Title:

Evaluation of Linear Solvers for Oil Reservoir Simulation
Problems Part 2: the Fully Implicit Case

Author(s):

Wayne Joubert
Raghunandan Janardhan

RECEIVED**DEC 16 1997****OSTI**

Submitted to:

For external distribution as requested.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

Los Alamos
National Laboratory

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. The Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

Evaluation of Linear Solvers for Oil Reservoir Simulation Problems

Part 2: The Fully Implicit Case

Wayne Joubert
Raghunandan Janardhan
Los Alamos National Laboratory
Parallel Architectures Team
Group CIC-19, Scientific Computing Group
Los Alamos, NM 87545
wdj@lanl.gov

Revision 9-97
September 1997

Abstract. A previous paper [Joubert/Biswas 1997] contained investigations of linear solver performance for matrices arising from Amoco's Falcon parallel oil reservoir simulation code using the IMPES formulation. In this companion paper, similar issues are explored for linear solvers applied to matrices arising from more difficult fully implicit problems. The results of numerical experiments are given.

1 Introduction

In [Joubert/Biswas 1997], a wide range of linear solver techniques was tested on a class of model problems simulated using Amoco's Falcon simulator under the IMPES model. In this companion paper, we perform similar linear solver experiments for fully implicit compositional simulations.

Though IMPES (implicit pressure, explicit saturation) simulations are effective in many practical oilfield situations, more complex fields such as those manifesting complicated phase change behavior require more highly accurate fully implicit simulations.

The matrices which arise from such simulations are significantly more challenging for sparse linear solvers, in particular parallel iterative solvers. The matrices are bordered, with added unknowns to account for well pressures. Furthermore, each grid cell is associated with multiple unknowns, for variables such as pressure, oil, gas and water.

Whereas IMPES matrices are typically nearly symmetric and have good diagonal dominance properties, matrices from fully implicit simulations are typically highly nonsymmetric and not diagonally dominant. These properties make it difficult to solve such problems by iterative methods, and it is commonly necessary to utilize strong preconditioning techniques, such as incomplete factorization methods with options for additional matrix fill-in.

This paper presents comparative results for various linear solvers applied to fully implicit simulations. The methodology follows along similar lines to the previous paper [Joubert/Biswas 1997].

2 Description of the Simulator

The numerical experiments in this paper were performed using the Amoco IGCAMP simulator [Amoco 1992b]. This simulator is a modification of Amoco's proprietary simulator, GCOMP [Amoco 1992a], [Young/Stephenson 1983]. This serial code has full capabilities for performing fully implicit compositional simulations for structured 3-D grids. This code was used rather than Amoco's Falcon code [Shiralkar et. al. 1997] because at the time these

experiments were performed, Falcon's fully implicit capabilities were not yet functional. It is expected that similar linear solver performance would be obtained for matrices from Falcon simulations.

3 Description of the Test Problems

Below is a set of sample datasets used with IGCMP. The datasets were selected to give a range of problem behaviors that might be encountered in fully implicit simulations, including difficult ones. Though most of the datasets have fixed problem size, the "S" dataset has variable problem dimensions, facilitating the possibility of performing scalability studies with the code.

