Parallel Inexact Constraint Preconditioners for Saddle Point Problems

Luca Bergamaschi and Angeles Martinez

Department of Mathematical Methods and Models for Scientific Applications
University of Padua, via Trieste 63, 35121 Padova, Italy
berga@dmsa.unipd.it, acalomar@dmsa.unipd.it

Abstract. In this paper we propose a parallel implementation of the FSAI preconditioner to accelerate the PCG method in the solution of symmetric positive definite linear systems of very large size. This preconditioner is used as *building block* for the construction of an indefinite Inexact Constraint Preconditioner (ICP) for saddle point-type linear systems arising from Finite Element (FE) discretization of 3D coupled consolidation problems. The FSAI-ICP preconditioner, based on an efficient approximation of the inverse of the $(1,1)$ block proves very effective in the acceleration of the BiCGSTAB iterative solver in parallel environments. Numerical results on a number of realistic test cases of size up to 6×10^6 unknowns and 3×10^8 nonzeros show the almost perfect scalability of the overall code up to 512 processors.

Keywords: Parallel computing Preconditioning Krylov subspace methods coupled consolidation.

1 Introduction

The time-dependent displacements and fluid pore pressure in porous media are controlled by the consolidation theory. This was first mathematically described by Biot [8], who coupled the elastic equilibrium equations with a continuity or mass balance equation to be solved under appropriate boundary and initial flow and loading conditions.

The coupled consolidation equations are typically solved numerically using FE in space, thus giving rise to a system of first-order differential equations whose solution is addressed by an appropriate time marching scheme. A major computational issue is the repeated solution in time of the resulting discretized indefinite equations, which can be generally written as

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \qquad \text{where} \qquad \mathcal{A} = \begin{bmatrix} K & B^T \\ B & -C \end{bmatrix}. \tag{1}$$

The sub-matrices K and C are both symmetric and positive definite (SPD). Denoting with m the number of FE nodes, $C \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times n}$, and $K \in \mathbb{R}^{n\times n}$, where n is equal to $2m$ or $3m$ according to the spatial dimension of the problem.

E. Jeannot, R. Namyst, and J. Roman (Eds.): Euro-Par 2011, LNCS 6853, Part II, pp. 78–89, 2011.

Matrix $\mathcal{A}$ in (1) is a classical example of saddle point problem, which is encountered in other fields as well including constrained optimization, least squares, and Navier-Stokes equations. Because of the large size of realistic three-dimensional (3D) consolidation models (and particularly so in problems related to fluid withdrawal/injection from/into geological formations) the use of iterative solvers is strongly recommended against direct factorization methods. However, well established iterative methods such as Krylov subspace methods are very slow or even fail to converge if not conveniently preconditioned. The constraint preconditioners for Krylov solvers in the solution of saddle point problems have been studied by a number of authors [1,4,5,11,14]. In this work we propose a fully explicit parallel ICP based on the FSAI preconditioner [16] of the matrices K and S where S is an approximate Schur complement of a block matrix $\mathcal{M}$ resembling $\mathcal{A}$. The FSAI preconditioner is based on *prefiltration* and *postfiltration* techniques and allows to choose nonzeros in the preconditioner factors in the same position as those of $\widetilde{A}^{d_K}$, where $\widetilde{A}$ is an sparse approximation of A obtained by eliminating the small entries below a given threshold and $d_K = 1, 2, 4$.

We have developed parallel codes which implement both the FSAI-PCG solver for solution of $K\boldsymbol{x} = \boldsymbol{b}$ and the BiCGSTAB solver preconditioned with the parallel FSAI-ICP preconditioner described above. We show numerical results obtained in the solution of a number of problems of large size arising from 3D FE discretization of realistic engineering problems.

