SPE 103578

Next Generation Reservoir Simulation Using Russian Linear Solvers

B.L. Beckner, SPE, A.K Usadi, SPE, and M.B. Ray, ExxonMobil Upstream Research Co., and O.V. Diyankov, NeurOK Techsoft

This paper was prepared for presentation at the 2006 SPE Russian Oil and Gas Technical Conference and Exhibition held in Moscow, Russia, 3–6 October 2006.

Abstract

During the late 1990's, Exxon and Mobil had each independently developed next-generation reservoir simulation systems. Both next-generation systems embodied a substantial number of step-out simulation technologies, which were extremely complementary. ExxonMobil moved aggressively to combine the best of both companies' technologies into one industry-leading simulation system called EM^{power} ™. This new simulation system is currently being used to actively manage ExxonMobil's global resource base. Key features of this industry-leading simulator are described in this paper.

The new simulator employs unstructured grids to more accurately model complex geologic features, near-wellbore flow, and aquifer support. The computations are performed within the unstructured grid fabric. Interactive simulation ties together the geologic and reservoir simulation models with production data yielding high-confidence forecasts of future performance.

The linear systems that are created by an unstructured grid cannot be efficiently solved by conventional solution methods normally applied to simulators using rectilinear gridding. As a result, ExxonMobil looked to collaborate on research and development activities in the area of linear solvers for unstructured grids area and found capable research and development peers in Russia. The technical capabilities of the Russian scientific community were a natural fit to help develop a new library of linear solution methods geared towards the matrices created by the EM^{power} unstructured simulator. This collaborative effort yielded an unstructured linear solution library called SparSol and is the result of over 20 work years of effort at four Russian research organizations and contains numerous methods for efficiently solving the unstructured matrices produced by the EM^{power} unstructured reservoir simulator.

Introduction

Reservoir simulator development has historically been an active area of internal research and development at Exxon and Mobil.[1-6] During the late 1990's, Exxon and Mobil each independently developed next-generation reservoir simulation systems. ExxonMobil moved aggressively to combine the best of each company's technologies into one reservoir simulation system called EM^{power}. This new simulation system is now being used to actively manage resources within ExxonMobil's global asset base.

Potentially the most significant innovation adopted in the new simulator is unstructured gridding, with all associated computations performed within an unstructured framework. A number of researchers have contributed to developments in unstructured gridding over the past two decades[7-9]. Advances in solver and gridding technologies have also made it possible to create an unstructured reservoir simulator that is appropriate for commercial use.

As the accuracy and detail of data input to reservoir simulators continues to improve, unstructured gridding can be used to represent these features with greater fidelity than Cartesian-based reservoir grids. The increasing use of geologic modeling software programs has made the creation of detailed, multi-million cell geologic models a routine occurrence. These geologic modeling packages allow geologists to create complex, three dimensional shapes that need to be reproduced in the reservoir simulation grid. Slanted faults, intersecting faults, deviated unit boundaries and pinchouts combine to create topologies that are not suited to rectilinear grids. To this geologic complexity add the requirement to incorporate slanted and helical well geometries through the reservoir and the advantages of an unstructured grid are clear.

A key business driver in the design of the simulator is the need to reduce the overall turnaround time between formulation of the simulation problem and the generation of meaningful results. Also key is the need to have reservoir simulation used by the broadest reservoir engineering community, from entry-level reservoir engineer to senior reservoir simulation staff, on the widest range reservoir engineering evaluations, from single well workovers to rapid screening and assessment of large prospects to full-field models for development and depletion planning.

™ EM^{power} is a trademark owned by ExxonMobil Upstream Research Company.

Reservoir engineers of all skill levels are able to apply reservoir simulation technology to a range of applications through EM^{power}'s modern, intuitive Graphical User Interface (GUI). A comprehensive GUI is a generally new feature for a reservoir simulator and its creation is a significant undertaking in its own right; it required a software development effort on the same order as the simulator's computational engine. The GUI facilitates rapid and efficient data input of small-to-average sized models as well as having a range of batch-like processing capabilities that facilities workflows when modeling fields with many hundreds of wells.

Enterprise level data management is included as part of the EM^{power} reservoir simulator. Simulation input data are stored in databases allowing active data management of the large amount of simulation data distributed around the world. Results metadata, also stored in the database, are linked to results files and provide the necessary data management support. The database is tightly linked to the GUI allowing users to define access rights to other simulation engineers and monitor their data usage from within the simulator's interface.

The unstructured grid of the reservoir cells are described by a node-connection graph. Furthermore, complex surface facilities are natively described in the same way. Thus, the simulator naturally handles flow between reservoir, wells, flowlines, and surface facilities in a seamless manner to provide a robust solution for coupled reservoir-facilities flow problems. Multiple fields with multiple facilities can be simulated using this integrated flow modeling capability. The practical benefit is to better model the physical constraints imposed by facility networks and optimize operation and development of single or multiple fields.

