Parallel Preconditioning for Sedimentary Basin Simulations

R. Masson[1], P. Quandalle[1], S. Requena[1], and R. Scheichl[2]

[1] Institut Français du Pétrole, Rueil-Malmaison, France
(philippe.quandalleifp.fr)
[2] Deptartment of Mathematical Sciences, University of Bath, UK
(r.scheichlbath.ac.uk)

Abstract. The simulation of sedimentary basins aims at reconstructing its historical evolution in order to provide quantitative predictions about phenomena leading to hydrocarbon accumulations. The kernel of this simulation is the numerical solution of a complex system of non-linear partial differential equations (PDE) of mixed parabolic-hyperbolic type in 3D. A discretisation and linearisation of this system leads to very large, ill-conditioned, non-symmetric systems of linear equations with three unknowns per mesh cell, i.e. pressure, geostatic load, and oil saturation. This article describes the parallel version of a preconditioner for these systems, presented in its sequential form in [7]. It consists of three steps: in the first step a local decoupling of the pressure and saturation unknowns aims at concentrating in the "pressure block" the elliptic part of the system which is then, in the second step, preconditioned by AMG. The third step finally consists in recoupling the equations. Each step is efficiently parallelised using a partitioning of the domain into vertical layers along the y-axis and a distributed memory model within the `PETSc` library (Argonne National Laboratory, IL). The main new ingredient in the parallel version is a parallel AMG preconditioner for the pressure block, for which we use the `BoomerAMG` implementation in the `hypre` library [4].
Numerical results on real case studies, exhibit (i) a significant reduction of CPU times, up to a factor 5 with respect to a block Jacobi preconditioner with an ILU(0) factorisation of each block, (ii) robustness with respect to heterogeneities, anisotropies and high migration ratios, and (iii) a speedup of up to 4 on 8 processors.

1 Introduction

In recent years the 3D modelling and simulation of sedimentary basins has become an integral part in the exploration of present and future reservoirs for almost all major oil companies. It aims at reconstructing the historical evolution of a sedimentary basin in order to provide quantitative predictions about phenomena leading to hydrocarbon accumulations.

A simplified kinematic model of the geometry of the basin is first computed from the geology of the basin using a back-stripping algorithm. The full model comprising hydrocarbon generation, heat transfer, compaction of the porous media, and multi-phase Darcy flow is then calculated forward in time adding layer after layer using the geometric history calculated during the back-stripping process.

The most computing intensive part of the simulation consists in solving numerically a complex system of non-linear partial differential equations (PDEs) modelling the compaction of the porous media and multi-phase Darcy flow in 3D. These equations are discretized using a cell-centred finite volume method in space and an implicit Euler scheme in time, and linearised using Newton's method. In the case of two-phase flow (water – oil), we will therefore have to solve at each time step and in each Newton iteration, a system of coupled linear equations with three unknowns per element (pressure, geostatic load, oil saturation), very large, non-symmetric, and ill-conditioned, in particular due to the strong heterogeneities and anisotropies of the porous media.

The idea for our preconditioning strategy is based on the methods developed in [2, 3, 5, 9] for the related problem of oil reservoir simulation and is described in detail in [7]. It consists of three steps. First of all the equations for pressure and saturation are locally decoupled by taking linear combinations of the two equations on each element. This decoupling aims not only at reducing the coupling, but also at concentrating in the pressure block the elliptic part of the system which requires a good global preconditioning. The second step consists then in preconditioning the pressure block by an algebraic multi-grid method. The recoupling of the pressure with the other unknowns is then in a final step achieved by applying a block Gauss-Seidel strategy or the combinative technique defined in [2] for the reservoir case.

Each step of the preconditioner is efficiently parallelised using a partitioning of the domain into vertical layers along the y-axis and a distributed memory model within the `PETSc` library (Argonne National Laboratory, IL). Only slight modifications are necessary for most of the components of the sequential preconditioner in [7]. The main new ingredient which is necessary however, is a parallel preconditioner for the pressure block for which we use `BoomerAMG` [4] from the `hypre` package (Lawrence Livermore National Laboratory, CA).