- FRAC1GN. This is a 2-d black oil model to simulate a fracture. The model generates matrices of dimension $15 \times 23 \times 1$ times the number of unknowns per grid block. The model has 1 well. The simulation time is 100 days.
- FRAC35.C. This is a 3-d implicit black oil model to simulate a fracture. The geometry represents a quarter of a five spot pattern. The model generates matrices of dimension $18 \times 34 \times 2$ times the number of unknowns per grid block. The model has 2 wells. The simulation time is 100 days.
- GAS. This is a 3-d gas problem. The model generates matrices of dimension $8 \times 8 \times 2$ times the number of unknowns per grid block. The model has 2 wells. The simulation time is 2800 days.
- GCONEB. This is a 2-d coning simulation. The model generates matrices of dimension $1 \times 11 \times 21$ times the number of unknowns per grid block. The model has 1 well. The simulation time is 360 days.
- S. This is a 3-d black oil model. The geometry represents a quarter of a five spot pattern. The model is identical to the one used in [Joubert/Biswas 1997] except for a small change to force one of the wells to be implicit for part of the run. The model generates matrices of dimension $8 \times 8 \times 2$ times the number of unknowns per grid block; however, for some studies we will change the grid refinement in order to test scalability of linear solvers. The model has 2 wells. The simulation time is 1000 days.
- ANTONBO. This is a 3-d, 5 layer 11×11 5-spot problem. The model generates matrices of dimension $11 \times 11 \times 5$ times the number of unknowns per grid block. The model has 5 wells. The simulation time is 6174 days.
- ARSHAD. This case represents the combination of the five adjacent 7-spot patterns with no-flow boundaries around the edge. The model generates matrices of dimension $33 \times 15 \times 6$ times the number of unknowns per grid block. The model has 10 wells. The simulation time is 13140 days.
- BIGN. This is a 3-d waterflood model. The model generates matrices of dimension $37 \times 59 \times 6$ times the number of unknowns per grid block. The model has 60 wells. The simulation time is 6909 days.

4 Matrix Properties

In order to get a better understanding of the expected linear solver performance for fully implicit simulations, in this section we examine the properties of the matrices over a simulation run. For this run, the linear systems are solved to a very accurate tolerance, to disallow for any effects of the linear solver on the matrix properties for the run.

In particular, we consider the properties of matrices arising from the "S" dataset. This problem has $8 \times 8 \times 2$ grid blocks and two vertical wells. In the resulting matrix, each well gives rise to one unknown, in addition to the grid unknowns (four per grid cell), and each well is coupled to the grid variables in each block it passes through. The well-well coupling matrix is diagonal (in fact, the identity), and the grid-grid coupling matrix corresponds to a 3-D 7-point finite difference stencil with multiple degrees of freedom per grid block.

Figures 1 and 2 in the Appendix section show basic run information for the simulation. For this and all cases, the x-axis of the graph is the linear solve number. The following observations may be made for this case:

- The "Day #" and "Time step size" data indicate that the simulation began at a very small time step size and gradually increased this time step as it was determined by the code that this could be done without loss of accuracy of the simulated solution. The time step also was decreased briefly at the 500 day mark when the injection well was turned on.
- Only 1-2 Newton steps were required per time step, until injection was begun, at which time the number of Newtons increased.
- The code determines automatically at each time step which wells, if any, must be made implicit.

Figure 3 gives the magnitudes of the largest and smallest entries of different sections of the matrix. Here it is assumed that the matrix is partitioned

$$\begin{bmatrix} A_{gg} & A_{gw} \\ A_{wg} & A_{ww} \end{bmatrix},$$

where the subscripts g and w pertain to the grid and well parts, respectively. The data indicate that A_{gg} and A_{wg} contain the entries of largest magnitude. In particular, A_{wg} becomes large when the water injection begins.

On the other hand, Figure 4 shows the same information for the matrix with a left block diagonal scaling applied, where the blocks are the 4×4 diagonal blocks for the grid submatrix. Performing this scaling significantly improves the sizes of the entries of the matrix.

Figures 5 through 8 give the extremal entries of the main 4×4 -block main diagonal part of A_{gg} , where the minimum or maximum is taken across all grid cells. It should be noted that the largest entries can be found in the fourth row of the 4×4 blocks, corresponding to the pressure equation. This suggests that the CPR or Combinative methods may be effective for these problems. Also, the sizes are large even near the beginning of the run, when the time step is small and the matrix properties would be expected to be well-behaved.

Figures 9 and 10 give similar results for the off-diagonal blocks, which refer to the connections of grid blocks to their neighbors. These entries are considerably smaller in magnitude than the entries of the main diagonal blocks. Also, the entries are small at the beginning of the run when the time step is small. Similar results are given in Figures 11 and 12 for the block diagonal scaled matrix.

