel highly scalable methods in the modern reservoir simulator has made routine field-scale simulation an effective and useful part of resource planning and analysis.

Field-scale analyses are often desired over sector simulation for a comprehensive understanding of overall reservoir behavior and recovery processes performance. However, special study involving an area of interest (AOI) frequently arises in a full-field project. For example, evaluation of alternative designs for expensive maximum reservoir contact (MRC) wells with intelligent downhole controls and production equipments requires near-wellbore reservoir simulation and optimization workflow. Thus, the demand for simulation capabilities with mixed structured and unstructured grids for fast field-scale megacell modeling is very high. The capability to refine and coarsen at ease regionally, and perform simulation and analyses at multiple scales within a single project is a primary near-term goal.

This paper addresses one critical component of the tool set required to accomplish the mission stated above, namely, the linear solver. The primary solver methods in the old generation of reservoir simulators typically use nested factorization (NF) or variants of incomplete LU (ILU) factorization method for preconditioning. While extension to small scale parallel processing was achieved in the late 1990s, these methods have limitations in terms of scalability and/or robustness for the very large scale simulation where parallel processing with hundreds or even thousands of processors are required for speed and performance.

Previously, within the structured grid framework, a solver method known as the z-line Neumann series, which is more scalable for parallel field-scale simulation of structured grid, was documented by Dogru et al.[1] Later, a parallel structured multigrid method was introduced by Fung and Dogru[2] for treating the local grid refinement (LGR) problems. The additional solver method for the dual porosity dual

permeability (DPDP) system was later described in Fung and Al-Shaalan.[3]

In this work, new ideas in the fully unstructured setting are being researched and developed. These ideas involve both the single-stage method and the multistage method. In the single-stage method, a novel idea of building an approximate inverse preconditioner via matrix substructuring of the Jacobian matrix was investigated. This substructuring method, which we refer to as line solve power series (LSPS), is a powerful generalization of the z-line Neumann series method. The method is fully unstructured. It is application specific in that it is sensitive to the nature of the simulation problems. It significantly increases the robustness while maintaining the parallel efficiency and performance.

In the multistage method, the two-stage constraint pressure residual (CPR) method was investigated. CPR was first introduced into the petroleum literature by Wallis.[4, 5] It was recently applied by Gratien et al.[6] and Cao et al.[7] in new simulator development where they have used parallel algebraic multigrid (PAMG) as the pressure preconditioner. The research documented here explores the Quasi- IMPES reduction methods and the use of various approaches to approximately solve the pressure as a first stage preconditioning to the full system matrix. Solver results for several sample problems are included for comparison of the various methods. These include the public domain data sets SPE1[8] and SPE10[9] comparative solution projects, as well as several megacell simulation models. To add some challenge for the solver methods, the SPE1 grid system has been uniformly refined to 300,000 cells.

To put all the methods into a proper prospective, the three variants of the ILU factorizations [ILU0, ILU(k), and ILUT] are used as baseline comparison for some problems. The ILU preconditioners are very well known and are described in the reference text by Saad,[10] thus their descriptions are not included here. Interested readers can refer to Saad's book or the many other reference papers on them.

Single-Stage Method – Matrix Substructuring LSPS

The mechanics of the method is to subdivide the Jacobian matrix into two parts:

$$A = P + E \tag{1}$$

In this approach, the matrices A, P, and E are fully unstructured. The important aspect of the approach is to choose P such that it includes the dominant terms of A and at the same time remains inexpensive to compute the inverse. Thus, matrix A can be written as:

$$A = \left(I + EP^{-1}\right)P \tag{2}$$

The approximate inverse preconditioner for A using the N term power series is:

$$A^{-1} \approx M_N^{-1} = \left[I + \sum_{k=1}^{N} (-1)^k \left(P^{-1}E\right)^k \right] P^{-1} \tag{3}$$

The method works well if P is chosen such that

$$\left\| EP^{-1} \right\| < 1 \tag{4}$$

Next, we discuss the basic principle and some useful methods for partitioning the submatrices P and E.