We present results with the sequential preconditioner for a new test case with very high migration ratios which was not available for [7]. For this test case, the Quasi-IMPES decoupling together with a recoupling using the combinative technique outperforms all other combinations suggested in [7] significantly. This suggests that in the case of high migration ratios a local decoupling of the two mass conservation equations does not reduce the global coupling of these two equations significantly. Moreover, Gauss decoupling and Householder decoupling destroy the ellipticity of the pressure block and thus AMG does not work very well anymore. All the performance tests (sequential and parallel) were therefore carried out with Quasi-IMPES decoupling.

In a series of numerical tests, the performance of the preconditioner is compared to block Jacobi with an incomplete LU factorisation, ILU(0), of each block, previously the only way to precondition these linear systems efficiently in parallel. We observe a very significant reduction of the CPU-time for the linear solver on one processor, up to a factor 5 with respect to ILU(0). The performance of the preconditioner shows no degradation with respect to the number of elements or to strong heterogeneities and anisotropies in the porous media, and depends only very mildly on the migration ratios of oil. Furthermore, the parallel solver exhibits good speedup on an SGI Origin2000. In all the test cases we obtain speedups between 3 and 4.

The paper is organised as follows. In Section 2 we will give a detailed description of the preconditioning strategy, followed by a series of numerical tests in Section 3.

2 Preconditioning strategy

The mathematical model for the forward problem is formulated as a system of 4 coupled non-linear PDEs and accounts for the conservation equations of two phase Darcy flow (water and oil), as well as the conservation equation of vertical momentum describing the compaction of the porous media together with a rheology law relating the porosity to the geostatic load and to the average fluid pressure. We refer to [7] and [8] and to the references therein for a detailed description of the model.

The equations are discretized using a finite volume scheme in space and a fully implicit time integration. After a Newton linearisation and an elimination of the rheology law we end up at each time step and in each Newton iteration with a system of linear equations of the form

$$A\,\boldsymbol{\delta y} \;:=\; \begin{pmatrix} A^{w,p} & A^{w,\sigma} & A^{w,s} \\ A^{\sigma,p} & A^{\sigma,\sigma} & 0 \\ A^{o,p} & A^{o,\sigma} & A^{o,s} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta p} \\ \boldsymbol{\delta \sigma} \\ \boldsymbol{\delta s} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^w \\ \mathbf{f}^\sigma \\ \mathbf{f}^o \end{pmatrix} \;=:\; \mathbf{f}, \tag{1}$$

where each one of the blocks $A^{\alpha,\beta}$ in A is an $N \times N$ matrix with N being the number of mesh cells. The block rows correspond to the discretisations of the water conservation equation, the vertical compaction equation and the oil conservation equation, respectively. The block columns correspond to the derivatives with respect to the primary variables p, σ, s (i.e. average pore pressure, geostatic load and oil saturation), respectively. The details of the discretisation and the linearisation can be found in [7].

System (1) is very large, non-symmetric and ill-conditioned, in particular due to the strong heterogeneities and anisotropies of the porous media. For an efficient and robust iterative solution of this system, it is therefore crucial to find good preconditioners.

The idea for our preconditioning strategy stems from the related field of oil reservoir simulation, and was first presented for sedimentary basin simulations in [7]. In reservoir simulations, the geological time spans that are simulated are much shorter in comparison to the time spans considered here (i.e. years instead of million years), so that the porosity and the effective stress σ can be kept constant, thus leading to systems similar to (1), but in pressure and saturation only. In [2, 3, 5] a preconditioning strategy has been developed for these problems decoupling the treatment of pressure and saturation. In [7] we have used a similar strategy for (1) which consists of three steps: (i) local decoupling of the two mass conservation equations on each cell Ω_k of the mesh (or in other words, choosing an appropriate equation for the average pore pressure in Ω_k), (ii) preconditioning of the pressure unknowns, (iii) preconditioning of the remaining unknowns (Block Gauss-Seidel), or if this is not sufficient, feedback-coupling with the pressure equation (two-stage preconditioners). We will describe these three steps briefly and explain in particular how they are parallelised. In order to do this we need to first describe the parallelisation strategy.

