Comparing some Methods and Preconditioners for Hydropower Plant Flow Simulations

Luiz M. Carvalho[1], Wagner Fortes[1], and Luc Giraud[2]

[1] Group of Environmental Studies of Hydropower Reservoirs (GESAR), Rio de Janeiro State University, R. Fonseca Teles, 121, 20550-013, Rio de Janeiro, RJ, Brazil
[luizmc,wfortes]@gmail.com
[2] ENSEEIHT-IRIT ,Toulouse, France
giraud@n7.fr

Abstract. We describe some features of a three-dimensional numerical simulator currently under development for studying water physico-chemical properties during the flooding of hydroelectric plants reservoirs. The work is sponsored by the Brazilian Electric Energy National Agency (ANEEL) and conducted with Furnas Centrais Elétricas S. A., the leading Brazilian power utility company. An overview of the simulator requirements is given. The mathematical model, the software modules, and engineering solutions are briefly discussed, including the finite element based transport module. We compare methods, iterative methods and preconditioners used to solve the sparse linear systems which arise from the discretization of three-dimensional partial differential equations.

1 Introduction

Is flooding of soils, consecutive to the creation of water reservoirs, a significant anthropic source of greenhouse gases (GHG) emissions? In a mid and long term perspective, can hydroelectrical energy be considered a clean energy? The answers of the scientific and industrial communities to these questions are not conclusive [1], [2]. In order to participate in this discussion, a group of researchers have been developing a numerical simulator for studying water physico-chemical properties during the flooding of hydroelectric plants reservoirs [3,4]. In the near future, this simulator will be able to analyze the production, stocking, consumption, transport, and emission of carbon dioxide (CO_2) and methane (CH_4) in reservoirs. The simulator comprises a Graphical User Interface (GUI) using OpenGL, and a Shell Interpreter. Geographical data in various formats are fed to the *Terrain* module, that generates the level sets and prepares the site geometry for the next module, *Phyto*. Drainage and phyto-physionomy data are added by *Phyto* and handed over to the *Mesh* module which generates the grid for the transport simulator. The prototype was developed with *Matlab* and is currently being rewritten in *C++*. The *Transport* module comprises the core of the simulator and uses a mixed finite element scheme.

The simulator is based on a nonlinear system of partial differential equations, the Navier-Stokes equation and scalar transport [5,6], presented bellow in a nondimensional form:

$$
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} &= -\nabla p + \frac{1}{\text{Re}}\nabla\cdot\nu_t(\nabla\mathbf{u}+\nabla\mathbf{u}^T) \\
\nabla\cdot\mathbf{u} &= 0 \\
\frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c &= \frac{1}{\text{Re Sc}}\nabla\cdot D_t\nabla c + S,
\end{aligned}
\tag{1}
$$

where $\mathbf{u}$, p are, respectively, the nondimensional velocity and pressure, *Re* is the Reynolds number, ν_t is the nondimensional effective viscosity, c is the advected scalar, Sc is the Schmidt number, D_t is the nondimensional effective diffusely and S is a font term.

In the current version, the simulator implements a cubic tetrahedron element, mini element [7,8], with the velocity evaluated at the vertices and the centroid of the element; pressure is evaluated only in the vertices, see Figure 1.

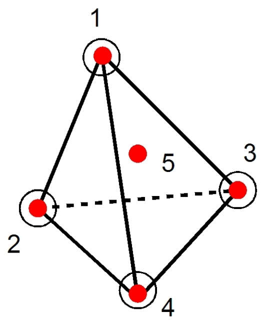


Fig. 1: Mini element with velocity in five points (centroid included) and pressure in four.

For the treatment of this equations a common approach is the Galerkin method [9] which transforms the equations (1) in an system of ordinary differential equations:

$$
\begin{aligned}
M\dot{\mathbf{u}} + \frac{1}{Re}K\mathbf{u} - Gp &= \hat{b}_1 \\
D\mathbf{u} &= b_2 \\
M_c\dot{c} + \frac{1}{Re\,Sc}K_c c &= b_3,
\end{aligned}
\tag{2}
$$

in equations (1) and (2) we are using the same symbols $\mathbf{u}$, p, c for continuous and discrete variables.