The development of efficient new solvers was required to allow the practical solution of simulations employing unstructured grids. A library of linear solvers was developed, in collaboration with Russian Institutes, to achieve practical solution of the systems of sparse unstructured matrix equations formed by the EM^{power} simulator for both serial and parallel computational platforms.

In addition to the technological advances ExxonMobil chose to develop its own proprietary reservoir simulator primarily to provide a platform for the rapid implementation of significant technical enhancements closely aligned with the company's asset base. Experience has shown that such enhancements continually arise and add significant value through improved reservoir management capabilities. The object-oriented design of the new simulator was selected to provide the necessary flexibility for future software development.

Features and General Specifications

The general architectural specifications for EM^{power} require that the simulator be computationally efficient, flexible, extensible, maintainable and able to be run in parallel on multiple platforms. These requirements led to the use of objected-oriented programming techniques and the C++ programming language, which promote the separation of functionality via non-strict encapsulation. Computational efficiency concerns about C++ were addressed early in the project, and those issues were satisfactorily resolved.

The EM^{power} reservoir simulator is not one application but a suite of inter-connected applications which can work on a single laptop or a heterogenous collection of networked machines. A view of these multiple EM^{power} applications and how they interact in shown in **Figure 1**. The EM^{power} GUI is the presentation layer for the data input and results analysis for simulation studies. It provides a secure method for users to add, modify, and share model data. When the user runs a simulation case, the GUI presents the user with a list of available computing resources, handles data movement, and manages all data controls. Specifically, it extracts the needed set of input data from the database and writes a platform independent compressed file. EM^{power} then launches the simulation engine locally or remotely and informs the computational engine where the data file resides in the file server. The computational engine writes results and log files to the same file server. The user may monitor simulation progress and results using the GUI via TCP across the network or via the files. Some of the GUI results analysis requires the generation of a meta description to be imported into the database while other results may be analyzed via the use of analysis tools integrated into a web browser .

This multi-tiered software architecture affords the user significant flexibility in using system resources efficiently and managing large numbers of simulation models across a geographically distributed workforce.

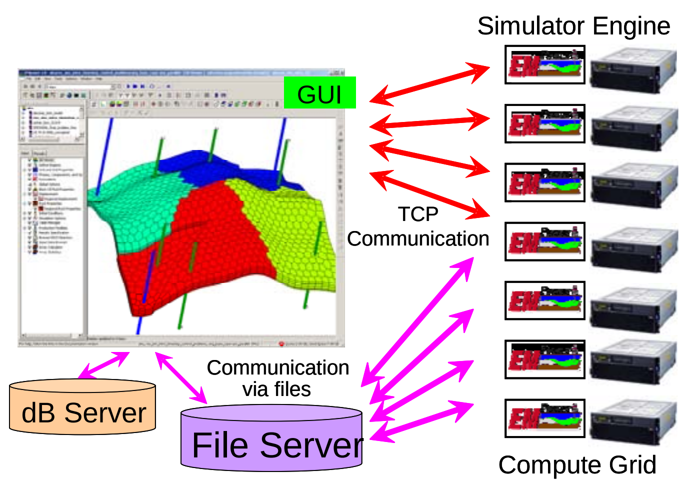

Figure 1 - EMpower Architecture

EM^{power} was designed from the ground up to be a totally unstructured reservoir simulator with respect to grid cells and grid properties. Unstructured gridding allows the grid spacing to flexibly vary according to the modeling need. The unstructured framework also provides a natural architecture for directly coupling a surface facility model with the reservoir model, increasing the computational stability of the entire system. When viewed in a general unstructured framework, the integrated facility network is treated as a set of nodes and connections that is simply an extension of the reservoir model.

EM^{power} parallelism is focused on shared-memory multiprocessor machines in order to take advantage of the fast data accesses these class of parallel machines offer. Parallelism is achieved via a proprietary messaging service using pthreads. Ensuring that this software design is also compatible with unstructured gridding and adaptive, localized modeling required the development of a complex but efficient data management infrastructure. As a design principle, the parallel computational algorithms do not impose restrictions on the selection of the grid employed.

EM^{power} provides users with an easy-to-use and comprehensive graphical interface to build, manipulate, and analyze reservoir simulation model input data and results. Interactive, 3D visualization of simulation input data and results are available from within the GUI. A comprehensive embedded on-line help and tutorial system assists novice users in getting started with simulation and provides expert users the ability to review less frequently used features as needed. Users also are able to monitor the progress of simulation runs in real time using interactive simulation monitoring tools. A batch queuing system is built into the application to allow management of multiple jobs.

The facility networks are graphically and interactively constructed via "Visual Facility Management". Operational strategies to manage facility networks (wells, pipelines, platforms) during a simulation run are implemented via well management logic. Well management logic is a custom programming language that allows very detailed operating instructions for the various facility objects during the simulation run. There are many ways the user can construct well management logic for the facility network, from explicitly coding all the actions and constraints for a particular operating strategy to using predefined libraries of commonly used operating strategies (like voidage replacement) to specifying only high-level strategies and constraints, like minimize water, and using EM^{power}'s integrated optimization capability.