2.1 Parallelisation Strategy

At the moment, the parallelisation uses a partitioning of the domain into vertical layers along the y-axis and a distributed memory model. However, this one-dimensional topology for the parallel model is inherited from the parallelisation of the entire sedimentary basin simulation code and the preconditioner itself is not restricted to a partitioning into vertical layers. Communications between the processors and synchronisation are handled using `MPI` (Message Passing Interface) within the `PETSc` library (Argonne National Laboratory, IL) on the basis of a ghost point strategy (see [1] for details).

The parallelisation of each individual component of the preconditioner presented in [7] is discussed in the following.

2.2 First Step: Local Decoupling – Choice of the Pressure Equation

The decoupling step corresponds to multiplying system (1) by

$$\mathcal{P}_{\mathcal{L}}^{-1} := \begin{pmatrix} \mathrm{Diag}(\gamma_k^{p,w}) & 0 & \mathrm{Diag}(\gamma_k^{p,o}) \\ 0 & I & 0 \\ \mathrm{Diag}(\gamma_k^{s,w}) & 0 & \mathrm{Diag}(\gamma_k^{s,o}) \end{pmatrix} \tag{2}$$

and leads to the transformed system

$$\mathcal{A}\,\boldsymbol{\delta y} := \begin{pmatrix} \mathcal{A}^p & \mathcal{A}^{p,\sigma} & \mathcal{A}^{p,s} \\ \mathcal{A}^{\sigma,p} & \mathcal{A}^{\sigma} & 0 \\ \mathcal{A}^{s,p} & \mathcal{A}^{s,\sigma} & \mathcal{A}^{s} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta p} \\ \boldsymbol{\delta \sigma} \\ \boldsymbol{\delta s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^p \\ \boldsymbol{f}^\sigma \\ \boldsymbol{f}^s \end{pmatrix} =: \boldsymbol{f} \tag{3}$$

with $\mathcal{A} := \mathcal{P}_{\mathcal{L}}^{-1} A$ and $\boldsymbol{f} := \mathcal{P}_{\mathcal{L}}^{-1}\,\mathbf{f}$. This corresponds to a left preconditioning of (1) by $\mathcal{P}_{\mathcal{L}}^{-1}$. The matrix $\mathcal{P}_{\mathcal{L}}^{-1}$ is chosen such that

(a) $\gamma_k^{p,w}\,[A^{w,s}]_{kk} + \gamma_k^{p,o}\,[A^{o,s}]_{kk} = 0$,
(b) the block $\mathcal{A}^p$ concentrates the elliptic part of the system,
(c) the norm of $\mathcal{P}_{\mathcal{L}}^{-1}$ is $O(1)$.

In [7] we proposed three different choices for $\mathcal{P}_{\mathcal{L}}^{-1}$ based on Gaussian elimination, on Householder reflection, and on a quasi-explicit treatment of the saturation (Quasi-IMPES), respectively. Unfortunately none of these choices satisfies all the three criteria above. The first choice $\mathcal{P}_{GE}^{-1}$ satisfies (a), but does not necessarily satisfy (b) and (c) very well; the second choice $\mathcal{P}_{HH}^{-1}$ satisfies (a) and (c), but again does not satisfy (b) very well; the third choice $\mathcal{P}_{QI}^{-1}$, on the other hand does not satisfy (a), but satisfies (b) and (c). We will analyse the importance of these three criteria more carefully in Section 3.1.

No additional work is necessary to parallelise this step, since all the operations are local to each mesh cell and thus local to each processor.

2.3 Second Step: Pressure Block Preconditioner

From the decoupling step, the block $\mathcal{A}^p$ should concentrate the ellipticity, and thus the ill-conditioning of the system and basically correspond to a discrete diffusion operator with a strongly heterogeneous and anisotropic diffusivity tensor. We propose to use algebraic multi-grid (AMG) which is known to be efficient and robust for such stiff elliptic linear systems, even in the presence of strong heterogeneities and anisotropies. More precisely, we shall use `BoomerAMG` by Henson and Yang [4], a very efficient and robust parallel implementation of AMG within the `hypre` software package (Lawrence Livermore National Lab, CA).