A semi-Lagrangian method [10] is used for the time discretization. This approach changes the system (2) into:

$$
\begin{aligned}
M(\frac{\mathbf{u}_i^{n+1}-\mathbf{u}_d^n}{\Delta t})+\frac{1}{Re}K\mathbf{u}^{n+1}-Gp^{n+1}&=\hat{b}_1\\
D\mathbf{u}^{n+1}&=b_2\\
M_c(\frac{c_i^{n+1}-c_d^n}{\Delta t})+\frac{1}{Re\,Sc}K_c c^{n+1}&=b_3.
\end{aligned}
\tag{3}
$$

The variables from the previous time step are used to evaluate the chemical species field, uncoupling the hydrodynamics from the transport of the scalars variable. As the simulator models three-dimensional space, the linear systems arising from these schemes are huge and is mandatory to implement iterative methods for their solutions.

2 Coupled and Segregated Methods

The first two equations of (3) yield the following linear system:

$$
\begin{pmatrix} A & G \\ D & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix}=\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \qquad \text{or} \qquad \mathcal{A}x=b, \tag{4}
$$

where $A=(\frac{1}{\Delta t}M+\frac{1}{Re}K)\in\mathbb{R}^{n\times n}$ is symmetric and positive definite, G, $D^T\in\mathbb{R}^{n\times m}$ and $n\geq m$, both have complete rank. The unknown u represents the velocity and p the pressure in each point. Vectors $b_1=(\hat{b}_1+\frac{1}{\Delta t}M-\mathbf{u}_d^n)$ and b_2 compose the constant right-hand side. The matrix $\mathcal{A}$ is known as saddle point matrix [11]. The third equation in (3) produces an easier system, that we are not treating is this work.

In order to uncouple the velocity and the pressure components in (4), one may use a segregated approach performing a block LU factorization of the original matrix:

$$
\begin{pmatrix} A & G \\ D & 0 \end{pmatrix}\sim\begin{pmatrix} A & 0 \\ D & -D\tilde{A}^{-1}G \end{pmatrix}\begin{pmatrix} I & \tilde{A}^{-1}G \\ 0 & I \end{pmatrix}, \tag{5}
$$

where $\tilde{A}$ is either equal to A or an approximation of A that is easier to solve. For the former, $S=-D\tilde{A}^{-1}G$ is the exact Schur complement matrix of the zero block of the matrix $\mathcal{A}$, and for the latter is an approximation of the Schur complement matrix. One has to compute:

Algorithm 1 (Projection Method)
Let δu be an auxiliary variable.

```
1. $A\delta u = b_1$;
2. $-D\tilde{A}^{-1}Gp = b_2 - D\delta u$;
3. $u = \delta u - \tilde{A}^{-1}Gp$;
```

in a Fluid Dynamics framework, this alternative is called Projection method [12].

Depending on boundary conditions the complete system can be symmetric or nonsymmetric. When A is symmetric and positive definite (SPD) and $G=D^T$, if $\tilde{A}$ is a diagonal matrix, then $-D\tilde{A}^{-1}G$ is also symmetric. In this case, both systems can be solved using the preconditioned conjugate gradient method (PCG) [13, 14]. When $G \neq D^T$ the approximated Schur complement matrix can be solved using GMRes [15] or BiCGStab [16]. The right-hand side of (5) replaces the original matrix involved in the linear system (4), and the computed solution accuracy depends on the quality of the approximation $\tilde{A}$ of A. Yet another segregated option is to obtain an approximation for the Schur complement matrix $-D\tilde{A}^{-1}G$ instead of an approximation for A. In this case, it is necessary to substitute the third step of Algorithm 1 by

$$Au = b_1 - Gp, . \tag{6}$$

The Schur complement matrix S, in our application, is dense and huge with order equal to the number of vertices of a three-dimensional mesh. However, Figure 2 represents a typical S matrix with tiny entries spread out of a few central diagonals. One may implement implicit and explicit alternatives for treating S. In the implicit case the matrix S is kept as the product $-D\tilde{A}^{-1}G$ and in the explicit alternative an approximation of S is computed. In section 3, we address some explicit alternatives we have tested.