Framework for Constructing P

The optimal approach for the choice of non-zeros in P and system ordering is such that P has narrow bandwidths on each decomposed computational domain. The best choice of P is usually derived from the specific nature of the physical problems.

The basic procedure is:

1. Order system based on maximum connectivity graph (MCG). The initial P matrix is block tridiagonal and has the maximum coefficients on each row.
2. Select additional coefficients to include in P such that Eqn. 4 can be "mostly" satisfied.
3. Reorder system if necessary to minimize fills based on P matrix.

This procedure is done at the beginning of a simulation run as a preprocessing step. Domain partitioning can be adjusted or weighted based on the work required to treat P and E in the preconditioning operation in Eqn. 3 for load balancing. The strength of the method is increased by including more large non-zeros elements in P and/or increasing N in the series.

Maximum Connectivity Graph (MCG)

For the general unstructured grid system, the P matrix is constructed by ordering the cells based on the magnitude of the connection factors (transmissibility) with neighboring cells. The algorithm begins the search from cell 1. At a particular grid cell, all its neighbors which have not been ordered are searched to locate the next grid cell which has the largest transmissibility with the current cell. Thus the MCG is the line of maximum connection factors through the unstructured or structured grids. An example of MCG in a 2D x-z grid with a row of fracture cells is shown in Fig. 1. In the example, the vertical transmissibility is larger than the lateral transmissibility due to grid-cell geometry. Fracture transmissibility is larger than vertical transmissibility due to high fracture conductivity.

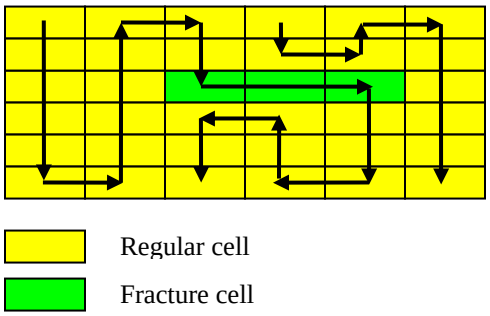

Fig. 1- MCG for a 2D grid with one row of fracture cells.

Line Bundling with MCG

For problems which involve large conductive fractures and faults, it is beneficial to selectively include the large fracture transmissibility terms in the P matrix in addition to large vertical transmissibility terms. One method to construct P in this case is to increase P by bundling lines which are ordered using the MCG algorithm. The MCG naturally traces the large lateral transmissibility. Adjacent lines can then be bundled in P for the line solve.

Similarly, in dual permeability problems, the inclusion of matrix-fracture transfer terms in the P matrix is beneficial if they are large relative to the other lateral transmissibilities. The local P matrix is reordered to minimize bandwidth.

The LSPS method discussed above works out to be a scalable, efficient, and robust method for parallel reservoir simulation of many giant field-scale problems. In this development, the method is accelerated by using either the restart version of the parallel generalized minimal residual PGMRES(k) method or the restart version of the parallel generalized conjugate residual PGCR(k) method. The LSPS is also well suited for incorporation as a component in the two-stage CPR method which will be discussed in the next section. Comparison with the three variants of the ILU method (ILU0, ILUK, ILUT) will be shown later via example problems in the section on test cases.

Multistage Method – Variable Partitioning CPR

The CPR method is a kind of divide-and-conquer strategy which specifically attacks the nature of multiphase multicomponent transport problems in reservoir simulation. The basic premise of CPR is that the full system conservation equations are mixed parabolic hyperbolic in nature with the pressure part of the problem being mainly parabolic and the concentration and saturation part of the problem being mainly hyperbolic. Therefore, CPR aims to decompose the pressure part and approximately solve it. The pressure solution is used to constraint the residual on the full system, thus achieving a more robust and flexible overall solution strategy.