To test whether this implementation is competitive in the sequential case, we will also include some results in Section 3 comparing `BoomerAMG` on one processor with the implementation `AMG1R5` by Ruge and Stüben [6] which was used in [7].

2.4 Third Step: Recoupling

The decoupling step is aimed at reducing the dependency of the pressure equations on the saturation. Thus, provided the decoupling is effective, we can assume that the block $\mathcal{A}^{p,s}$ is small compared to $\mathcal{A}^p$. Furthermore, physical considerations also allow us to make the same assumption about the block $\mathcal{A}^{p,\sigma}$ (see [7] for details). This leads to the idea for the following Block Gauss-Seidel (BGS) preconditioner for (3):

$$\mathcal{P}_{BGS}^{-1} := \begin{pmatrix} \mathcal{P}^p & 0 & 0 \\ \mathcal{A}^{\sigma,p} & \mathcal{A}^{\sigma} & 0 \\ \mathcal{A}^{s,p} & \mathcal{A}^{s,\sigma} & \mathcal{P}^s \end{pmatrix}^{-1} \tag{4}$$

Here, $\mathcal{P}^p$ is the AMG preconditioner for the pressure block $\mathcal{A}^p$ discussed above, and $\mathcal{P}^s$ is a preconditioner for the saturation block $\mathcal{A}^s$. Since the dependency on saturation is hyperbolic or transport dominated parabolic, it is sufficient to use a simple relaxation scheme, like Jacobi or Gauss-Seidel, for $\mathcal{P}^s$. In Section 3 we will use one iteration of a hybrid version of Jacobi and Gauss-Seidel for $\mathcal{P}^s$ which involves no communication between the different processors and reduces to Gauss-Seidel on each processor. Finally, since the compaction equation is one dimensional and can be solved in each column independently, the block $\mathcal{A}^{\sigma}$ can be cheaply inverted by back substitutions. This is also the case in the parallel version, provided each vertical mesh column is assigned to one processor.

The assumption that the blocks $\mathcal{A}^{p,\sigma}$ and $\mathcal{A}^{p,s}$ can be neglected may be too strong. It is necessary in such a case that the preconditioner also provides some feedback from the second and the third block row to the first. A way of providing such a feedback-coupling is to combine the Block Gauss-Seidel preconditioner $\mathcal{P}_{BGS}^{-1}$ in a multiplicative way with a second preconditioner that provides (at least locally) such a coupling and that is cheap to invert. A good candidate for such a preconditioner is zero fill-in incomplete LU factorisation, ILU(0), of $\mathcal{A}$. It is cheap and provides a good local coupling of the different physical unknowns. Again to reduce communication in the parallel version we carry out the incomplete LU factorisation only locally on each processor. In other words, the second preconditioner is chosen to be a block Jacobi preconditioner with an ILU(0) factorisation of each block, denoted $\mathcal{P}_{BILU}^{-1}$. The combined preconditioner can then be written as

$$\mathcal{P}_{C2S}^{-1} \;:=\; \mathcal{P}_{BGS}^{-1} \;+\; \mathcal{P}_{BILU}^{-1}\left(I \;-\; \mathcal{A}\mathcal{P}_{BGS}^{-1}\right)\;. \tag{5}$$

Similar techniques in reservoir simulations, notably in [2] and [5], are usually referred to as combinative two-stage preconditioners (C2S) and we will adopt the same name.

The recoupling preconditioners $\mathcal{P}_{BGS}^{-1}$ and $\mathcal{P}_{C2S}^{-1}$ are used as right preconditioners for (3). And so in summary (putting all three steps together) we obtain the following preconditioned system:

$$\tilde{A}\,\boldsymbol{\delta}\tilde{\boldsymbol{y}} \;:=\; (\mathcal{P}_{\mathcal{L}}^{-1}A\mathcal{P}_{\mathcal{R}}^{-1})\,(\mathcal{P}_{\mathcal{R}}\boldsymbol{\delta}\boldsymbol{y}) \;=\; (\mathcal{P}_{\mathcal{L}}^{-1}\mathbf{f}) \;=:\; \tilde{\mathbf{f}} \tag{6}$$

where $\mathcal{P}_{\mathcal{L}}^{-1}$ is either $\mathcal{P}_{GE}^{-1}$, $\mathcal{P}_{HH}^{-1}$ or $\mathcal{P}_{QI}^{-1}$ and $\mathcal{P}_{\mathcal{R}}^{-1}$ is either $\mathcal{P}_{BGS}^{-1}$ or $\mathcal{P}_{C2S}^{-1}$.