The paper is organized as follows. Section 2 gives a brief description of the consolidation equations. In Section 3 we describe the Inexact Constraint Preconditioner and recall the main spectral properties of the block preconditioned matrices. Section 4 describes the parallel preconditioner used in this work and explains in detail how it is implemented and applied during the BiCGSTAB iteration. Section 5 contains the numerical results obtained with PCG accelerated with FSAI preconditioner on seven test cases arising from realistic engineering applications as well as the results of the FSAI-ICP code on a difficult problem arising from a Coupled Consolidation model. We include also a scalability study of the parallel solution of system (1). Finally, some conclusions are stated in Section 6.

2 Finite Element Coupled Consolidation Equations

The system of partial differential equations governing the 3D coupled consolidation process in fully saturated porous media is derived from the classical Biot's formulation [8] and successive modifications as:

$$(\lambda + \mu)\frac{\partial \epsilon}{\partial i} + \mu \nabla^2 u_i \;=\; \alpha \frac{\partial p}{\partial i} \qquad i = x,\ y,\ z \tag{2}$$

$$\frac{1}{\gamma}\nabla(k\nabla p) \;=\; \left[\phi\beta + c_{br}(\alpha - \phi)\right]\frac{\partial p}{\partial t} + \alpha\frac{\partial \epsilon}{\partial t} \tag{3}$$

where c_{br} and β are the volumetric compressibility of solid grains and water, respectively, ϕ is the porosity, k the medium hydraulic conductivity, ϵ the medium

volumetric dilatation, α the Biot coefficient, λ and μ are the Lamé constant and the shear modulus of the porous medium, respectively, γ is the specific weight of water, t is time, and p and u_i are the incremental pore pressure and the components of incremental displacement along the $i-$direction, respectively.

Use of standard linear Galerkin FE in space yields a system of first order differential equations which can be integrated by the Crank-Nicolson scheme. The resulting linear system has to be repeatedly solved to obtain the transient displacements and pore pressures. The nonsymmetric matrix controlling the solution scheme reads:

$$A = \begin{bmatrix} K/2 & -Q/2 \\ \dfrac{Q^T}{\Delta t} & H/2 + \dfrac{P}{\Delta t} \end{bmatrix} \tag{4}$$

where K, H, P and Q are the elastic stiffness, flow stiffness, flow capacity and flow-stress coupling matrices, respectively. Matrix A can be readily symmetrized by multiplying the upper set of equations by 2 and the lower set by $-\Delta t$, thus obtaining the sparse 2×2 block symmetric indefinite matrix (1) where $B = -Q^T$ and $C = \Delta t H/2 + P$.

3 Inexact Constraint Preconditioners

To solve system (1) we look for a preconditioner $\mathcal{M}^{-1}$ where

$$\mathcal{M} = \begin{bmatrix} G_1 & B^T \\ B & -C \end{bmatrix},$$

with G_1 an SPD approximation of the 1×1 block K. Its inverse, G_1^{-1}, which can be viewed as a preconditioner for K, is assumed to be explicitly known. To fulfill such a requirement we compute G_1^{-1} using FSAI [15,16] which is readily available in the factorized form $K^{-1} \simeq G_1^{-1} = W_1^T W_1$. The Inexact Constraint Preconditioner (ICP) is written as ${\mathcal{M}_{\mathcal{I}}}^{-1}$ where:

$$\mathcal{M}_I^{-1} = \begin{bmatrix} I_n & -G_1^{-1}B^T \\ 0 & I_m \end{bmatrix} \begin{bmatrix} G_1^{-1} & 0 \\ 0 & -G_S^{-1} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -BG_1^{-1} & I_m \end{bmatrix}. \tag{5}$$

I_i begin $i \times i$ identity matrix and G_S^{-1} an approximation of the inverse of the Schur complement matrix S relative to $\mathcal{M}$: $S = BG_1^{-1}B^T + C$.