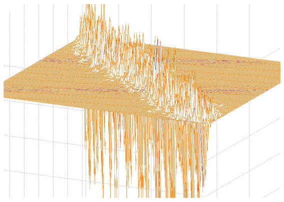


Fig. 2: An example of a Schur complement matrix with tiny entries far from the main diagonal.

The coupled method disregards the saddle point structure and (4) can be solved, for instance, by GMRes or BiCGStab. Nevertheless, this statement can be relaxed with the use of preconditioners that take into account the saddle point structure. The coupled approach spends much more computational resources than the segregated alternative but, in general, with a better numerical behavior, see section 6. Also, in order to obtain convergence it is necessary to use preconditioners, and reorderings.

3 Explicit Schur Complement Matrix

We have implemented three alternatives in order to assemble $S = -D\tilde{A}^{-1}G$: diagonal, probe, and complete approximation matrix.

3.1 Diagonal

$\tilde{A}$ is a diagonal matrix with two possibilities: diagonal and lumped. In the former, $\tilde{A} = \mathrm{diag}(A)$, the diagonal of A, in the latter

$$\tilde{A}_{ii} = \sum_{n}^{j=1} A_{ij},$$

where $\tilde{A}_{ii}$ is the diagonal element of $\tilde{A}$ in position (i,i). In both cases S is still sparse, as the product $\tilde{A}^{-1}G$ does not change the sparsity pattern of G, and the product of D by $\tilde{A}^{-1}G$, as long as the sparsity is considered, only puts in relation vertices that are "neighbors of neighbors", which is still quite sparse in the problem's three-dimensional mesh.

3.2 Probe

In [17], Chan and Mathew presented a quite simple idea for retrieving elements of an unassembled matrix E originated from a structured mesh. A set of probe vectors composed by zeros and ones forms a rectangular matrix W, for instance, when using three probe vectors

$$W = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 1\,0\,0 \\ 0\,1\,0 \\ \vdots\,\vdots\,\vdots \end{pmatrix}.$$

If E is a tridiagonal matrix the product EW retrieves exactly the three diagonals of E. If the diagonals of E far from the main diagonal have an important decreasing, this approach can provide a good approximation for E. Although the simulator mesh is not structured, we have noted, during the experiments, that S has tiny elements far from the main diagonal. So we have tested a probed approximation for S. As an approximation of the Cholesky factors of A are available, for instance, from a previous solution of the $(1,1)$-block using an incomplete Cholesky factorization of A, we have tested the following algorithm for mounting an approximation of S:

Algorithm 2 (Probing S)
Let C and C^T be approximated Cholesky factors of A.

```
1. $F = GW$;
2. $H = C^{-1}(C^{-T}F)$, forward and backward substitutions;
3. $\tilde{S} = DH$;
4. $S = f(\tilde{S})$, where $f$ retrieves the desired entries.
```

This alternative gave poor numerical results, we are not even presenting numerical results. Although we were aware of approaches tailored to saddle point problems with unstructured meshes [18], their implementation is a matter of future tests.

3.3 Complete approximated matrix

Another alternative is to use the following algorithm:

Algorithm 3 (Approximating S)
Let F has the same size as G.

1. *$AF = G$, multiple right-hand side problem;*
2. *$\tilde{S} = -DF$;*
3. *$S = g(\tilde{S})$ where g modifies the sparse pattern of the operated matrix by applying a threshold.*

The first step is the most expensive one. Although it can be done in various ways, for instance as a multiple right-hand side problem, we have implemented it using the approximated Cholesky factors of A. In Matalb, this alternative can be quite time costly as one need to exclude tiny elements in first and third steps out of the internal Matlab loops, but deserves further developments, mainly in C++, as the numerical results are comparable to other alternatives.

4 Segregated Solvers

After assembling the Schur complement matrix or an approximation of it (in the following we call both S), one has to solve the block system

$$\begin{pmatrix} A & 0 \\ D & S \end{pmatrix} \begin{pmatrix} I & \tilde{A}^{-1}G \\ 0 & I \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} b1 \\ b2 \end{pmatrix} \tag{7}$$

using Algorithm 1. As A is SPD, the PCG is the chosen algorithm for the first step of Algorithm 1; for solving the second step, for S, we have implemented GMRes.