3 Numerical Results

In this section we will discuss the results of the numerical tests which we effectuated to evaluate the performance of the preconditioner in terms of efficiency and robustness on some test cases from real case studies. The five test cases are subsections of sedimentary basins taken from real case studies of major oil companies. These studies are confidential and so we will refer to the test cases as Problems **A** – **E**. Problems **A** and **D** are problems with strong heterogeneities and anisotropies in the permeability tensors of the porous media but with low migration ratios, and thus represent almost single-phase flow. Problems **B** and **C** are problems with higher migration ratios, but without any strong heterogeneities in the permeability tensors. Problem **E**, on the other hand, comprises both these difficulties and is thus the hardest test case (not yet available in [7]). We will only regard a set of time steps towards the end of the simulation, when most the z-layers are deposited and the linear systems are the largest. The number of unknowns ranges from 16974 for Problem **A** to 129756 for Problem **E**. Unfortunately, larger problems of entire sedimentary basins were not available for reasons of confidentiality.

As the iterative solution method for the preconditioned systems (6) we use the Bi-CGStab method by Van der Vorst [10] which is particularly suited for non-symmetric linear systems. The stopping criterion is the reduction of the residual by a factor of $\varepsilon = 10^{-6}$. The CPU-times are those obtained on an SGI Origin2000 using between 1 and 8 processors. We will compare the results obtained with our preconditioner with the results using block Jacobi preconditioning with an incomplete LU factorisation ILU(0) of each block, previously the only way to precondition these linear systems efficiently in parallel. As we

Table 1. Comparison of the different decoupling/recoupling strategies – Iterations

Problem	N	$\mathcal{P}_{BILU}$	Block Gauss-Seidel		Combinative Two-Stage	
			$\mathcal{P}_{GE}$	$\mathcal{P}_{QI}$	$\mathcal{P}_{GE}$	$\mathcal{P}_{QI}$
A	16947	47.8	4.2	4.2	3.5	3.5
B	23703	20.1	9.4	10.2	4.6	3.8
C	69312	21.0	13.0	13.6	2.8	6.0
D	94089	103.8	4.6	4.6	3.8	3.8
E	129756	138.1	111.0	88.7	55.4	8.4

can see in Table 1, the performance of ILU(0) deteriorates rather strongly as the mesh size gets larger even in the sequential case, in particular for problems with strong heterogeneities (i.e. **A**, **D**, and **E**). Moreover, the block Jacobi parallelisation of it provides no global preconditioning at all and will eventually be not sufficient anymore for convergence (when the number of processors gets too large). Increasing the fill-in for the incomplete LU decomposition does not cure these problems, and the reduction in the number of iterations is in all cases weighed off by the extra cost of each iteration in terms of efficiency.

3.1 Comparison of the different decoupling/recoupling strategies

To compare the robustness of the different decoupling and recoupling strategies which have been proposed in Section 2, we will look at the average number of Bi-CGStab iterations per Newton iteration in the sequential case. These are given in Table 1.

Based on the results for Problems **A** – **D**, we concluded in [7] that the choice of decoupling has no strong influence on the robustness of the preconditioner, but that for problems with high migration ratios the block Gauss-Seidel recoupling was not sufficient to guarantee the robustness of the preconditioner anymore. The combinative two-stage preconditioner, on the other hand, proved to be very robust in all cases.