A further approximation can be used by simply neglecting the right matrix in the above expression thus obtaining a Triangular ICP preconditioner:

$$\mathcal{M}_T^{-1} = \begin{bmatrix} I_n & -G_1^{-1}B^T \\ 0 & I_m \end{bmatrix} \begin{bmatrix} G_1^{-1} & 0 \\ 0 & -G_S^{-1} \end{bmatrix}. \tag{6}$$

Following the approach in [3], we construct an approximate Schur complement $\widehat{S} = BG_2^{-1}B^T + C$, with the aim of reducing its fill-in. G_2^{-1} is computed as a further (sparser) FSAI approximation for the inverse of the structural block. A third FSAI preconditioner is used to approximate the inverse of $\widehat{S}$, $G_S^{-1} \approx \widehat{S}^{-1}$.

3.1 Eigenvalue Distribution of the Preconditioned Matrices

Let G_1 and G_S be SPD approximations of K and $S = C + BG_1^{-1}B^\top$, respectively. G_1^{-1} and G_S^{-1} can also be viewed as preconditioners for the corresponding matrices, so that we can define the following SPD preconditioned matrices:

$$K_P = G_1^{-1/2} K G_1^{-1/2} \qquad \text{and} \qquad S_P = G_S^{-1/2} S G_S^{-1/2}$$

Let us assume that

$$\begin{aligned} 0 < \alpha_K = \lambda_{\min}(K_P) < 1 < \lambda_{\max}(K_P) = \beta_K, \\ 0 < \alpha_S = \lambda_{\min}(S_P) < 1 < \lambda_{\max}(S_P) = \beta_S. \end{aligned} \tag{7}$$

The conditions $1 \in [\alpha_K, \beta_K]$ and $1 \in [\alpha_S, \beta_S]$ are very often fulfilled in practice since preconditioners G_1 and G_S are expected to cluster eigenvalues around unit.

The following two theorems give bounds on the eigenvalues of the preconditioned matrix using ICP and TICP. They show that the eigenvalues of the preconditioned matrix are clustered around one if those of the preconditioned K and the preconditioned Schur complement are so. An exhaustive spectral analysis can be found in [2]. We denote a generally complex eigenvalue λ as $\lambda_R + i\lambda_I$.

Theorem 1

If $\beta_K < 2$ then the real eigenvalues of the ICP preconditioned matrix satisfy:

$$\min\left\{\alpha_K, \frac{\alpha_S}{\beta_K}\right\} \le \lambda \le \max\{(2-\alpha_K)\beta_S, \beta_K\}.$$

If $\lambda_I \neq 0$ then

$$\frac{\alpha_K + \alpha_S(2-\beta_K)}{2} \le \lambda_R \le \frac{\beta_K + \beta_S(2-\alpha_K)}{2} \qquad |\lambda_I| \le \sqrt{\beta_S}\max\{1-\alpha_K, \beta_K - 1\}.$$

Proof. See proof of Theorem 3 in [2].

Theorem 2

The eigenvalues of $\mathcal{M}_T^{-1}\mathcal{A}$ satisfy the following bounds. If $\lambda_I \neq 0$ then

$$|\lambda - 1| \le \sqrt{1-\alpha_K}, \qquad \textit{and} \qquad \frac{\alpha_K}{2} \le \lambda_R \le \min\left\{\frac{1+\beta_S}{2}, 2\right\}.$$

The real eigenvalues satisfy:

$$\min\left\{\alpha_K, \frac{\alpha_S}{\beta_K + \alpha_S}\right\} \le \lambda_R \le \beta_S + \beta_K.$$

Proof. See proof of Theorem 5 in [2].