The CPR preconditioning involves a pressure predictor-corrector step at every iteration and can be written as:

$$M_{CPR}^{-1} = M^{-1}\left[I - \tilde{A}CA_p^{-1}C^T\right] + \left(CA_p^{-1}C^T\right) \quad (5)$$

Where

$$\tilde{A} = W * A \quad (6)$$

And the pressure matrix is

$$A_p = C^T \tilde{A} C \quad (7)$$

C is given by

$$C = \begin{bmatrix} e_p & & & \\ & e_p & & \\ & & \ddots & \\ & & & e_p \end{bmatrix} \quad (8)$$

Suppose pressure is the first unknown for each cell, then, let e_p be the neq x 1 vector where neq is the number of equations per grid cell:

$$e_p = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (9)$$

The aim of *W* is to perform an IMPES-like reduction step on *A*. For example, *W* can be computed as:

$$W = DIAG^{-1}(A) \quad (10)$$

The notation *DIAG(A)* denotes the main diagonal neq*neq submatrix blocks of *A*. Note that *W* needs to be chosen such that $\tilde{A}_p$ is close to being positive definite in order for PAMG to work well as the pressure solver. This is not a requirement for LSPS or any of the ILU variants.

Thus, the CPR preconditioning step $x = M_{CPR}^{-1} r$ can be outlined as follows:

1. Restrict the full system residual to the pressure system

$$r_p = C^T W r \quad (11)$$

2. Iteratively solve the pressure system

$$A_p x_p = r_p \quad (12)$$

3. Expand pressure solution to full system

$$s = Cx_p \quad (13)$$

4. Correct the full system residual from pressure residual

$$r_c = r - A \cdot s \quad (14)$$

5. Solve the full system using second stage preconditioning

$$Mx = r_c \quad (15)$$

6. Correct full system solution from pressure solution

$$x_c = x + s \quad (16)$$

In this work, the preconditioner for the pressure solve is either the parallel algebraic multigrid PAMG or the LSPS method discussed in the previous section. A discussion of AMG method can be found in Ruge and Stuben.[11] While the variants of the ILU method can also be used here, our research results indicate that they are generally not as competitive for our problems. PAMG is primarily for positive definite system and is found to be generally very effective for pressure solution whereas the LSPS method in many cases is also very competitive for P solve but with somewhat lower convergence rate than PAMG. For full system preconditioning, the LSPS is the method of choice. More detail comparisons and results follow.

From the performance perspective, two issues can be problematic for the PAMG method. The first issue is that for the very large-scale highly heterogeneous problems, the computational complexity can grow as ln(k) where k is the size of the problem. Increase in computational complexity can be defined in this context as the total number of coefficients in the entire multigrid hierarchy over the original number of coefficients in the base grid. In this situation, computational work will grow as ln(k) as problem size increases. The second issue is that for very large systems, PAMG will require a fairly deep hierarchy of coarsened grid levels. The network communication cost for the coarser grid levels can be an issue for parallel performance. With these caveats, PAMG is a valuable component for P solver in the CPR method for reservoir simulation.

Test Cases

All the test cases presented here were run using a PC cluster with AMD Opteron processor 252 at clock speed 2.6GHz with 1 MB cache. The network interconnect is a Myrinet switch. The cluster has 256 compute nodes. Each compute node has two processors and we use both of the processors in distributed memory mode with MPI parallelization.

Test 1: Comparison of Single-Stage Methods

The first test compares 5 preconditioning methods for a typical million grid cell (110*306*39) Middle-Eastern black-oil reservoir model. In order to compare all the methods, the model was run in IMPES mode for 100 time steps (about 8 years of history simulation) and the linear solver is required to achieve relative residual norm tolerance of 1.0E-3. (Normally, solver residual convergence tolerances in field-scale production runs are typically set to ~1.0E-2). In these test runs, for ILU(k), the degree k was set to 2. For ILUT, the number of term for p fills was set to 6. For LSPS, MCG ordering with 3-term series was used. For AMG, preconditioning was set to one V cycle. The problem was run on 44 compute nodes (88 processors). Table 1 summarizes the average iteration counts per Newton cycle and the total solver time for this problem.

One-stage methods	Avg. linear iterations / Newton	Linear solver time / Newton (secs)
ILU0	75.3	0.60
ILU(2)	64.6	1.25
ILUT	70.2	0.98
LSPS	13.8	0.39
PAMG	2.8	1.95

Table 1 - Comparison of iterative behaviors for 5 preconditioning methods for a 1.31-million cell full-field model running on 88 processors (test case 1).