Figures 13 through 15 give extremal values pertaining to the connection of the wells to the four grid unknowns, for the unscaled and the scaled matrices. Apparently the largest entries are the connection of the wells to the first of the four grid components in A_{wg} .

Figures 16 through 18 denote diagonal dominance matrix properties for the unscaled and scaled matrices. These figures have to do with $\max_i \sum_{j \neq i} |a_{ij}|/|a_{ii}|$. This calculation is first performed within the main diagonal grid blocks, then within the grid matrix as a whole, then across the A_{gg}/A_{gw} submatrix, and finally across the A_{wg}/A_{ww} submatrix. The results again show that much of the difficulty in these matrices is in the main diagonal blocks, particularly the fourth equation, and can be remedied by a block scaling.

Figures 19 through 21 denote the largest absolute value of any entry in the skew-symmetric part of the matrix, $(A + A^T)/2$. This is given for the diagonal blocks and for the off-diagonal blocks for the unscaled and scaled cases, for the grid submatrix. The results show high nonsymmetry, particularly in the main diagonal blocks, and particularly in the fourth row and column of the main diagonal blocks.

Figures 22 and 23 give similar results for the well-grid part of the skew-symmetric part of the matrix.

The main result of these studies is that most of the difficulty with these matrices seem to come from the diagonal blocks of the grid matrices, and the largest of these entries are those associated with the pressure equation.

5 Performance of Some Basic Iterative Methods

The following is a basic set of experiments with linear solvers applied to the model problems. For these experiments, GMRES(s) is applied, with the maximum allowable number of iterations specified by itsmax=100. These tables give the following information for the simulations:

- The stopping tolerance ζ , assuming stopping test $\|Q_L r^{(n)}\|/\|Q_L r^{(0)}\| \leq \zeta$, where Q_L is the left preconditioner, if any.
- The type of preconditioning of the grid matrix, either none, point Jacobi, block Jacobi, or relaxed no-fill ILU preconditioning on the grid matrix.

- Whether the iteration is performed on the Schur complement system $A_{gg} - A_{gw}A_{ww}^{-1}A_{wg}$ or on the original system.
- The number of days of simulation performed.
- The ILU relaxation parameter ω ($\omega = 0$ gives ILU, $\omega = 1$ gives MILU).
- Whether the preconditioning is applied on the left or the right side. When not specified, the preconditioning is on the left.
- The GMRES restart frequency s .
- The total number of time steps in the simulation.
- The number of time steps that were forced to repeat, e.g. due to inaccuracy in the linear solve.
- The total number of linear solves.
- The total number of linear solver iterations for all solves.
- The number of linear solves that iterated to itsmax without indicating convergence according to the stopping test.
- The number of ILU factorizations which incurred a negative pivot.
- Number of negative pivot ILU factorizations which also failed to give convergence in itsmax iterations.
- The flops per unknown per linear solve incurred by the linear solver.
- The flops per unknown per linear solve incurred by the preconditioning setup phase of the linear solves.
- The flops per unknown per linear solve incurred by the preconditioning application phase of the linear solves.