4 FSAI-Based ICP

The FSAI preconditioner, initially proposed in [15] and [16], has been later developed and implemented in parallel by Bergamaschi et al. in [6]. Here, we only shortly recall the main features of this preconditioner. Given and SPD matrix K the FSAI preconditioner approximately factorize its inverse as a product of two sparse triangular matrices as

$$K^{-1} \approx G^{-1} = W^T W.$$

The choice of nonzeros in W are based on a sparsity pattern which in our work may be the same as $\widetilde{K}^k$ where $\widetilde{K}$ is the result of *prefiltration* [7] of K i.e. dropping of all elements below of a threshold parameter δ. In the present paper we allow the power k to be equal to $1, 2$ or 4. The entries of W are computed by minimizing the Frobenius norm of $I - WL$ where L is the exact Cholesky factor of K. The computed W is then sparsified by dropping all the elements which are below a second tolerance parameter (ε). The final FSAI preconditioner is therefore related to the following three parameters: δ, prefiltration threshold; $d_K = 1, 2, 4$, power of K generating the sparsity pattern; ε, postfiltration threshold.

Recalling equation (5), the full ICP can be written as:

$$\begin{aligned} \mathcal{M}_I^{-1} &= \begin{bmatrix} I_n & -W_1^T W_1 B^T \\ 0 & I_m \end{bmatrix} \begin{bmatrix} W_1^T W_1 & 0 \\ 0 & -W_S^T W_S \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -B W_1^T W_1 & I_m \end{bmatrix} \\ &= \begin{bmatrix} W_1^T & -W_1^T W_1 B^T W_S^T \\ 0 & W_S^T \end{bmatrix} \begin{bmatrix} W_1 & 0 \\ W_S B W_1^T W_1 & -W_S \end{bmatrix} \end{aligned} \tag{8}$$

where $G_1^{-1} = W_1^T W_1$ and W_S is the FSAI factor of the approximate Schur complement matrix $\widetilde{S}$, $\widetilde{S}^{-1} = W_S^T W_S$. The Schur complement matrix S is evaluated as $S = B W_2^T W_2 B^T + C = S_0 + C$, W_2 being the triangular factor of a sparser FSAI approximation of K^{-1}, obtained from W_1 by a further postfiltration.

Analogously the Triangular ICP can be written as

$$\mathcal{M}_T^{-1} = \begin{bmatrix} W_1^T & -W_1^T W_1 B^T W_S^T \\ 0 & W_S^T \end{bmatrix} \begin{bmatrix} W_1 & 0 \\ 0 & -W_S \end{bmatrix}. \tag{9}$$

The application of $\mathcal{M}^{-1}$ requires the explicit computation of the Schur complement matrix S whose construction may be time and memory consuming, However, it should be noted that the evaluation of $S_0 = B W_2^T W_2 B^T$, which involves the main computational burden in building S, is independent of the time step Δt, and therefore can be done just once at the beginning of the simulation. The construction of the preconditioner is therefore based on the following parameters:

1. δ_1, d_K and ε_1, for the 1st FSAI preconditioner (W_1).
2. ε_2, postfiltration threshold for W_2
3. δ_S, d_S and ε_S, for the FSAI preconditioner applied to the Schur complement matrix (W_S).

4.1 Parallel Implementation

Our code is written in FORTRAN 90 and exploits the MPI library for exchanging data among the processors. We used a block row distribution of all matrices, that is, with complete rows assigned to different processors. All these matrices are stored in static data structures in CSR format.

Any row i of matrix W of FSAI preconditioner is computed independently of each other, by solving a small SPD dense linear system of size n_i equal to the number of nonzeros allowed in row i of W. Some of the rows which contribute to form this linear system may be non local to processor i and should be received from other processors. To this aim we implemented a routine called *get_extra_rows* which carries out all the row exchanges among the processors, before starting the computation of W, which proceed afterwards entirely in parallel. Since the number of non local rows needed by each processor is relatively small we chose to temporarily replicate these rows on auxiliary data structures. Once W is obtained a parallel transposition routine provides every processor with its part of W^T.

The FSAI and the FSAI-ICP preconditioners will be used to accelerate the PCG and the BiCGSTAB Krylov subspace methods. These iterative solvers are essentially based on matrix-vector products. We made use of an optimized parallel matrix-vector product which has been developed in [17] showing its effectiveness up to 1024 processors.