As the actual solution for (4) was available, we could measure the error, instead of simply the residual, and we observed that, although cheaper than the coupled alternatives, the segregated solvers gave worse solutions. So we also tested an iterative refinement algorithm [19,20] for the projection method by applying it to the residual of the computed solution. In our experiments three to five iterative steps were enough to improve the solution quality, see section 6.

5 Preconditioners

We describe preconditioners for both segregated and coupled methods. Based on [21], we also address preconditioners for the coupled approach by applying the segregated method to the residual. As the reorderings have an important impact on the global performance, for the sake of completeness, we also describe the used alternatives.

5.1 Preconditioners for Coupled Methods

We have tested seven preconditioners for the solution of $\mathcal{A}x = b$ by a coupled method using iterative Krylov subspace projection methods: GMRes and BiCGStab. For all tested cases, we approximate $\mathcal{A}$ by $\tilde{\mathcal{A}}$.

Diagonal

1. Traditional, where $\tilde{A}$ is the diagonal of A.
2. Lumped, where $\tilde{A}$ is such that $\tilde{A}_{ii} = \sum_{n}^{j=1} A_{ij}$.

ILU(0) $\tilde{A}$ is such that its incomplete LU decomposition preserves the same sparse structure as A.

ILUT(τ) The ILUT(τ) is the Incomplete LU with a threshold τ; this preconditioner is dynamically formed, as the elements smaller than τ are discarded, so one does not know beforehand its sparsity pattern.

Segregated Preconditioner Based on [21], we have used the segregated method, presented in section 4, as a preconditioner for the coupled method. The segregated methods are inexpensive, although, sometimes, have a poor numerical result, they can perform very well in practice as preconditioners .

Block Preconditioners In [22] some preconditioners were proposed, we have implemented two of them: a block diagonal and a block triangular. The applications of these preconditioners generate inner-outer iterations schemes [23–25], and generate two linear system that ought to be solved for the right-hand side residual.

Block Diagonal: This preconditioner uncouples the variables and writes

$$P_1 = \begin{pmatrix} A & 0 \\ 0 & DA^{-1}G \end{pmatrix}. \tag{8}$$

In exact arithmetic, the preconditioned matrix $P_1^{-1}\mathcal{A}$, as $\mathcal{A}$ is nonsingular, has three distinct eigenvalues 1, and $\frac{1\pm\sqrt{5}}{2}$. So the minimum polynomial has degree three, which guarantees the convergence of a Krylov subspace projection method in three iterations.

Algorithm 4 (Block Diagonal)

```
1. $\mathbf{A}\mathbf{z}_1 = \mathbf{r}_1$
2. $\mathbf{D}\mathbf{A}^{-1}\mathbf{G}\mathbf{z}_2 = \mathbf{r}_2$
```

In the first step of algorithm 4, we apply the PCG method with an incomplete Cholesky with threshold (ICCT) preconditioner. In the second step, we choose one of the available approximations to S, and we apply the GMRes method with an ILUT(τ) preconditioner.

Block Triangular: Another preconditioner based on the saddle point structure writes

$$P_2 = \begin{pmatrix} A & G \\ 0 & DA^{-1}G \end{pmatrix}. \tag{9}$$

The preconditioned matrix $P_2^{-1}\mathcal{A}$, as $\mathcal{A}$ is nonsingular, has two distinct eigenvalues: ± 1. In this case, the minimum polynomial has degree two, and a Krylov subspace projection method converges in two iterations, in exact arithmetic.

Algorithm 5 (Block Triangular)

```
1. $\mathbf{D}\mathbf{A}^{-1}\mathbf{G}\mathbf{z}_2 = \mathbf{r}_2$
2. $\mathbf{A}\mathbf{z}_1 = \mathbf{r}_1 - \mathbf{G}\mathbf{z}_2$
```

In this case, we apply the PCG with ICCT preconditioner in the first step. In the second step, after choosing an approximation for S, we apply the GMRes method with ILUT(τ) as the preconditioner.

5.2 Segregated Preconditioners

We need two preconditioners in Algorithm 1: one for the first and third (depending on the construction of S) steps and other for the second step. We observe that the incomplete factorizations of A used as preconditioners in the first and third steps can be used for approximating S. Although the preconditioners are applied independently, we can interpret as we have implemented the block diagonal version of [22], as addressed just above.