ζ	prec	Schur	ω	s	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
1e-4	none	no	-	20	38	2	99	7149	54	0	0	30219	0	0
1e-4	none	yes	-	20	33	1	74	4383	30	0	0	24708	0	0
1e-4	pjac	no	-	20	38	2	99	7149	54	0	0	30512	0	293
1e-4	pjac	yes	-	20	33	1	74	4383	30	0	0	24953	4	241
1e-4	bjac	no	-	20	33	1	70	1520	0	0	0	9984	228	727
1e-4	bjac	yes	-	20	33	1	70	1630	0	0	0	10721	228	777
1e-4	ilu	no	0.	20	33	1	70	561	0	0	0	6982	1764	1983
1e-4	ilu	yes	0.	20	33	1	70	522	0	0	0	6669	1764	1861
1e-4	ilu	no	.5	20	33	1	70	566	0	0	0	7241	1977	1999
1e-4	ilu	yes	.5	20	33	1	70	526	0	0	0	6752	1812	1873
1e-4	ilu	no	1.	20	33	1	70	568	0	0	0	7259	1977	2005
1e-4	ilu	yes	1.	20	33	1	70	527	0	0	0	6762	1812	1876
1e-4	none	no	-	50	33	1	71	4551	24	0	0	40652	0	0
1e-4	none	yes	-	50	33	1	71	3650	17	0	0	32189	0	0
1e-6	none	no	-	20	38	2	99	7838	68	0	0	33135	0	0
1e-6	none	yes	-	20	33	1	74	5146	42	0	0	29066	0	0
1e-6	pjac	no	-	20	33	1	84	6433	59	0	0	32387	0	311
1e-6	pjac	yes	-	20	33	1	74	5146	42	0	0	29352	4	282
1e-6	bjac	no	-	20	33	1	70	2479	0	0	0	16043	228	1165
1e-6	bjac	yes	-	20	33	1	70	2558	0	0	0	16559	228	1201
1e-6	ilu	no	0.	20	33	1	70	715	0	0	0	8382	1764	2467
1e-6	ilu	yes	0.	20	33	1	70	672	0	0	0	8019	1764	2332
1e-6	ilu	no	.5	20	33	1	70	720	0	0	0	8644	1977	2483
1e-6	ilu	yes	.5	20	33	1	70	676	0	0	0	8106	1812	2345
1e-6	ilu	no	1.	20	33	1	70	723	0	0	0	8673	1977	2492
1e-6	ilu	yes	1.	20	33	1	70	678	0	0	0	8126	1812	2351
1e-6	none	no	-	50	33	1	71	5004	42	0	0	45259	0	0
1e-6	none	yes	-	50	33	1	71	4506	24	0	0	39846	0	0

Table 1. S dataset, $n_x \times n_y \times n_z = 8 \times 8 \times 2$.

We may draw the following conclusions from the runs for dataset S in Table 1, a relatively easy and small problem:

- The simulation problem could be successfully solved by all linear solution techniques. Difficulties occurred (e.g. unconverged linear solves or repeated time steps) due to weak preconditioning, e.g. none or point Jacobi, or a weak convergence tolerance.
- Decreasing the convergence tolerance increased the number of unconverged linear solves but decreased the number of time step repeats. In general, decreasing the tolerance, as could be expected, increased the number of linear solver iterations.
- ILU preconditioning with $\zeta = 10^{-4}$ was the most efficient solver, by up to about a factor of five. At least block Jacobi preconditioning is necessary in order to give effective solution to these problems.
- In general, it was slightly more efficient to use Schur complement than not to.
- In general, ILU was slightly more efficient than MILU.
- Increasing s increased the solver robustness but also increased the work required.

Let us now considered some more difficult simulations. The following runs assume $\zeta = 10^{-6}$.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
frac1gn	bjac	20	0	r	no	0	199	78	1293	126148	1240	0	0	44288	228	3153
frac1gn	ilu	50	0	r	no	100	97	4	249	3685	0	66	0	11316	1764	3256
frac1gn	ilu	20	0	r	no	100	97	4	249	3866	0	66	0	11638	1764	3422
frac1gn	ilu	20	.1	r	no	100	97	4	250	5431	0	228	0	15697	1812	4797
frac1gn	ilu	20	.5	r	no	100	97	4	248	15269	80	248	80	41234	1812	13644
frac1gn	ilu	20	.9	r	no	100	128	20	473	39715	359	473	359	55689	1812	18653
frac1gn	ilu	20	1.	r	no	100	178	37	715	63718	596	715	596	58984	1812	19799
frac1gn	ilu	50	0	r	yes	100	93	2	253	4001	0	71	0	11895	1764	3479
frac1gn	ilu	20	0	r	yes	100	93	2	253	4175	0	71	0	12182	1764	3637
frac1gn	ilu	20	1.	r	yes	100	182	39	786	69919	657	786	657	58854	1812	19759
frac1gn	ilu	50	0	l	no	100	93	2	253	4102	0	71	0	12408	1764	3787
frac1gn	ilu	20	0	l	no	100	93	2	253	4295	0	71	0	12711	1764	3955
frac1gn	ilu	20	1.	l	no	100	131	19	552	46861	401	552	401	56284	1812	18897
frac1gn	ilu	50	0	l	yes	100	93	2	253	4014	0	71	0	12159	1764	3710
frac1gn	ilu	20	0	l	yes	100	93	2	253	4207	0	71	0	12462	1764	3878
frac1gn	ilu	20	1.	l	yes	100	133	17	517	42401	365	512	365	54427	1812	18263