5 Numerical Results

5.1 Solution of $\boldsymbol{Kx = b}$.

Since the key of the success of ICP is related to the goodness of the preconditioner for matrix K (numerical experience shows that the Schur complement matrix is instead well-conditioned), we analyze the performance of our FSAI preconditioner when used within the PCG method to solve a linear system $K\boldsymbol{x} = \boldsymbol{b}$.

The test cases are all realistic examples of large size arising from 2D and 3D FE discretization of geomechanical problems. In detail:

1. FAULT-639: arises from the numerical solution by a linear FE of the inequality-constrained minimization problem governing the mechanical equilibrium of a 3D body with contact surfaces [12]. The contact is solved with the aid of a penalty formulation that gives rise to an SPD ill-conditioned linear system.
2. STOCF-729: arises from the FE integration of the diffusion partial differential equation governing the 3D transient flow of groundwater in saturated porous media. The problem is solved assuming a stochastic distribution of the hydraulic conductivity tensor with a large permeability contrast in adjacent elements.
3. GEO-1438: arises from a regional geomechanical model of the sedimentary basin underlying the Venice lagoon discretized by a linear FE with randomly heterogeneous properties [18].

4. FLAN-1565: arises from the mechanical equilibrium of a steel flange discretized by a 3D 8-node brick FE [13].
5. HOOK-1498: arises from the mechanical equilibrium of a steel hook discretized by 3D 4-node tetrahedral FE [13].
6. PO-878: arises in the simulation of the consolidation of a real gas reservoir of the Po Valley, Italy, used for underground gas storage purposes (for details, see [9]).
7. CUBE-6536: simulates the compaction of a shallow confined aquifer due to groundwater withdrawal in a representative 3D sedimentary basin at a regional scale. The discretization employes 1 171140 grid nodes, giving raise to a very large problem of more than 6 million unknowns.

The size and number of nonzero terms for each matrix is provided in Table 1. The linear system is solved by PCG using the exact solution as a vector of all ones. The exit test for the iterative solver is $\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{b}\|} \leq 10^{-10}$, $\boldsymbol{r}_k$ being the relative residual at iteration k. Each matrix has been preliminarily reordered by a Reverse Cuthill McKee (RCM) algorithm [10].

Table 1. Size n and number of nonzeros `nnz` of the test matrices

name	n	nnz
FAULT-639	638 812	14 626 683
STOCF-729	729 400	10 765 586
GEO-1438	1 437 960	63 156 690
FLAN-1565	1 564 794	117 406 044
HOOK-1498	1 498 023	60 917 445
PO-878	878 355	38 896 749
CUBE-6353	6 353 100	282 438 234

All tests are performed on the IBM SP6/5376 cluster at the CINECA Centre for HCP, equipped with IBM Power6 processors at 4.7 GHz with 168 nodes, 5376 computing cores, and 21 Tbytes of internal network RAM. The code is written in Fortran 90 and compiled with `-O4 -q64 -qarch=pwr6 -qtune=pwr6 -qnoipa -qstrict -bmaxdata:0x70000000` options.

In Table 2 we report the results of the PCG runs for the seven test cases and a number of combination of the FSAI parameters. In particular we provide the number of iteration (iter) the density of the FSAI preconditioner computed as $\rho = \frac{\mathtt{nnz}(G_1^{-1})}{\mathtt{nnz}(K)}$ as well as three CPU times referring to the cost of FSAI computation (T_P), the cost of iterative solver (T_{sol}) and the total time ($T_{tot} = T_P + T_{sol}$).) For a fixed test case all the runs have been performed using a fixed number of processors.

Inspection of Table 2 reveals that the choice of $d_K = 4$ produces in all tests the smallest number of iterations and (with the only exception of Problem FLAN-1565) the smallest T_{sol} CPU time. However, in some instances the large cost to compute the FSAI preconditioner may greatly influence the total CPU time.