For this problem, LSPS is the fastest method whereas PAMG has the highest convergence rate. While the 3 ILU variants can solve this problem, their convergence rate and speeds are not attractive compare to the other methods. In our experience, PAMG tends to converge very rapidly for straightly M-matrix which is the case for the IMPES formulation. It can deteriorate rapidly when it is not. The multigrid setup cost is relatively high. Therefore, PAMG becomes more attractive in solutions which require multiple solves with the same matrix or when the solution requires many iterations due to a requirement for high level of residual reduction.

Test 2: Comparison of Two-Stage CPR Methods

Next, the same 1.31-million cell black-oil model was run in fully implicit (FI) mode for 100 time steps and two variants of the CPR methods where the pressure preconditioning using either the PAMG or the LSPS preconditioners are compared. The FI preconditioner is the LSPS in both cases. In order to test the methods, we set a relative residual norm tolerance of 1.0E-8. Table 2 shows the results for the two solver methods.

CPR Methods (#CPUs=88)	Avg. linear iterations / Newton	Linear solver time / Newton (secs)
CPR-LSPS	7.7	1.53
CPR-PAMG	3.3	2.72

Table 2 - Comparison of iterative behavior for CPR methods with residual tolerance at 1.0E-8 running on 88 processors (test case 2).

To converge to a normalized residual norm of less than 1.0E-8, CPR-LSPS is 1.8 times faster than CPR-PAMG. However, the convergence rate of CPR-PAMG is 2.3 times that of CPR-LSPS for this problem. Fig. 2 illustrates the typical rates of convergence for the two solver methods.

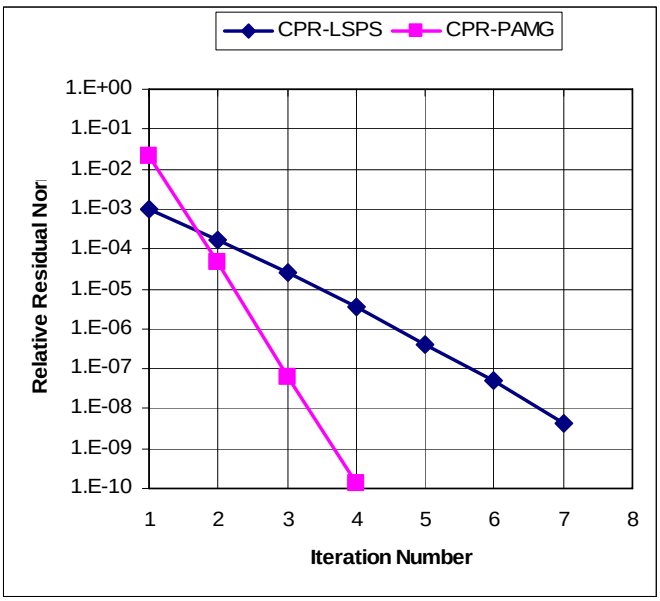

Fig. 2 - Average convergence rate of 2 CPR variants for a 1.3-million cell black-oil model.

Test 3: The Refined SPE1 Comparative Problem

The original SPE1 comparative problem[8] which was a 300 grid cell (10*10*3) black-oil gas injection model with a single gas injector (Q_g=100 MMscf/D) at one corner and a producer (Q_o=20,000 STB/D, constrained to minimum BHP=1000 psi) at the diagonal corner of the grid. To add some challenge to the solver, this original model is modified by subdividing each grid block uniformly into 10*10*10 small grid blocks. Thus, the refined SPE1 problem has 300,000 grid blocks (100*100*30) with lateral cell dimension of 100 ft and thickness of between 2 to 5 ft. All other aspects, including the injection/production rates, are kept the same as the original problem. The simulation run was for a period of 10 years.

The model was run on 16 compute nodes (32 CPUs) using three different methods: (1) one-stage LSPS, (2) two-stage CPR-LSPS, and (3) two-stage CPR-PAMG. Table 3 summarizes the solver statistics for the three methods.