5.3 Reorderings

In addition, we consider the previous preconditioners combined with two block reorderings applied to (4). Namely, we investigate the column approximate minimum degree permutation (AMD) [26] and the symmetric reverse Cuthill-McKee (SRCM) [27, 28]. In order to preserve the saddle point structure, we reordered firstly the symmetric matrix A.

$$\hat{A} = P_A A P_A^T$$

where P_A is a column-permutation matrix for A. As the Schur complement matrix, S, was almost symmetric by structure, we also applied a symmetric reordering besides AMD.

$$\hat{S} = P_S S P_S^T$$

where P_S is a column-permutation matrix for S.

Then, we reordered matrices D and G accordingly:

$$\hat{G} = P_S G P_A^T, \text{ and } \hat{D} = P_A D P_S^T$$

are the reordered matrices. In Figure 3, we present the structure of a simulator typical saddle point matrix before and after reordering.

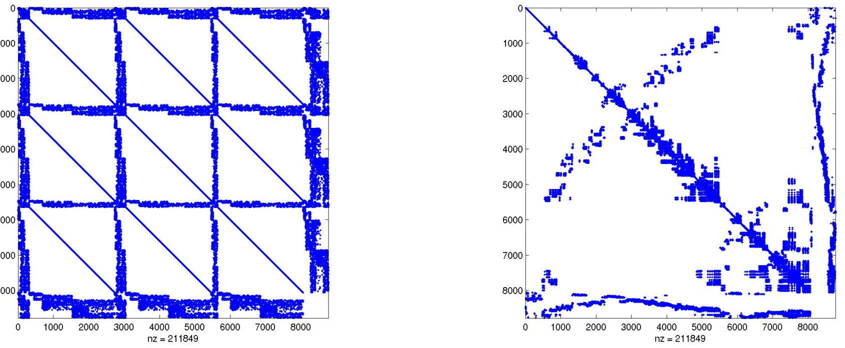
6 Numerical Tests

We have used Matlab 7, on a AMD X2 4200+ (dual core - 1024MB of cache), with 4Gb of RAM. The tested matrices are issued by the following three-dimensional problems:

1. a channel with a matrix of size 42,630 with 0.072% nonzeros,
2. a channel with step, the matrix size is 42,630 with 0.072% nonzeros,
3. a compartment of an actual reservoir, the matrix size is 34,578 with 0.085% nonzeros.

The problems were tested with CFL=1 and CFL=5, and Reynolds number of 10,000. The convergence criterion is Matlab's default: relative residual less than 10^{-6}. All the measures were done for one time step of a simulation when solving one linear system of the problems. The Krylov methods have 200 as the maximum number of iterations and GMRes was implement without restart.

As long as reorderings are concerned, some remarks are necessary. Firstly, without any reordering the time for solving a typical problem is 80 times slower than with AMD



(a) Saddle point matrix without reordering.

(b) Saddle point matrix with AMD reordering.

Fig. 3: Examples of saddle point matrix before and after reordering.

or SRCM reorderings. As matter of fact, from a performance viewpoint, the reordering is the most important step in the reservoir problem simulation as soon as, as depicted in Figure 3a, the arrowed structure of the matrix implies in a tremendous fill in process when computing a complete or an incomplete factorization. Secondly, the two ordering schemes are equivalent with a very light bias towards SRCM, so we are addressing figures only for this alternative.

The fill-in zero preconditioners, ILU(0) and ICC(0), did not converged or become singular for every test, for the three problems, so we are not presenting figures for this preconditioners. Also the probing construction for the Schur complement matrix, as we should have expected, presented a very bad behavior.

We present three types of tables: tables with complete preconditioners comparisons for the reservoir problem - Tables 1, 2, and 3-, tables comparing the performance of the methods and preconditioners for the three problems with CFL =1 and =5 - Tables 4 and 5-, and tables comparing the segregated and coupled methods for the reservoir simulation - Tables 6 and 7.