5.2 Parallel Results and Scalability

We will use a strong scaling measure to see how the CPU times vary with the number of processors for a fixed total problem size. We will denote with T_p

Table 2. Iteration number, iter, density ρ of the preconditioner, CPU times obtained using a fixed number of processors for each combination of parameters. Best iteration number, smallest T_{sol} and T_{tot} for each test are printed in boldface.

name	p	d_K	δ	ϵ	iter	ρ	T_P	T_{sol}	T_{tot}
FLAN-1565	64	4	0.1	0.1	4546	0.12	12.60	67.62	80.22
		4	0.1	0.01	**2785**	1.17	11.79	82.06	93.85
		4	0.1	0.05	3909	0.29	12.47	63.44	75.91
		2	0.1	0.1	5414	0.10	0.81	**62.49**	**63.30**
		1	0.01	0.1	6064	0.09	0.72	75.55	76.27
FAULT-639	16	4	0.1	0.01	**674**	1.32	5.90	21.92	27.82
		4	0.2	0.01	986	0.18	0.35	**13.54**	**13.89**
		2	0.2	0.01	1667	0.10	0.23	26.35	26.58
		2	0	0.01	938	1.41	8.03	29.64	37.67
		1	0	0.01	1745	0.56	0.83	38.25	39.08
HOOK-1498	16	4	0.1	0.1	3511	0.28	49.29	**142.05**	191.34
		4	0.1	0.01	**2362**	2.76	46.38	267.64	314.02
		2	0.2	0.01	5195	0.10	0.49	215.56	216.05
		1	0.01	0.1	4164	0.18	1.12	149.00	**150.12**
		1	0.01	0.01	3416	0.66	0.96	168.83	169.79
GEO-1438	16	4	0.1	0.1	585	0.34	20.12	**20.34**	40.46
		4	0.1	0.01	**405**	2.13	26.77	42.93	69.70
		2	0.1	0.1	766	0.21	1.24	34.06	**35.30**
		2	0.1	0.01	671	0.58	1.42	38.65	40.07
		1	0.0	0.01	818	0.65	1.13	45.03	46.16
STOCF-729	16	4	0.1	0.05	**755**	1.61	1.96	17.06	19.02
		4	0.1	0.1	881	0.95	1.51	**9.96**	11.47
		2	0.1	0.01	1230	1.11	0.30	11.75	12.05
		2	0.2	0.1	2030	0.24	0.17	11.00	**11.17**
		1	0.01	0.01	1699	0.77	0.20	15.67	15.87
PO-878	64	4	0.2	0.1	844	0.14	0.27	4.47	**4.74**
		4	0.1	0.1	728	0.26	2.99	**3.55**	6.54
		4	0.1	0.01	**698**	1.42	2.75	7.30	10.05
		2	0.1	0.1	1414	0.17	0.34	6.27	6.61
		1	0.01	0.1	2297	0.13	0.22	8.31	8.53
CUBE-6353	256	4	0.1	0.01	**459**	1.13	5.24	12.29	17.53
		4	0.1	0.1	649	0.20	5.76	**8.56**	**14.32**
		2	0.01	0.01	511	1.09	3.68	16.02	19.70

the total CPU elapsed times expressed in seconds on p processors. As relative measures of the parallel efficiency achieved by the code we denote as $S_p^{(\bar{p})}$ the pseudo speedup computed with respect to the smallest number of processors ($\bar{p}$) used to solve a given problem and $E_p^{(\bar{p})}$ the corresponding efficiency:

$$S_p^{(\bar{p})} = \frac{T_{\bar{p}}\bar{p}}{T_p}, \qquad E_p^{(\bar{p})} = \frac{S_p^{(\bar{p})}}{p} = \frac{T_{\bar{p}}\bar{p}}{T_p p}.$$

Scalability of FSAI-PCG. In Table 3 we report number of iterations and timings in solving problems GEO-1438 and CUBE-6536 by FSAI-PCG with varying number of processors. The parameters used are: $d_K = 4, \delta = 0.1$ and $\varepsilon = 0.1$ for both cases. We also report the scaled speedups and efficiencies for the total CPU time. Speedups larger than p and efficiencies larger than 1 are printed in boldface. They can be put in connection both with cache effects and with the not optimal use of the memory for small number of processors which slow down the performance the code. We note from the table that our code scales almost perfectly up to 128 processors for problem GEO-1438 and up to $p = 512$ for problem CUBE-6536 which is roughly 4 times larger. This is also accounted by the results of Figure 1 where pseudo-speedups vs processor number are displayed in a log-log plot.