Methods (#CPUs=32)	Avg. linear iterations / Newton	Total solver time (secs)	% of solver time / total run time
LSPS	5.53	1029	65.2
CPR-LSPS	1.42	537	50.7
CPR-PAMG	1.73	1494	78.0

Table 3 - Solver statistics for the refined grid 300,000 cell SPE1 model running on 32 processors (test case 3).

With small grid cells and a high gas injection rate, this model is moderately difficult for the solver. The number of solver iterations required to appropriately converge the linear part of the problem is modest. The two-stage CPR-LSPS method's convergence rate is about a factor of 4 faster then one-stage LSPS. This factor is, of course, strongly dependent on the difficulties of the problem. However, a factor from 3 to 6 is frequently observed for moderately difficult to very difficult problems. This increase in convergence rate is attributable to CPR preconditioning.

Test 4: The SPE10 Comparative Problem

Model 2 of the SPE10 comparative problem[9] is a highly heterogeneous fine grid (20ft*10ft*2ft) model with a total of 1.1-million grid blocks (60*220*85). The model consists of one rate-controlled water injector (Q_w=5,000BPD, constrained to maximum BHP=10,000psi) at the center and four bottom-hole-pressure controlled producers (BHP=4,000psi) at the respective four corners of the grid system. The overall system is nearly incompressible and is a two-phase (oil-water) immiscible problem. The simulation run was for a period of five years.

This model was run on 32 compute nodes (64 CPUs) using the three solver methods as in the refined SPE1 model discussed in test case 3. Both CPR-LSPS and CPR-PAMG have excellent convergence rate for the problem and converged to the desired tolerance in a few iterations. The one-stage LSPS method requires about 4 times more iterations and is slower than the CPR-LSPS method. CPR-PAMG timing appears to be out-of-pace as it is 4 to 6 times larger than the other methods. This is in part due to parallel efficiency issues as revealed by the scalability analysis below. Table 4 summarizes the solver statistics for these runs:

Methods (#CPUs=64)	Avg. linear iterations / Newton	Total solver time (secs)	% of solver time / total run time
LSPS	8.66	595	82.2
CPR-LSPS	2.17	367	74.9
CPR-PAMG	2.93	2285	95.5

Table 4 - Solver statistics for the SPE10 model running on 64 processors (test case 4).

The parallel efficiency (scalability) of the three solver methods are compared by running the model using different number of processors and comparing the speed-up factors. For this purpose, the model was run using 4, 8, 16, 32 compute nodes (8, 16, 32, 64 processors) on the PC clusters. The scale-up results are summarized in Fig. 3.

Both the one-stage LSPS and the two-stage CPR-LSPS scale-up linearly to superlinearly to beyond 64 processors for the SPE10 model. For the particular parallel hardware used for this work, the PAMG method currently can scale down to ~50,000 grid cells/proc, beyond which scalability starts to deteriorate. Although parallel performance is strongly affected

by how well a code has been engineered for this purpose, the practical implication for PAMG is that for the coarsened grid levels in the multigrid hierarchy, the ratio of communication versus computation cost can become a scalability constraint. For the LSPS and CPR-LSPS methods, our implementation has achieved near perfect scalability down to <10,000 grid cells/proc for most of the problems running on the test hardware.