Other EM^{power} features include integrated surface networks, special fluid models for water vaporization, various formulations for miscible and near-miscible displacements, rock compressibility options, and multiple well options including non-Darcy flow. The SparSol library provides an extensive suite of numeric tools to efficiently solve the unstructured matrices produced by an unstructured simulator. The next sections will highlight the use of some of the novel capabilities of the EM^{power} simulator.

Unstructured Grids

A key element of the new simulator is the use of unstructured grids. The use of unstructured grids enables more accurate and detailed representation of complex geologic and engineering features such as faults, pinchouts, fluid contacts, and horizontal or multi-lateral wells. The unstructured nature of the EM^{power} simulator means that the familiar concepts of I-J-K indexing are no longer relevant to describe grid nodes or locations of grid properties. This change influences code design, reservoir engineers' preparation of input data, and design and construction of simulation models. One practical outcome of the move to an unstructured grid is the heavy reliance on interactive, 3D visualization and associated tools to crop, isolate and interrogate data volumes with no regular frame of reference. The simulators data structure is based on a nodes and connections framework.

Figures 2 and **3** show images of simulation grids that illustrate various uses of unstructured gridding. Incorporating actual topological shapes of faulted surfaces, though inherently complex becomes tractable using unstructured grids. Extensive use of the ability to grid to faults was employed for the reservoir seen in Figure 2. As the fault surfaces of this figure are not orthogonal or parallel a Cartesian representation of these faults must be an approximation to the actual modeled geology.

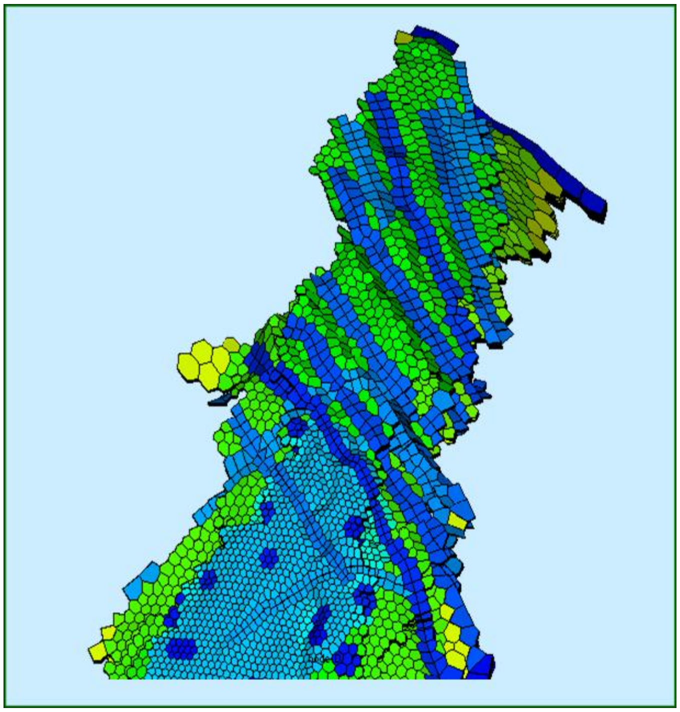

Figure 2 - A Highly Faulted Reservoir Represented By Unstructured Grids

Unstructured gridding in EM^{power} allows a fit-for-purpose discretization of each reservoir adapted to the specific situation being modeled. This approach avoids the work-arounds needed to force-fit a grid within the constraints of a hierarchically structured Cartesian gridding system of the kind used by conventional simulators. Unstructured gridding provides a more natural framework for modeling complex wells, e.g. slanted, horizontal, and multi-lateral wells, while honoring the geology. The reservoir in **Figure 3** is an example of a meandering channel sand with horizontal wells placed along channel. Finer grids are used along the channel sand and in the local vicinity of the horizontal wells. Using a finer grid only where needed employs computational resources most effectively. For detailed understanding of behavior in a particular region, localized grid refinement provides flow details without impacting the model in other connected regions. Grids in areas of lesser interest are easily coarsened. General local grid refinement is another beneficial result of an unstructured grid. It is straightforward to create a varying grid density throughout the field to seamlessly mesh fine grids around wells to very coarse grids in places like aquifer cells.