For every table, the labels mean: **Preconditioner**, the preconditioner - None (without a preconditioner), Diagonal (classical diagonal), Lumped (lumped diagonal), ILUT(10^{-3}) (ILUT with an absolute threshold of 10^{-3}), Projection (the projection method used as a preconditioner), MGW1 (block diagonal described in section 5.1), and MGW2 (block triangular described in section 5.1)-, **Approx.** stands for the kind of Schur complement matrix approximation (**cam** is the complete approximated matrix), as described in section 3, **T.Prec** is related to the time, in seconds, of the preconditioner construction, **T.Sol** is the time, in seconds, of the complete iterations of the preconditioned iterative Krylov methods, **Error** is the relative Euclidian norm of the actual error, i.e., the norm of the difference between the correct and the approximated solution divided by. the correct solution norm. For computing the actual solution, we have used the backslash Matlab operator with iterative refinement in order to reach an accuracy of 10^{-15}.

In Tables 1 to 3, the column **Iter.** shows the number of iterations of the Krylov method. In Tables 4 and 5, **Problem** is the problem kind - a channel, a channel with a step, or a reservoir branch. In Tables 4 to 7, **Method** is the segregated or the coupled methods, **Solver** means the type of segregated or coupled method. **IR** means the iterative refinement for the projection method. In Tables 6 and 7, **T.Tot** is the total time for both the construction of the preconditioner and the iterative method.

There are still some conventions. When the iterative method fails, we use **nc**. In the Approx. column, $\cdots$ means that there is no approximation for the Schur matrix, as we are treating, in this case, the coupled approach with a preconditioner that does not take into account the saddle point structure.

In Tables 1 to 3, we show figures for the reservoir problem, comparing different methods, Schur complement matrix approximations, and preconditioners using the same reordering scheme (SRCM) and CFL=5. We can observe that in Table 1 using the coupled method with left-preconditioned GMRes, Projection and MGW2 present almost the same behavior, and ILUT(10^{-3}) although has a better numerical performance spends too much time for its construction. As we have mentioned before the Schur complement "cam" approximation is still too expensive in the construction step although with a good numerical result. In Table 2 using the coupled method with right-preconditioned GMRes, we can observe that almost the same behavior is found, with MGW2 and Projection alternatives performing similarly. Table 3, using BiCGStab for the same problem, shows that this iterative method does not performs well for this problem. Another remark is that the preconditioners that do not take into account the saddle point structure do not perform well; in this class, ILUT(10^{-3}) although having a good numerical performance, has an expensive construction step.

In Tables 4 and 5, all three problems are compared for the times of construction and iterative method convergence, with CFL=1 and CFL=5, respectively. The saddle point based preconditioners with quite simple approximations for the Schur complement matrix performs quite well. For the easier problems BiCGStab converges with no problems, outperforming GMRes.

Finally, in Tables 6 and 7, we address comparisons between the segregated and the coupled methods. For the segregated problem, we have implemented the preconditioner ICC(10^{-3}) for the (1,1)-block and ILUT(10^{-3})for the Schur complement block. We can see that, depending on the accuracy of the solution, the segregated method with iterative refinement is quite competitive with the more expensive coupled approaches.

Acknowledgments The first author would like to thank Norberto Mangiavacchi for fruitful discussions during the preparation of this work. The authors would like to thank to one of the anonymous referees for his (or her) careful reading of the submitted abstract and we apologize for not been able to take into account every remark, but we are developing all the given ideas and suggestions, and we hope to treat them in our future researches.

References

1. Duchemin, E.: Hydroélectricité et gaz à effet de serre: évaluation des emissions et identification de processus biogéochimiques de production. PhD thesis, Université du Québec

Table 1: Reservoir, coupled approach, left-preconditioned GMRes, CFL=5.

Preconditioner	Approx. of S	T.Prec.	T.Sol.	Error	Iter.
None	$\cdots$		57.9	0.94	200
Diagonal	$\cdots$	1.21	58.1	0.92	200
Lumped	$\cdots$	1.21	58.3	0.99	200
ILUT(10^{-3})	$\cdots$	76.6	12.9	6.2e-7	9
Projection	diagonal	2.25	12.1	3.6e-5	13
	lumped	2.26	13.5	9.2e-5	12
	cam	51.2	28.3	7.2e-6	17
MGW1	diagonal	2.23	26.9	1.1e-4	27
	lumped	2.10	26.9	1.1e-4	27
	cam	51.2	46.2	6.7e-6	35
MGW2	diagonal	2.24	12.7	3.0e-5	13
	lumped	2.10	12.8	3.0e-5	13
	cam	51.2	23.6	9.9e-6	17

Table 2: Reservoir, coupled approach, right-preconditioned GMRes, CFL=5.