Table 2. FRAC1GN dataset.

The following observations can be drawn from the FRAC1GN dataset:

- The problem was too difficult to solve by anything less than ILU/MILU preconditioning.
- Increasing ω decreased efficiency by as much as a factor of four. $\omega = 1$ is best. Increasing ω increased the number of negative pivots, unconverged linear solves, and repeated time steps.
- In most cases, right preconditioning was slightly better than left preconditioning.
- For right preconditioning, Schur complement was slightly worse, whereas the opposite was true for left preconditioning.
- $s = 50$ was more efficient than $s = 20$, both in operation counts and linear solve iterations.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
frac35_c	bjac	20	0	r	no	6	165	64	1009	84831	778	0	0	38173	228	2716
frac35_c	ilu	50	0	r	no	100	103	26	486	12031	4	396	0	20557	1764	5449
frac35_c	ilu	20	0	r	no	100	125	36	638	25546	69	453	12	27234	1764	8845
frac35_c	ilu	20	.1	r	no	5	98	35	556	33362	258	360	205	40170	1812	13307
frac35_c	ilu	20	.5	r	no	1	89	32	516	33389	286	303	216	43225	1812	14360
frac35_c	ilu	20	.9	r	no	0	91	36	531	33643	285	320	226	42365	1812	14061
frac35_c	ilu	20	1.	r	no	0	82	31	514	33479	280	291	204	43498	1812	14453
frac35_c	ilu	50	0	r	yes	100	108	26	471	10017	0	384	0	17618	1764	4680
frac35_c	ilu	20	0	r	yes	100	108	26	462	15079	13	375	13	22478	1764	7196
frac35_c	ilu	20	1.	r	yes	0	84	33	483	33956	307	400	307	46857	1812	15610
frac35_c	ilu	50	0	l	no	100	99	23	462	12188	0	379	0	21864	1764	6024
frac35_c	ilu	20	0	l	no	100	111	29	556	17981	0	469	0	22482	1764	7335
frac35_c	ilu	20	1.	l	no	0	75	30	475	34776	323	392	321	48798	1812	16327
frac35_c	ilu	50	0	l	yes	100	96	23	455	11814	0	373	0	21557	1764	5932
frac35_c	ilu	20	0	l	yes	100	107	28	525	16844	0	443	0	22324	1764	7278
frac35_c	ilu	20	1.	l	yes	0	72	26	458	33947	314	385	314	49367	1812	16526

Table 3. FRAC35_C dataset.

The following observations can be drawn from the FRAC35_C dataset:

- The problem could be solved only by ILU preconditioning with $\omega = 0$.
- The Schur complement iteration was better than the full system iteration.
- In some cases left preconditioning was more efficient, and in some cases right preconditioning.
- $s = 50$ was more efficient than $s = 20$, both in operation counts and linear solve iterations.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
gas	bjac	20	0	r	no	2800	207	0	272	14396	0	0	0	23844	228	1695
gas	ilu	50	0	r	no	2800	207	0	272	3312	0	0	0	9334	1764	2679
gas	ilu	20	0	r	no	2800	207	0	272	3312	0	0	0	9334	1764	2679
gas	ilu	20	.1	r	no	2800	207	0	272	3331	0	242	0	9430	1812	2694
gas	ilu	20	.5	r	no	2800	207	0	272	3305	0	248	0	9367	1812	2673
gas	ilu	20	.9	r	no	2800	213	3	313	7647	40	292	40	17299	1812	5407
gas	ilu	20	1.	r	no	2800	224	9	381	16624	109	360	109	29737	1812	9674
gas	ilu	50	0	r	yes	2800	207	0	272	3312	0	0	0	9334	1764	2679
gas	ilu	20	0	r	yes	2800	207	0	272	3312	0	0	0	9334	1764	2679
gas	ilu	20	1.	r	yes	2800	224	9	381	16624	109	360	109	29737	1812	9674
gas	ilu	50	0	l	no	2800	207	0	272	3107	0	0	0	9052	1764	2733
gas	ilu	20	0	l	no	2800	207	0	272	3107	0	0	0	9052	1764	2733
gas	ilu	20	1.	l	no	2800	212	3	317	10120	63	296	63	23068	2582	7243
gas	ilu	50	0	l	yes	2800	207	0	272	3107	0	0	0	9052	1764	2733
gas	ilu	20	0	l	yes	2800	207	0	272	3107	0	0	0	9052	1764	2733
gas	ilu	20	1.	l	yes	2800	212	3	317	10120	63	296	63	22298	1812	7243

Table 4. GAS dataset.

The following observations can be drawn from the GAS dataset:

- Block Jacobi and M/ILU preconditionings all performed adequately for this problem. ILU variants with $\omega > 0$ performed worse, sometimes resulting in negative pivots or failure to converge. Block Jacobi took significantly more work but always converged.
- The value of s didn't matter, presumably because few iterations were required and thus no restarts were performed.
- Schur complement and full system iteration apparently performed equivalently.
- Left preconditioning was slightly better than right preconditioning.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
gcneb	bjac	20	0	r	no	45	152	49	817	79558	785	0	0	44191	228	3147
gcneb	ilu	50	0	r	no	360	96	5	305	4378	0	258	0	10856	1764	3158
gcneb	ilu	20	0	r	no	360	96	5	305	4378	0	258	0	10857	1764	3159
gcneb	ilu	20	.1	r	no	360	96	5	305	5920	1	268	1	14148	1812	4285
gcneb	ilu	20	.5	r	no	360	138	19	542	38509	288	514	288	47275	1812	15760
gcneb	ilu	20	.9	r	no	360	265	75	1410	118045	1043	1386	1043	55455	1812	18588
gcneb	ilu	20	1.	r	no	360	290	87	1575	134443	1201	1551	1201	56512	1812	18953
gcneb	ilu	50	0	r	yes	360	96	5	305	4345	0	258	0	10792	1764	3134
gcneb	ilu	20	0	r	yes	360	96	5	305	4345	0	258	0	10793	1764	3135
gcneb	ilu	20	1.	r	yes	360	294	88	1593	135933	1222	1569	1222	56492	1812	18946
gcneb	ilu	50	0	l	no	360	96	5	305	4276	0	258	0	10832	1764	3304
gcneb	ilu	20	0	l	no	360	96	5	305	4276	0	258	0	10832	1764	3304
gcneb	ilu	20	1.	l	no	360	156	28	735	63255	591	711	591	57013	1812	19153
gcneb	ilu	50	0	l	yes	360	96	5	305	4243	0	258	0	10767	1764	3281
gcneb	ilu	20	0	l	yes	360	96	5	305	4243	0	258	0	10767	1764	3281
gcneb	ilu	20	1.	l	yes	360	146	24	718	62173	584	694	584	57359	1812	19270

Table 5. GCONEB dataset.