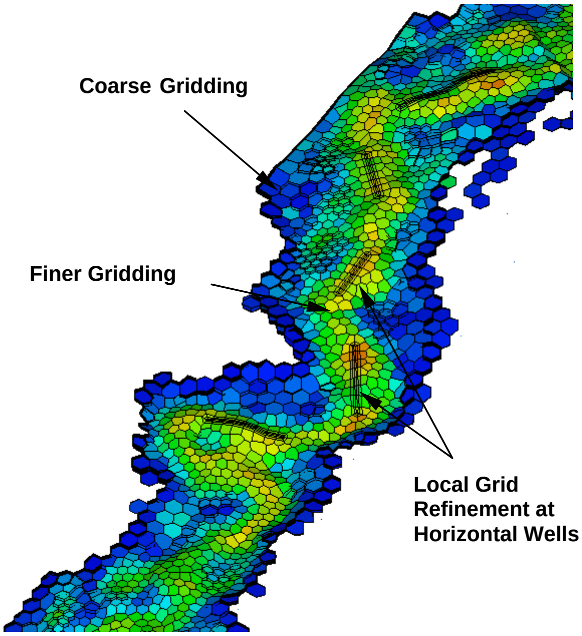

Figure 3 - An Unstructured Grid of a Meandering Channel

As is well known, flexibility in gridding typically comes at the expense of some additional effort in the solution method. However, in practice we have found there are efficiencies and simplifications associated with areal grid refinement, handling of pinch-outs, and the elimination of inactive cells, that can more than offset the increased work of an unstructured solver. It is expected that as solution methods for unstructured grids continue to improve and the complexity of geologic models continues to increase, unstructured gridding will become even more advantageous.

Surface Networks and Multi-Field

The mathematical model in EM^{power} for subsurface flow which uses an array of pressure nodes and the connections between those nodes, allows the user to selectively include as an integral part of the simulation some or all of the production and injection facilities (e.g. wells, tubing, surface flowlines, manifolds, separators). An integrated reservoir and surface facility modeling capability is attractive for two reasons. First, it allows more accurate modeling of the entire reservoir/facility system. Second, this implicit solution option provides improved stability by circumventing problems associated with rapid pressure and saturation changes near wells that are often encountered in conventional reservoir modeling approaches where surface facility modeling is either absent or entirely explicit.

This capability allows all the facility components of wellbore to surface, flowlines and riser, to be included in the computational model. For multiple fields the interaction between facility rates and pressures is implicitly solved at each time step. This capability is also important for multi-lateral well modeling where pressure drops in wellbores and the interaction between well branches are important.

EM^{power} provides a graphical interface for defining the integrated facility network. This interface displays the facility network as a collection of icons attached to one another via connection arrows. An example of a facility network diagram for the a three-field model is shown in **Figure 4**. Different icons are used to represent distinct facility types such as wells, separators, network junctions and manifolds. The connection arrows represent flowlines between the "node-like" facilities.

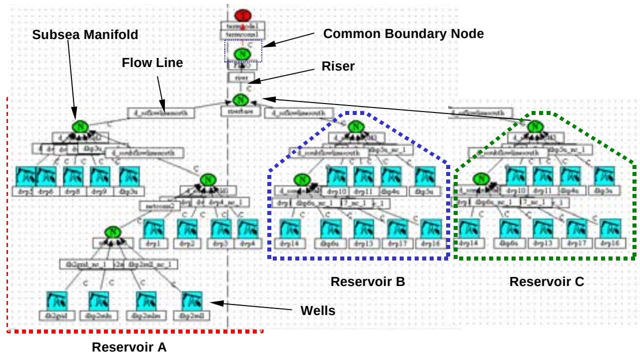

Figure 4 – Integrated Surface Facility Representation

The flow rates and pressure drops within the surface facilities can be accurately modeled if the user chooses to integrate more of the facility network into the calculation model. Facility management is fully integrated with the facility network layout so that each node or connection in the facility network can be queried and modified in various ways during the simulation.

The EM^{power} facility network data design was developed to maximize flexibility and extensibility so that facility types can be extended and new facility types added without requiring extensive software changes. Facilities are defined "generically" and contain "attributes." The attributes and much of the specialization for given facility types are defined in a data file, not in compiled code. With this design, extensions to the facility network functionality, such as adding new facility attributes or defining new facility types, can be made simply by changing the data file.

EM^{power} supports import of a variety of facility network data from external sources. Historical or time-varying data, including well perforation intervals, production rates, injection rates and pressure measurements, can be imported from text-delimited files. Such data are automatically converted to the required boundary conditions in the simulation model, saving the user time and effort when constructing a simulation model or developing sensitivity cases.

A natural extension of the integrated subsurface/surface model is the inclusion of a multi-field capability. EM^{power}'s multi-field simulation capability emphasizes ease-of-use. With just a few clicks of the mouse, individual field simulation models can be merged into a multi-field simulation model. EM^{power} automatically handles the time-consuming tasks of creating a uniform timeline, merging input data, facility networks and facility management logic, and dealing with duplicate names in regions, wells, facilities, etc. The multi-field model is ready to run as soon as the engineer adds the facility network components (that connect the fields together) and updates the field-level rates and constraints.

Facility Management Logic

Another significant aspect of EM^{power} is the use of programmable facility management. ExxonMobil has advocated this approach to simulator facility control for many years[10]. One goal of the new reservoir simulator is to allow all reservoir engineers to have extensive capabilities for constructing, sharing and customizing facility management routines. This has led to the introduction of a "Visual Facility Management" capability illustrated in **Figure 5**. "Visual Facility Management" within the user interface comprises two views: facility network view and logic diagram view. The logic diagram view serves as a way for users to develop and execute Facility Management Logic (FML). FML is a customized sequence of steps used to monitor and control the behavior of facilities throughout the simulation run. The FML functionality is flexible and powerful enough to enable an engineer to model and solve complex reservoir development schemes, yet has a convenient graphical user interface that is both intuitive and informative.