The new test case **E** however, which combines both strong heterogeneities and high migration ratios, gives us some very valuable new insight into the robustness of our preconditioner. In particular, we can observe that Gauss decoupling, $\mathcal{P}_{GE}$ (as well as Householder decoupling $\mathcal{P}_{HH}$), lead to a much worse performance than Quasi-IMPES decoupling $\mathcal{P}_{QI}$, i.e. 55.4 iterations for $\mathcal{P}_{GE}$ in comparison to 8.4 iterations for $\mathcal{P}_{QI}$ using two-stage recoupling. This is probably due to the loss of ellipticity in the pressure block. The results also underline (even more dramatically) that block Gauss-Seidel recoupling is not sufficient for problems with high migration ratios, i.e. 88.7 iterations for block Gauss-Seidel in comparison to 8.4 iterations for the two-stage recoupling (using $\mathcal{P}_{QI}$). Thus, we can conclude that in the case of high migration ratios a local decoupling of the two mass conservation equations does not reduce the global coupling of the two equations significantly. However, the combinative two-stage preconditioner provides the necessary recoupling and is very robust in combination with Quasi-IMPES decoupling $\mathcal{P}_{QI}$ (see the final column in Table 1).

3.2 Comparison of the pressure preconditioners

In order to allow for a parallelisation of AMG the `BoomerAMG` implementation by Henson and Yang [4] differs from the original AMG code `AMG1R5` [6] in its coarsening strategy. (Here we use Falgout coarsening within `BoomerAMG`.) To test whether `BoomerAMG` is competitive as the pressure preconditioner $\mathcal{P}^p$ in the sequential case, we compare in Table 2 the results for Problems **D** and **E** using one V-cycle of `BoomerAMG` on one processor with the results using one V-cycle of `AMG1R5` [6]. We see that apart form slight variations they perform almost identically.

Table 2. Comparison of pressure preconditioners $\mathcal{P}^p$: Problems **D** (left) and **E** (right)

$\mathcal{P}^p$	Iterations		CPU-Time		Iterations		CPU-Time	
	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$
`BoomerAMG`	4.4	3.4	6.1 s	6.6 s	76.3	7.8	81.4 s	16.2 s
`AMG1R5`	4.6	3.8	5.0 s	6.0 s	88.7	8.4	86.9 s	15.7 s
exact – LU	3.0	1.8			58.4	6.8		

We also include (in Table 2) the number of iterations for our preconditioner when the pressure block is inverted exactly using an LU factorisation of $\mathcal{A}^p$. The number of iterations decreases at most by a factor of 2 which is astonishing.

3.3 Comparison with Block Jacobi – Parallel Speedup

Finally and most importantly we look at the parallel robustness and efficiency of our preconditioner and compare it to block Jacobi preconditioning with ILU(0) per block, $\mathcal{P}_{BILU}$. The results (with Quasi-IMPES decoupling) for Problems **D** and **E** are given in Tables 3 and 4, respectively.

First of all, observe the extremely good parallel robustness of our preconditioner for all of the recoupling strategies. The number of iterations hardly grows

Table 3. Comparison with Block Jacobi: Problem **D** ($N = 94089$)

Processors	Iterations			CPU-Time		
	$\mathcal{P}_{BILU}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$	$\mathcal{P}_{BILU}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$
1	103.8	4.4	3.4	29.4 s	6.1 s	6.6 s
2	103.2	4.2	3.2	16.7 s	4.5 s	4.7 s
4	109.2	4.4	3.4	6.9 s	2.8 s	2.9 s
8	116.6	4.6	3.4	3.0 s	2.0 s	2.3 s

Table 4. Comparison with Block Jacobi: Problem **E** ($N = 129756$)