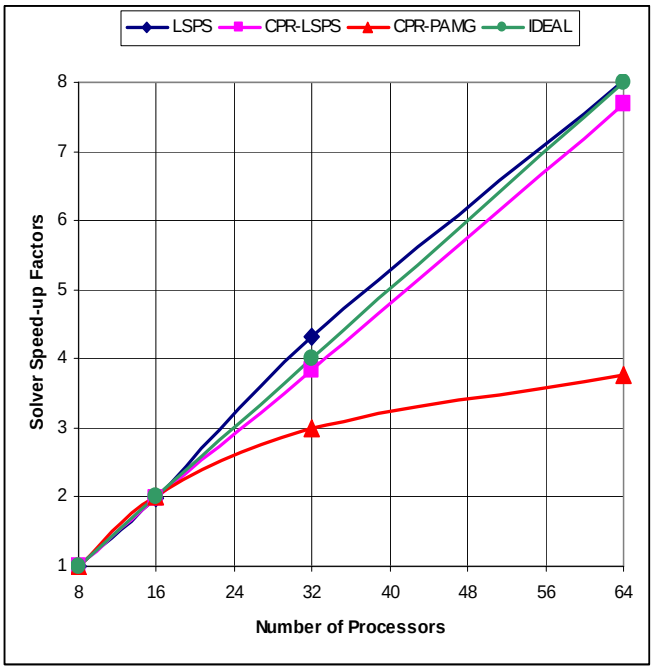

Fig. 3 - Scalability of three solver methods, LSPS, CPR-LSPS, and CPR-PAMG for the SPE10 fine-grid problem (test case 4).

Test 5: A 1.3-Million Cell Full-Field 2-Phase Oil-Water System

The grid system for this full-field problem is 110*306*39 for a total of 1.31-million grid cells. This reservoir has good permeability and pressure is maintained about bubble point pressure via peripheral water injection. There are over 100 wells in the model. The simulation is for a period of 50 years of history match. Table 5 summarizes the solver results for this case.

Methods (#CPUs=22)	Avg. linear iterations / Newton	Total solver time (secs)	% of solver time / total run time
LSPS	12.6	4055	73.3
CPR-LSPS	2.18	2358	63.2
CPR-PAMG	1.55	9941	88.8

Table 5 - Solver statistics for the 1.3 million cell full-field model running on 22 processors (test case 5).

Note that the CPR-LSPS method has significantly reduced the solver percent in this case. To compare parallel efficiency, the model was run on 22, 44, 88 processors. The scale-up in solver speeds are summarized in Fig. 4 for the 3 solver methods.

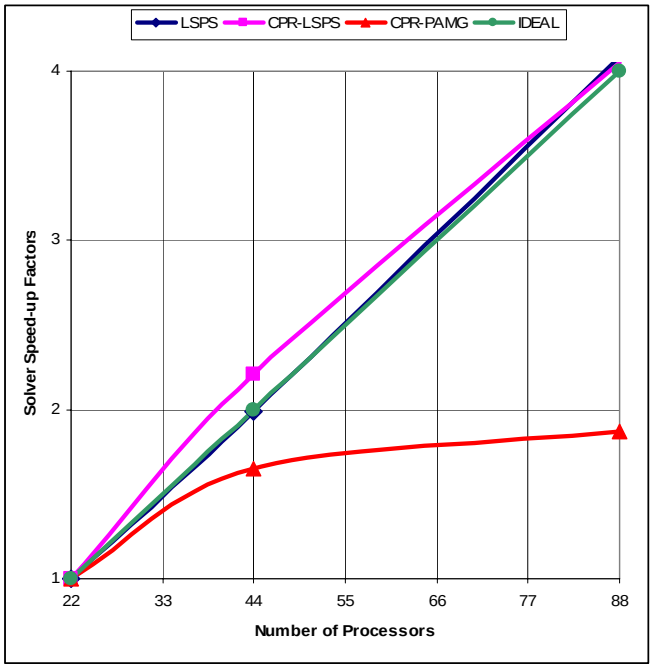

Fig. 4 - Solver scalability for the 1.3-million cell full-field model (test case 5).

Note also that CPR-LSPS is scaling superlinearly in the range of the number of processors tested. This is likely a result of the improved cache efficiency as the data arrays are partitioned into more processors. The improved cache efficiency has more than compensated for the parallelization overheads. The reason for the lower scalability of CPR-PAMG is the same as before. This model with the 50 years of history actually runs in a total time of about 15 minutes on 88 processors using the two-stage CPR-LSPS method.

Test 6: A 3.5-Million Cell Full-Field 3-phase System with Gas Cap

The grid system for this full-field model is 162*775*28 for a total of 3.52 million grid cells. The field has a large gas cap and is produced primarily from long-reach horizontal wells in the oil rim to reduce gas coning. The model has highly heterogeneous porosity and permeability fields. The areal grid size is about 275 ft by 275 ft. The simulation period was for 40 years, including both the history and prediction portions. For comparative statistics, the model was run with 60 processors. Table 6 summarizes the solver results.