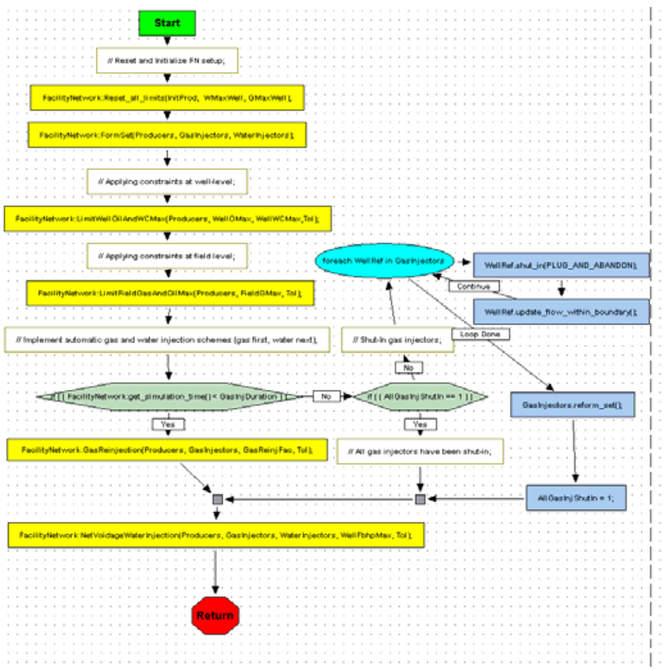

Figure 5 – Programable Logic Control of Simulation Facilities

When the user is ready to launch a simulation, EM^{power} converts the logic into standard C++ code. This C++ is sent to the simulator as part of the simulation input file, where it is compiled and dynamically linked into the simulator at run time.

Linear Solvers for Unstructured Matrices

SparSol is a powerful and efficient library of iterative solvers intended for solving large sparse linear systems of equations of the form

$$A \cdot x = b$$

The matrix *A* can be either symmetric or unsymmetric. The system can have a block multi-component structure. It is common practice to use preconditioned iterative methods with preprocessing to solve such large linear systems. The SparSol code is implemented in serial, SMP and MPI variants and can be run under a variety of operating systems. A set of program interfaces provides an easy-to-use access to the package.

The SparSol code has been developed through collaboration between ExxonMobil and NeurOK TechSoft with close cooperation with research teams from the Institute for Numerical Mathematics of the Russian Academy of Sciences (INM RAS) and the Computing Center of the Russian Academy of Sciences (CC RAS). This collaboration involved regular visits to both sites to maintain clear communication of goals and progress and to enable detailed technical discussions of implementation and performance optimization.

The effectiveness of algorithms is illustrated by numerical results performed for a set of seven test matrices from the ExxonMobil-NeurOK TechSoft matrix collection available on the site http://www.aconts.com/XOMMatrices. General information on these matrices is presented in the **Table 1**. Liberal exchange of unstructured matrices created by EM^{power} was an important component of the collaboration and required specialized code development in the simulator to facilitate this exchange.

Table 1 General information on matrices from ExxonMobil-NeurOK Techsoft collection

Matrix Name	Number of Rows	Number of Non Zeroes
CIT-1	17436	344245
CIT-2	249428	5613948
SBO-1	21700	145121
SBO-2	111756	888190
SBO-3	216051	1849317
SBO-4	93264	667882
SEO-1	22421	204784

The SparSol library contains many variants of scalings, reorderings, preconditioners and iterative methods. The application of linear solution methods to unstructured matrix systems for reservoir simulation problems is a new field so definitive knowledge of the most efficient solver methods for these new types of simulations does not exist.

System Preprocessing - Scalings/Reorderings

In many cases preprocessing the matrix system can significantly improve the convergence of a preconditioned iterative solver. During this step the original matrix is transformed to one whose properties, like matrix profile and scalability, are more conducive to a fast residual reduction. Preprocessing algorithms can be divided into two categories: scalings and reorderings. In principal, one is able to apply any combination of preprocessing algorithms, but in practice, one scaling and one or two reorderings is usually sufficient.

The main purpose for any scaling procedure is to equalize or normalize, in some way, the magnitudes of matrix entries by row and or column. Scaling can be particularly useful when applied to matrices which derive from problems that contain multiple, different material component unknowns and, hence, rows with significantly different scales.

The following scalings were tested on the EM^{power} matrices.

- *Norm scaling* makes row/column norms of the scaled matrix equal to 1;

- *Diagonal scaling*[11] - makes diagonal entries of the scaled matrix equal to 1;
- *Row-Column scaling*[12] approximately equalizes both row and column norms to 1.

Norm scaling has four variants.

- *Row norm scaling* makes all row norms equal to 1.
- *Column norm scaling* makes all column norms equal to 1.
- *Row then Column norm scaling* makes all column norms equal to 1 applying successively Row and then Column norm scaling.
- *Column then Row norm scaling* is like Row then Column norm scaling only in reverse order.