Preconditioner	Approx. of S	T.Prec.	T.Sol.	Error	Iter.
Diagonal	$\cdots$	1.21	58.0	0.92	200
Lumped	$\cdots$	1.21	57.9	0.99	200
ILUT(10^{-3})	$\cdots$	76.6	11.8	6.2e-7	9
Projection	diagonal	2.25	19.6	1.9e-5	20
	lumped	2.26	16.6	8.4e-6	16
	cam	51.2	31.6	4.2e-7	20
MGW1	diagonal	2.24	39.0	5.4e-5	38
	lumped	2.10	39.0	5.4e-5	38
	cam	51.1	51.9	5.4e-5	40
MGW2	diagonal	2.24	18.5	6.5e-5	19
	lumped	2.10	18.5	6.5e-5	19
	cam	51.2	26.4	2.3e-6	20

Table 3: Reservoir, coupled approach, preconditioned BiCGStab, CFL=5.

Preconditioner	Approx. of S	T.Prec.	T.Sol.	Error	Iter.
None	$\cdots$		14.3	0.92	190
Diagonal	$\cdots$	1.21	15.4	0.97	134
Lumped	$\cdots$	nc	nc	nc	nc
ILUT(10^{-3})	$\cdots$	76.6	11.5	1.1e-6	5
Projection	diagonal	2.25	269	4.6e-9	60
	lumped	2.26	315	1e-3.0	80
	cam	51.2	43.5	5.8e-8	14
MGW1	diagonal	2.24	314	7.0e-6	80
	lumped	2.10	314	7.0e-6	80
	cam	51.2	36.2	2.0e-7	14
MGW2	diagonal	2.24	314	7.0e-6	80
	lumped	2.04	314	7.0e-9	80
	cam	51.2	36.2	2.0e-7	14

Table 4: Comparing times in the three problems, with CFL=1.

Problem	Solver	Preconditioner	Approx.	T.Prec.	T.Sol.	Error
Channel	RP-GMRes	Projection	diagonal	3.30	8.20	3.1e-8
	RP-GMRes	MGW 2	lumped	3.13	9.01	6.3e-8
Step	RP-GMRes	Projection	diagonal	3.30	8.35	9.9e-9
	RP-GMRes	MGW 2	lumped	3.12	9.08	1.2e-7
Reservoir	LP-GMRes	MGW 2	lumped	2.04	9.35	8.0e-6
	LP-GMRes	Projection	diagonal	2.19	9.30	4.6e-6

Table 5: Comparing times in the three problems, with CFL=5.

Problem	Solver	Preconditioner	Approx.	T.Prec.	T.Sol.	Error
Channel	BiCGStab	Projection	lumped	0.49	1.17	1.4e-7
	RP-GMRes	Projection	diagonal	0.49	1.22	9.0e-8
Step	RP-GMRes	MGW 2	lumped	3.41	12.1	3.3e-8
	BiCGStab	MGW 1	lumped	3.42	12.2	2.9e-8
Reservoir	LP-GMRes	MGW 2	lumped	2.10	12.8	3.0e-5
	LP-GMRes	Projection	diagonal	2.25	12.1	3.6e-5

Table 6: Comparing segregated/coupled method for the reservoir problem with CFL=1.

Method	Solver	Preconditioner	Approx.	T.Tot	Error
Segregated	Projection	MGW 1	lumped	1.20	1.1e-2
	IR		lumped	5.94	1.7e-4
Coupled	LP-GMRes	MGW 2	lumped	11.4	8.0e-6
	BiCGStab	Projection	diagonal	18.3	1.8e-9

Table 7: Comparing segregated/coupled method for the reservoir problem with CFL=5.