The following observations can be drawn from the GCONEB dataset:

- Only variants of ILU preconditioning were able to solve the linear systems for the whole simulation. $\omega = 0$ performed best, whereas $\omega = 1$ performed much worse.
- The value of s didn't matter, presumably because few iterations were required and thus no restarts were performed.
- Use of the Schur complement technique was slightly better than solving the full system.
- Left preconditioning was slightly better than right preconditioning.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
s	bjac	20	0	r	no	1000	33	1	70	2381	0	0	0	15385	228	1090
s	ilu	50	0	r	no	1000	33	1	70	707	0	0	0	8095	1764	2222
s	ilu	20	0	r	no	1000	33	1	70	707	0	0	0	8095	1764	2222
s	ilu	20	.1	r	no	1000	33	1	70	708	0	0	0	8153	1812	2225
s	ilu	20	.5	r	no	1000	33	1	70	709	0	0	0	8162	1812	2228
s	ilu	20	.9	r	no	1000	33	1	70	709	0	0	0	8162	1812	2228
s	ilu	20	1.	r	no	1000	33	1	70	711	0	0	0	8181	1812	2235
s	ilu	50	0	r	yes	1000	33	1	70	659	0	0	0	7686	1764	2071
s	ilu	20	0	r	yes	1000	33	1	70	659	0	0	0	7686	1764	2071
s	ilu	20	1.	r	yes	1000	33	1	70	664	0	0	0	7781	1812	2087
s	ilu	50	0	l	no	1000	33	1	70	715	0	0	0	8382	1764	2467
s	ilu	20	0	l	no	1000	33	1	70	715	0	0	0	8382	1764	2467
s	ilu	20	1.	l	no	1000	33	1	70	723	0	0	0	8673	1977	2492
s	ilu	50	0	l	yes	1000	33	1	70	672	0	0	0	8019	1764	2332
s	ilu	20	0	l	yes	1000	33	1	70	672	0	0	0	8019	1764	2332
s	ilu	20	1.	l	yes	1000	33	1	70	678	0	0	0	8126	1812	2351

Table 6. S dataset.

The following observations can be drawn from the S dataset data in Table 6:

- Both Block Jacobi and ILU variants were able to solve the problem, though ILU variants were much more efficient. $\omega = 0$ was best.
- Right preconditioning was slightly better than left preconditioning.
- Schur complement iteration was slightly better than full system iteration.
- The value of s didn't matter, presumably because few iterations were required and thus no restarts were performed.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
antonbo	ilu	20	0	l	no	6174	270	44	897	23109	0	501	0	18436	1764	5888
antonbo	ilu	20	0	l	yes	6174	270	44	897	18545	0	501	0	15240	1764	4768
antonbo	ilu	20	0	r	no	6174	270	44	897	21143	2	501	2	16841	1764	5209
antonbo	ilu	20	0	r	yes	6174	270	44	897	17992	0	501	0	14640	1764	4443

Table 7. ANTONBO dataset.

prob	prec	s	ω	prec side	Schur	sim days	# t.s.	t.s. repeat	# linsol	linsol its	unconv linsol	neg piv	neg piv unconv	linsol f/p/s	prec set f/p/s	prec appl f/p/s
arshad	ilu	20	0	l	no	13140	516	20	1211	20450	0	522	0	12829	1764	3935
arshad	ilu	20	0	l	yes	13140	516	20	1211	19854	0	522	0	12519	1764	3827
arshad	ilu	20	0	r	no	13140	516	20	1211	20618	0	522	0	12701	1764	3761
arshad	ilu	20	0	r	yes	13140	516	20	1211	20028	0	522	0	12399	1764	3658

Table 8. ARSHAD dataset.

For the ANTONBO and ARSHAD datasets, Schur complement iteration was slightly better than full system iteration, and right preconditioning was slightly better than left preconditioning. ILU with $\omega = 0$ was an effective solution technique.

6 Conclusions

This paper has presented some matrix analyses and linear solver experiments using linear systems from a variety of fully implicit reservoir simulations.

The matrices examined exhibited most of their difficulties in the main diagonal blocks, particularly in the pressure equations. Though block Jacobi preconditioning is not adequate to solve very difficult problems, it does improve the matrices significantly.