Solution results for the test matrices using these scalings are presented in **Table 2**. In **Table 2** we reference each matrix by its number in **Table 1** and emphasize the best result for each matrix. The main criterion for comparison of the variants is the total solution time. The performance for the default variant, without any preprocessing is 1.0 and the relative performance to the default variant is presented in Table 2 (and all subsequent performance tables). Note also, that the maximal number of BiCGStab iterations is set to 300. A method is considered as diverged if the required accuracy was not achieved for less than 300 iterations; this is indicated by a '-' in the tables.

Table 2 - Performance of Norm Scaling Options

	Row Scaling		Column Scaling		Row+Col Scaling		Col+Row Scaling	
Matrix	Iters	t(%)	Iters	t(%)	Iters	t(%)	Iters	t(%)
CIT-1	25	**1.00**	104	2.53	22	1.06	25	1.14
CIT-2	300	-	300	-	52	**0.68**	207	2.18
SBO-1	175	1.11	167	1.10	194	1.26	158	**1.05**
SBO-2	112	0.86	106	0.81	119	0.89	101	**0.79**
SBO-3	154	0.82	152	**0.81**	173	0.91	157	0.85
SBO-4	120	1.10	111	**1.03**	117	1.10	125	1.16
SEO-1	87	1.07	69	**0.85**	65	**0.85**	91	1.13

Diagonal scaling makes diagonal entries of the scaled matrix equal to 1. After diagonal scaling an iterative balancing procedure may be applied.

Row-column scaling approximately equalizes both row and column norms to 1. This scaling is effective for most of the tested systems, but especially for systems containing variable unkowns of significantly different scacles. The results on the solution of the test matrices for Row-Column and Diagonal scalings are presented in **Table 3**.

Table 3 - Performance For Row-Column and Diagonal Scalings

	Row-Column		Diagonal	
Matrix	Iters	Time (%)	Iters	Time (%)
CIT-1	21	**0.89**	38	1.33
CIT-2	83	1.00	154	1.68
SBO-1	216	**1.24**	222	1.27
SBO-2	135	0.94	126	**0.89**
SBO-3	197	0.98	147	**0.77**
SBO-4	119	**1.10**	141	1.25
SEO-1	99	1.22	96	**1.15**

Reordering can significantly improve the quality of a preconditioner and accelerate the convergence of the iterative method. The following reorderings are available in SparSol.

- RCM implements the well-known Reverse Cuthill-McKee reordering.
- Weighted RCM implements a Weighted Reverse Cuthill-McKee reordering proposed by I. Kaporin
- Multicoloring is a Greedy Multicoloring Algorithm
- Diagonal Filtration
- Diagonal Dominant Filtration
- Independent Graphs Reordering searches for independent sub-graphs of a matrix, joins "small" sub-graphs into bigger ones, and then reorders them in descending order. This is particularly useful when the matrix represents a physical system with sub-regions that are not in communication with each other.
- Cell Based Reordering treats each multi-component cell of a matrix system as one element to be re-ordered using any of the structural reordering methods discussed above (e.g. RCM).

RCM-like reorderings were observed to be most effective in helping improve convergence performance on the problems. Furthermore, reordering is usually most effective in combination with scaling. This is shown in **Table 4**.

Table 4 - Performance of RCM Variants

	RCM		RCM + Row-Column		WRCM+Row-Column	
Matrix	Iters	Time(%)	Iters	Time(%)	Iters	Time(%
CIT-1	27	1.33	13	**0.81**	16	0.89
CIT-2	300	-	33	**0.51**	39	0.73
SBO-1	211	1.29	210	**1.19**	249	1.47
SBO-2	102	0.82	120	0.81	103	**0.74**
SBO-3	139	0.66	183	0.76	132	**0.60**
SBO-4	100	0.89	99	**0.86**	101	0.92
SEO-1	62	0.87	78	0.93	49	**0.72**

SparSol Preconditioners

Preconditioning is a key procedure for the iterative solution of large sparse liner systems. The convergence rate of the iterative method greatly depends on the quality of the preconditioner.

The preconditioners available in SparSol can be divided into the three classes:

- Incomplete Cholesky-type preconditioners construct the matrix M as $A \approx M = U^T \cdot U$ where U is an upper triangular matrix. These preconditioners are especially effective for SPD matrices, but cannot be applied to unsymmetric matrices.
- Incomplete LU-type preconditioners construct the matrix M as $A \approx M = L \cdot U$ where L is a strictly lower triangular matrix with unit diagonal and U is an upper triangular matrix. These preconditioners can be applied to general sparse matrices.
- Nested-type factorizations can be applied to matrices having a special nested or multi-level structure. Such matrices most easily arise in problems using finite difference discretization on structured grids.

ILU-type preconditioners are appropriate for the solution of sparse, non-symmetric matrices produced by the EMpower simulator. Preconditioners of this type construct the matrix M

as $A \approx M = L \cdot U$ where L is a strictly lower triangular matrix with unit diagonal and U is an upper triangular matrix.