Methods (#CPUs=60)	Avg. linear iterations / Newton	Total solver time (secs)	% of solver time / total run time
LSPS	1.51	2107	36,5
CPR-LSPS	1.15	2194	36.9
CPR-PAMG	1.11	10421	72.6

Table 6 - Solver statistics for the 3.5-million cell full-field model running on 60 processors (test case 6).

This problem is relatively easy for all three solver methods and requires very few iterations to achieve the desired residual tolerance. The parallel scalability tests were run using 15, 30, 60, 120, and 240 processors. The comparison of scalability for the methods is shown in Fig. 5.

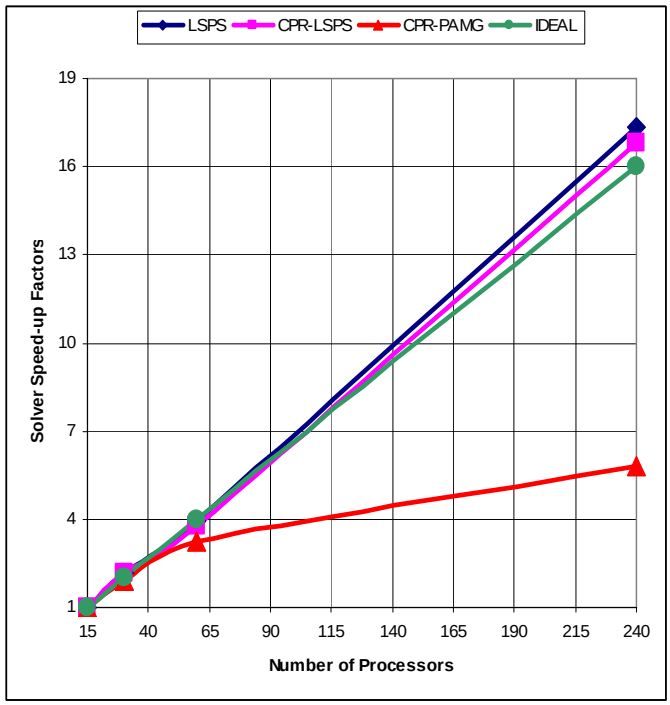

Fig. 5 - Solver scalability for the 3.5-million cell full-field model (test case 6).

Mild superlinear speed-up was observed to 240 processors with no degradation at all for the LSPS and CPR-LSPS methods. For this 3.5-million cell model, the 40 years of simulation run took a total time of 42 minutes using the one-stage LSPS method on 240 processors. The total time was 46 minutes for the two-stage CPR-LSPS method.

Test 7: A 38.2-Million Cell Full-Field Model of Ghawar

The grid system for this three-phase black-oil model is 900*2498*17 for a total of 38.2 million grid cells. It contains the entire Ghawar complex which is the largest conventional reservoir in the world. The model contains >3000 wells with over 60 years of historical data. The areal grid cell dimensions of the model are 400ft * 400ft with 17 layers. Thus, at 38 million cells, this is a relatively coarse grid model for this field. At this resolution, the simulation model is not a particularly difficult problem for the solver methods discussed. The model was run using the one-stage LSPS method on 240 processors. The 60+ years of history was completed in 8 hours 12 minutes which includes all the I/O and miscellaneous processing time. Table 7 shows the computational statistics for this problem.

38.2-million cell full-field Ghawar	Run time distribution (hrs)
Linear solver	2.2
Jacobian matrix building	2.9
Update + property calc.	0.7
I/O + others	2.4
Total	8.2

Table 7 - Run time distribution for the full history run of the 38.2-million cell full-field Ghawar model using the one-stage LSPS method (test case 7).

Conclusions

Two new unstructured parallel solver preconditioning methods have been described. The first method is based on matrix substructuring which we called line solve power series (LSPS). The second method is based on variable partitioning which is known as constraint pressure residual (CPR). These two methods can be combined to produce the two-stage CPR-LSPS method. We also discussed the two-stage CPR-PAMG method where parallel algebraic multigrid is use as the pressure-solve preconditioner.