It was found that hard problems require a comparatively strong preconditioner such as ILU(0). MILU preconditioning was always worse than ILU preconditioning. Using a higher value of the GMRES restart frequency s such as 50 was better than using a lower value such as 20 for the more difficult problems. In some experiments not reported here, some of the difficult problems required a tighter convergence tolerance of $\zeta = 10^{-6}$ rather than 10^{-4} in order to run the simulation in a reasonable time period. There was no clear winner between left and right preconditioning or Schur complement and full system iteration, though there was a general trend that right preconditioning was better and Schur complement iteration was better.

Acknowledgements

The Falcon development team is composed of Michael DeLong, Allyson Gajraj, Wayne Joubert, Kenneth Koch, Olaf Lubeck, Jim Sanderson (Los Alamos National Laboratory), Gautam Shiralkar, Bob Stephenson (Amoco), and Bart van Bloemen Waanders (SGI/Cray).

This work is supported in part by the Department of Energy through grant W-7405-ENG-36, with Los Alamos National Laboratory. This research was performed in part using the resources located at the Advanced Computing Laboratory of Los Alamos National Laboratory, Los Alamos, NM 87545.

References

1. Amoco, "Computer User Instructions for the Mass Variable Formulation of GCOMP," Amoco proprietary document, July 1992.
2. Amoco, "Computer User Instructions for the Implicit Generalized Compositional Model IGCOMP," Amoco proprietary document, July 1992.
3. Wayne Joubert, Deepankar Biswas, "Evaluation of Linear Solvers for Oil Reservoir Simulation Problems, Part 1: The IMPES Case," in preparation.
4. L. C. Young and R. E. Stephenson, "A Generalized Compositional Approach to Reservoir Simulation," *SPE Journal*, October 1983.
5. Gautam S. Shiralkar, R. E. Stephenson, Wayne Joubert, Olaf Lubeck and Bart van Bloemen Waanders, "Falcon: A Production Quality Distributed Memory Reservoir Simulator," SPE Paper 37975, presented at the 1997 SPE Reservoir Simulation Symposium, Dallas, TX, June 8-11, 1997.

Appendix: Figures

Figure 1: Basic run characteristics, dataset S.

Figure 2: Basic run characteristics, dataset S, magnified view.

Figure 3: Extremal entries of unscaled matrix, grid/well submatrices.

Figure 4: Extremal entries of block scaled matrix, grid/well submatrices.

unscaled min A_d

Dataset S

Figure 5: Minimal diagonal block entries.

unscaled min A_d (2)

Dataset S

Figure 6: Minimal diagonal block entries, magnified view.

Figure 8: Maximal diagonal block entries, magnified view.

Dataset S

unscaled max A_d (2)

Figure 7: Maximal diagonal block entries.

Dataset S

unscaled max A_d

Figure 9: Minimal off-diagonal block entries, unscaled matrix.

Figure 10: Maximal off-diagonal block entries, unscaled matrix.

Figure 11: Minimal off-diagonal block entries, block scaled matrix.

Figure 12: Maximal off-diagonal block entries, block scaled matrix.

Figure 13: Extremal grid-to-well connections, unscaled matrix.

Figure 14: Extremal grid-to-well connections, block scaled matrix.

Figure 15: Extremal grid-to-well connections, block scaled matrix, magnified view.

Figure 16: Diagonal dominance, unscaled matrix.

Figure 17: Diagonal dominance, block scaled matrix.

Figure 18: Diagonal dominance, block scaled matrix, magnified view.

Figure 19: Extremal magnitudes of skew-symmetric part, block diagonal.

Figure 20: Extremal magnitudes of skew-symmetric part, off-diagonal, unscaled.

Figure 21: Extremal magnitudes of skew-symmetric part, off-diagonal, block scaled.

Figure 22: Extremal magnitudes of skew-symmetric part, grid-well, unscaled.

Figure 23: Extremal magnitudes of skew-symmetric part, grid-well, block scaled.