It is well-known that complete LU factorization can take an enormous amount of computational and memory resources. Rules to drop "non-essential" entries during factorization are important in making these methods practical. SparSol discards elements whose magnitudes are less than a given constant or dropping tolerance multiplied by the norm of the corresponding row.

The following ILU-type preconditioners are implemented in SparSol :

- ILU(0)[13] is a straightforward variant of Incomplete LU factorization.
- RILU is the relaxed variant of Incomplete LU factorization without extension of the original pattern where discarded entries multiplied to the some relaxation coefficient ω are subtracted from the corresponding diagonal entry.
- FILU is a high quality preconditioning combining several preconditioning techniques and several dropping strategies. The basic FILU algorithm extends the original pattern of a matrix using two drop tolerances - one for adding new entries into preconditioning stencil and another for dropping sufficiently small new entries from the preconditioned row.
- ILU2 developed by I.Kaporin from CC RAS implements high quality preconditioning of a general sparse matrix based on its $LU + LR_1 + R_2^T U$ decomposition.

Iterative Methods

The following iterative methods are implemented in SparSol:

- CG - Conjugate Gradient[14]
- CGS – Conjugate Gradient Squared[15]
- BiCGSTab – BiConjugate Gradient Stabilized[16]
- BiCGSTab2 – Biconjugate Gradient 2 Stabilized[17]
- BiCGSTabL – BiConjugate Gradient Stabilized (L)[18]
- GMRes – General Minimal Residual[19]

BiCGStab exhibits fast and smooth convergence for most systems tested and is therefore SparSols default iterative solver. BiCGStab2 also shows good performance and can be chosen as an alternative of BiCGStab.

Comparison to Other Packages

The performance of the SparSol has been compared to that of several popular iterative solvers SparseKit2[20], PETSc[21] and with the direct solver PARDISO[22] for seven test problems from the ExxonMobil-NeurOK TechSoft matrix collection. General information on these matrices is presented in **Table 1**.

For the results presented in this paper, calculations were performed on an AMD Opteron computer with two 2GHz CPUs and 4GB RAM under SuSe 8.0. All programs were compiled with GCC 3.4.1. Furthermore, the solution procedure used the BiCGStab iterative method with stopping criterion $\left| r_i = b - A \cdot x_i \right|_2 < 10^{-5} \cdot \left| b \right|_2$ and a maximum of 300 iterations.

Table 4 contains comparative results for the ILU(k) preconditioner from PETSc, Table 5 for ILUT from SparseKit2 and Table 6 for the PARDISO solver with default parameters. SparSol's performance is very competitive with optimized results from these solver packages.

Table 4 – SparSol FILU and PETSc ILU(k) Results

Matrix	K	Iters	T_{ILUK}/T_{FILU}
CIT-1	1	26	2.61
CIT-2	-	-	-
SBO-1	1	74	1.38
SBO-2	3	163	7.10
SBO-3	3	530	19.53
SBO-4	3	606	36.57
SEO-1	4	315	49.00

Table 5 - SparSol FILU and SparseKit2 ILUT Results

Matrix	Iters	T_{ILUK}/T_{FILU}
CIT-1	300	-
CIT-2	300	-
SBO-1	46	1.30
SBO-2	53	1.52
SBO-3	75	1.22
SBO-4	35	1.42
SEO-1	63	1.96

Table 6 - SparSol FILU and PARDISO Results

Matrix	LU Extension	$T_{PARDISO}/T_{FILU}$
CIT-1	19.35	7.94
CIT-2	-	-
SBO-1	14.86	1.41
SBO-2	29.22	2.35
SBO-3	37.64	3.11
SBO-4	35.46	4.90
SEO-1	29.19	4.81

Conclusion

This paper highlights a number of distinguishing, next-generation reservoir simulation technologies including computations made on a native, unstructured grids, integration of facilities and models for multiple fields, and highly customizable facility/well management logic to provide engineers and geoscientist the technologies necessary for best-in-class asset management. EM^{power} is now the standard reservoir simulator for ExxonMobil and is the vehicle for the delivery of proprietary technologies of particular relevance to our assets that have been or will developed in the future.

As part of this next-generation reservoir simulator development a numerical library of scaling and reordering methods, preconditioners and iterative methods for large, sparse unstructured matrices has been developed. Results presented in the paper show that SparSol solver is faster that several popular, freely available packages for the set of matrices tested. Results also showed that no single combination of scaling, orderings, preconditioners and iterative methods produce the most efficient result.

The SparSol development is a successful example of collaboration between research groups in different parts of the world. Exchange of significant amounts of information, facilitated by world-wide computing networks, was key to producing a production quality product in current use on a wide variety of petroleum assets.

Acknowledgements
The authors wish to acknowledge the contributions of S.V. Koshelev, S.S. Kotegov, I.V. Krasnogorov, N.N. Kuznetsova and V.Y. Pravilnikov of NeurOK Techsoft. The authors also acknowledge the management of ExxonMobil Upstream Research Company for permission to publish this paper.