1. Unstructured parallel LSPS is a flexible parallel, robust, and highly scalable preconditioning strategy. It includes the z-line Neumann series method, which we introduced in earlier publications, as a special case. However, LSPS is more robust and general.
2. The strength of LSPS can be adjusted by including more terms in the P matrix and by increasing N terms in the power series. Within this novel framework, we discussed the basic procedure and some practical algorithms on how to construct P using maximum connectivity graphs (MCG) and MCG line bundling.
3. The two-stage CPR-LSPS preconditioning method is described and compared with the two-stage CPR-PAMG method. For many practical megacell problems, it is shown that CPR-LSPS is a very competitive solver method.
4. PAMG has excellent convergence rate for M-matrix and is well suited as a pressure solver. However, it is more difficult to achieve comparable parallel performance to the LSPS method.
5. PAMG is attractive for situations where the PAMG set-up can be reused multiple times or when alternative methods require significant iteration counts to achieve the desired residual tolerances.
6. The robustness, performance, and scalability of the various methods are discussed and contrasted via several megacell models. Two public domain test data sets, the SPE1 model modified by uniform refinement and the SPE10 fine grid model, are also included in the comparative results.

Acknowledgements

The authors would like to thank Saudi Aramco management for the permission to publish this paper. We also thank J.R. Wallis for the many useful discussions on the CPR method. The PAMG method in this paper was modified from Lawrence Livermore National Laboratory (LLNL) solver package HYPRE option BOOMERAMG. We also thank R. Falgout and U. Yang of LLNL for many useful communications.

References

1. Dogru, A.H, Sunaidi, H.A., Fung, L.S., Habiballah, W.A., Al-Zamel, N., and Li, K.G.: "A Parallel Reservoir Simulator for Large-Scale Reservoir Simulation", SPE Reservoir Evaluation & Engineering, Feb. 2002, pp. 11-23.
2. Fung, L.S. and Dogru, A.H.: "Efficient Multilevel Method for Local Grid Refinement for Massively Parallel Reservoir Simulation", M-28, Proceedings of the ECMOR VII, Lago Maggiore, Italy, Sep. 5-9, 2000.
3. Fung, L.S. and Al-Shaalan, T.M., "Parallel Iterative Solver for the Dual Porosity Dual Permeability System in Fractured Reservoir Simulation", IPTC 10343, Proceedings of the International Petroleum Technology Conference, Dohu, Qatar, Nov. 21-23, 2005.
4. Wallis, J.R.: "Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration", SPE 12265, Proceedings of the SPE Reservoir Simulation Symposium, San Francisco, Nov. 15-18, 1983.
5. Wallis, J.R., Kendall, R.P., and Little, T.E.: "Constrained Residual Acceleration of Conjugate Residual Methods", SPE 13563, Proceedings of the 8th SPE Reservoir Simulation Symposium, Dallas, Feb. 10-13, 1985.
6. Gratien, J.M., Magras, J.F., and Quandalle, P.: "Introducing a New Generation of Reservoir Simulation Software", P013, Proceedings of the ECMOR IX, Cannes, Aug. 30 – Sep. 2, 2004.
7. Cao, H., Tchelepi, H.A., Wallis, J., Yardumian, H.: "Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation", SPE 96809, Proceedings of the SPE Annual Technical Conference, Dallas, Oct. 9-12, 2005.
8. Odeh, A.S.: "Comparison of Solutions to a Three-Dimensional Black-Oil Simulation Problem", SPE 9723, JPT, January 1981, pp 13-25.
9. Christie, M.A., Blunt, M.J.: "Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques", SPE 72469, Proceedings of the SPE Reservoir Simulation Symposium, Houston, Feb 11-14, 2001.
10. Saad, Y.: "Iterative Methods for Sparse Linear Systems", SIAM, 2nd Edition, Philadelphia, 2003.
11. Ruge, J.W. and Stuben, K: "Algebraic Multigrid", in S.F. McCormick (Ed.), Multigrid Methods, SIAM, Philadelphia, 1987.