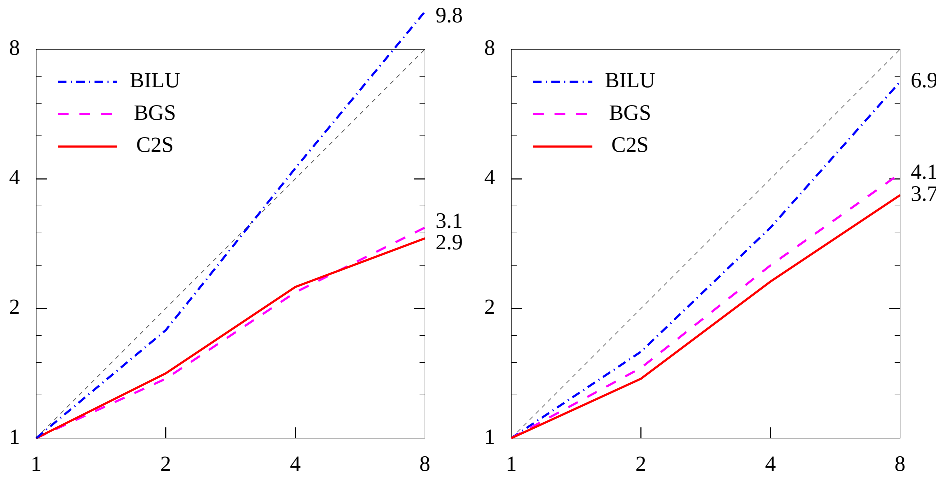
Processors	Iterations			CPU-Time		
	$\mathcal{P}_{BILU}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$	$\mathcal{P}_{BILU}$	$\mathcal{P}_{BGS}$	$\mathcal{P}_{C2S}$
1	138.1	76.3	7.8	56.3 s	81.4 s	16.2 s
2	147.3	78.3	8.2	35.6 s	56.5 s	11.9 s
4	167.8	76.0	8.0	18.0 s	32.3 s	7.0 s
8	191.5	78.6	8.0	8.2 s	19.8 s	4.4 s

at all with the number of processors. In terms of CPU-time, our preconditioner is more than 3 times more efficient than $\mathcal{P}_{BILU}$ on one processor and the parallel speedup is very good: between 3 and 4 on 8 processors (see Figure 1). This is in fact about the same speedup than the one reported in [4] for `BoomerAMG` on more simple test problems. Finally, it is interesting to note that the modification of the combinative two-stage recoupling seems to lead to a better efficiency of the preconditioner both in terms of iterations and CPU-time and that this method therefore is our method of choice.

References

1. S. Balay, W. Gropp, L.C. McInnes, and B. Smith, PETSc Users Manual, Technical Report ANL-95/11 – Revision 2.1.0, Argonne, IL (2001).
2. A. Behie and P.K.W. Vinsome, Block iterative methods for fully implicit reservoir simulation, *Soc. Petroleum Eng. J* **22** (1982), 658–668.
3. H.C. Edwards, A parallel multilevel-preconditioned GMRES solver for multiphase flow models in the Implicit Parallel Accurate Reservoir Simulator (IPARS), TICAM Report 98-04, The University of Texas at Austin (1998).
4. V.E. Henson and U.M. Yang, BoomerAMG: A parallel algebraic multigrid solver and preconditioner, *Applied Numerical Mathematics* **41** (2002), 155–177.
5. S. Lacroix, Y.V. Vassilevski, and M.F. Wheeler, Decoupling preconditioners in the Implicit Parallel Accurate Reservoir Simulator (IPARS), *Numer. Lin. Alg. Appl.* **8** (2001), 537–549.
6. J.W. Ruge and K. Stüben, Algebraic Multigrid (AMG), In *Multigrid Methods* (S.F. McCormick, ed.), Frontiers in Applied Mathematics, vol. **5**, SIAM, Philadelphia, 1986.

Fig. 1. Parallel Speedup: Problems **D** (left) and **E** (right)



7. R. Scheichl, R. Masson, and J. Wendebourg, Decoupling and block preconditioning for sedimentary basin simulations in TEMIS3D, submitted to *Computational Geosciences*, Bath Mathematics Preprint 02/05, University of Bath (2002).
8. F. Schneider, S. Wolf, I. Faille, and D. Pot, A 3D basin model for hydrocarbon potential evaluation: Application to Congo Offshore, *Oil and Gas Science and Technology - Rev. IFP* **55**, 1 (2000), 3–13.
9. Y.V. Vassilevski, Iterative Solvers for the Implicit Parallel Accurate Reservoir Simulator (IPARS), II: Parallelization Issues, TICAM Report 00-33, The University of Texas at Austin (2000).
10. H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Comput.* **12** (1992), 631–644.