References
1. Kendall, R.P., Morrell, G.O., Peaceman, D.W., Silliman, W.J. and Watts, J.W.: "Development of a Multiple Application Reservoir Simulator for use on a Vector Computer," presented at the Middle East Oil Technical Conference of the Society of Petroleum Engineers, Bahrain, March 14-17, 1983.
2. Harper, J.L., Heinmann, R.F., Ray, M.B. and Stephenson, P.E.: “A Compositional Simulator For Performing Large Field Studies in a Vector Computing Environment,” SPE 13714, presented at the 4th SPE Middle East Oil Technical Conference, Bahrain, March 11-14, 1985, 279-290.
3. Watts, J.W.: “A Compositional Formulation of the Pressure and Saturation Equations,” *SPE Reservoir Engineering*, vol. 1, no. 3. May 1986, 243-252.
4. El-Mandouh, M.S., Bette, S., Heinemann, R.F., Ogiamien, E.B., Bhatia, S.K.: “Full-Field Compositional Simulator of Reservoirs With Complex Phase Behavior,” SPE 25249, presented at the 12th SPE Reservoir Simulation Symposium, New Orleans, Louisiana, February 28 – March 3, 1993, 167-186.
5. Beckner, B.L., Chan, H.M., McDonald, A.E., Wooten, S.O. and Jones, T.A.: “Simulating Naturally Fractured Reservoirs Using a Subdomain Method,” SPE 21241, presented at the 11th SPE Reservoir Simulation Symposium, Anaheim, California, Feburaruy 17-20, 1991, 383-396.
6. Mifflin, R.T., Watts, J.W. and Weiser, A.: “A Fully Coupled, Fully Implicit Reservoir Simulator For Thermal and Other Complex Reservoir Processes,” SPE 21252, presented at the 11th SPE Reservoir Simulation Symposium, Anaheim, California, Feburaruy 17-20, 1991, 457-470.
7. Hienemann, Z.E.: "Advances in Gridding Techniques," presented at the Fifth International Forum on Reservoir Simulation, Muscat, Oman, December 10-14, 1994.
8. Verma, S., and Aziz, K.: "A Control Volume Scheme for Flexible Grids in Reservoir Simulation," SPE 37999, presented at 1997 Reservoir Simulation Symposium, Dallas, Texas, June 8-11 1997, 215-227
9. Gunasekera, D.L., Cox J., and Lindsey, P.: "The Generation and Application of K-Orthogonal Grid Systems," SPE 37998, presented at 1997 Reservoir Simulation Symposium, Dallas, Texas, June 8-11 1997, 199-214
10. Miertschin, J.W., and Weiser, A.: "A Flexible Approach to Predictive Well Management Via User-Defined Strategies," SPE 19848, presented at 64th Annual Technical Conference and Exhibition of the SPE, San Antonio, Texas, October 8-11, 1989, 817-826.
11. Kaporin, I.E., Konshin, I.: "Optimization of Matrix Scaling in CGSOL Code, " NeurOK Techsoft Report, June 2003.
12. de Almeida, V.F., Chapman, A.M., and Derby, J.J.: "On Equilibration and Space Factorization of Matrices Arising in Finite Element Solutions of Partial Differential Equations, " *Numerical Methods Partial Differential Equations,* 16, 2000, 11-19.
13. Kershaw, D.: "The Incomplete Cholesky Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations, " *Journal of Computational Physics*,26, 1978, 43-65.
14. Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V. Pozo, R., Romine, C., van der Vorst, H. : *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods*, SIAM, 1994.
15. Sonneveld, P.: "CGS, A Fast Lanczos Type Solver for Nonsymmetric Linear Systems, " *SIAM J. Sci. Statist. Comput*., 10, 1989, 36-52.
16. van der Vost, H. : "Bi-CGSTAB : A Fast and Smooth Converging Variant of Bi-CG for the Solution of Non-Symmetric Linear Systems", *SIAM J. Sci. Statist. Comput*., 12, 1992, 631-644.
17. van der Vost, H., Chan, T.F. : Linear System Solvers : Sparse Iterative Methods
18. Sleijpen, G.L.G., and Fokkema, D.R. : "BiCGStab(L) for Linear Equations Involving Unsymmetric Matrices With Complex Spectrum, " Electronic Transactions on Numerical Analysis, 1, Sept 1993, 11-32.
19. Saad, Y. : "GMRES : A Generalized Minimal Residual Algorithim for Solving Nonsymmetric Linear Systems, " *SIAM J. Sci. Statist. Comput*., 7,1986, 856-869.
20. Saad, Y. : "SPARSEKIT : A Basic Tool for Sparse Matrix Computations, " *Tech. Rep. 90-20*, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1990.
21. PETSc : Portable, Extensible Toolkit for Scientific Computation, http://www-unix.mcs.anl.gov/petsc/.
22. Sheck, O. and Gartner, K. : "PARDISO : A High Performance Serial and Parallel Sparse Linear Solver in Semiconductor Device Simulations, " *Future Generation Computer Systems*, 789, 2001, 1-9.