Method	Solver	Preconditioner	Approx.	T.Sol	Error
Segregated	Projection	MGW 1	lumped	3.30	8.6e-2
	IR		lumped	7.80	2.2e-3
Coupled	LP-GMRes	MGW 2	lumped	14.9	3.0e-5
	BiCGStab	Projection	cam	94.7	5.8e-8

Montréal (Avril 2000)

2. Rosa, L.P., dos Santos, M.A.: Certainty and uncertainty in the science of greenhouse gas emissions from hydroelectric reservoirs. In: WCD Thematic Review Environmental Issues II.2. Volume II., Cape Town, Secretariat of the World Commission on Dams (November 2000)
3. Carvalho, L.M., Mangiavacchi, N., Soares, C.B.P., Rodrigues, W.F., Costa, V.S.: Comparison of preconditioners for the simulation of hydroeletric reservoir flooding. In: Proceedings of the 2007 International Conference On Preconditioning Techniques For Large Sparse Matrix Problems In Scientific And Industrial Applications. (2007)
4. Mangiavacchi, N., Carvalho, L.M., Soares, C.B.P., Fortes, W., Costa, V.S.: Preconditioners for hydropower plant reservoir flow simulations. In: Annals of the 6th International Congress on Industrial and Applied Mathematics. (2007)
5. Batchelor, G.K.: An introduction do fluid dynamics. Cambridge University Press (2000)
6. Lifshitz, E.M., Landau, L.D.: Fluid Mechanics, Volume 6 (Course of Theoretical Physics). 2nd edn. Butterworth-Heinemann (1987)
7. Arnold, D.N., Brezzi, F., Fortin, M.: A stable finite element for the Stokes equations. Calcolo **21**(4) (December 1984) 337–344
8. Boffi, D., Brezzi, F., Demkowicz, L.F., Durn, R.G., Falk, R.S., Fortin, M.: Mixed Finite Elements, Compatibility Conditions, and Applications. Springer (2008)
9. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Springer–Verlag, Berlin (1994)
10. Durran, D.R.: Numerical Methods for Wave Equations in Geophysical Fluid Dynamics. Springer (1998)
11. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numerica **14** (May 2005) 1–137
12. Chorin, A.J.: Numerical solution of the Navier-Stokes equations. Mathematics of Computation **22** (1968) 745–762
13. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand **49** (1952) 409–436
14. Reid, J.K.: On the method of conjugate gradients for the solution of large sparse systems of linear equations. In Reid, J.K., ed.: Large Sparse Sets of Linear Equations, New York, Academic Press (1971) 231–254
15. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3) (1986) 856–869
16. Van der Vorst, H.: BICGSTAB: a fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems. SIAM Journal on Scien. and Stat. Computing **13**(2) (March 1992) 631–644
17. Chan, T.F., Mathew, T.: The interface probing technique in domain decomposition. SIAM J. Matrix Analysis and Applications **13** (1992)
18. Siefert, C., de Sturler, E.: Probing methods for saddle-point problems. ETNA **22** (2006) 163–183

19. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. 2nd edn. SIAM (2002)
20. Wilkinson, J.H.: Rounding Errors in Algebraic Processes. Notes on Applied Science No. 32, Her Majesty's Stationery Office, London (1963) Also published by Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New York, 1994.
21. Patankar, S.V.: Numerical heat transfer and fluid flow. Washington, DC, Hemisphere Publishing Corp., 1980. 210 p. (1980)
22. Murphy, M.F., Golub, G.H., Wathen, A.J.: A note on preconditioning for indefinite linear systems. SIAM Journal on Scientific Computing **21**(6) (2000) 1969–1972
23. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM Journal on Scientific Computing **14**(2) (1993) 461–469
24. Simoncini, V., Szyld, D.B.: Theory of inexact Krylov subspace methods and applications to scientific computing. SIAM Journal on Scientific Computing **25**(2) (2003) 454–477
25. de Sturler, E.: Inner-outer methods with deflation for linear systems with multiple right-hand side. In: Proceedings of the Householder Symposium on Numerical Algebra, Pontresina, Switzerland, Householder Symposium XIII (1996) 193–196
26. George, A., Liu, J.W.: The evolution of the minimum degree ordering algorithm. SIAM Review **31**(1) (1989) 1–19
27. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proc. 24th Nat. Conf. ACM. (1969) 157–172
28. George, A., Liu, J.: Computer solution of large sparse positive definite systems. Computational Mathematics. Prentice-Hall, Englewood Cliffs (1981)