Table 3. Number of iterations and timings of FSAI-PCG in the solution of problems GEO-1438 (left) and CUBE-6536 (right)

p	iter	T_P	T_{sol}	T_{tot}	$S_p^{(2)}$	$E_p^{(2)}$
2	585	195.0	175.4	370.4		
4	585	83.5	95.5	179.0	**4.1**	**1.03**
8	585	45.1	40.7	85.8	**8.6**	**1.08**
16	585	20.1	20.3	41.4	**17.9**	**1.12**
32	585	11.0	10.4	21.4	**34.6**	**1.08**
64	585	5.9	5.2	11.1	**66.7**	**1.04**
128	585	3.1	2.7	5.9	125.6	0.98
256	585	2.0	1.8	3.8	195.0	0.76
512	585	1.0	1.4	2.4	308.7	0.60

p	iter	T_P	T_{sol}	T_{tot}	$S_p^{(16)}$	$E_p^{(16)}$
16	459	76.9	198.4	275.3		
32	459	43.6	88.2	131.8	**33.6**	**1.05**
64	459	22.3	45.4	67.7	**65.3**	**1.02**
128	459	10.0	24.1	34.1	**129.8**	**1.01**
256	459	5.2	12.3	17.5	252.2	0.99
512	459	3.2	6.7	9.9	444.8	0.87

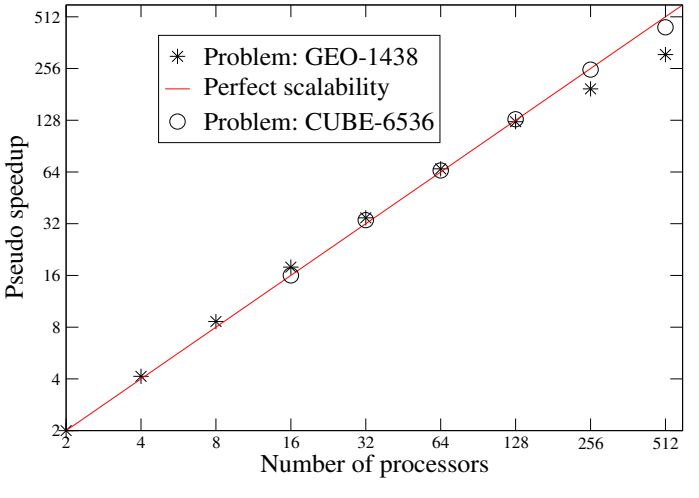

Fig. 1. Speedups vs number of processors. Problems GEO-1438 and CUBE-6536

Scalability of ICP preconditioner. We report in this Section the results obtained in the solution of our saddle point problem with PO-878 as the test example whose main features are summarized as follows.

m	n	N	$\mathtt{nnz}(K)$	$\mathtt{nnz}(B)$	$\mathtt{nnz}(C)$	$\mathtt{nnz}(\mathcal{A})$
292785	878355	1 171 140	38 896 749	12 965 583	4 321 861	69 039 776

We choose this problem among the seven presented in Section 5.1 since it is the most challenging one due to the complexity and the heterogeneity of the geological domain which give raise to a large number of distorted tetrahedra. This produces a very ill-conditioned matrix $\mathcal{A}$, especially for small timesteps. Moreover, the bandwidth is very large and this forces a large amount of interprocessor communication. We solved symmetrized system (4) using $\Delta t = 1$ after an intensive testing to tune the parameters. We choose BiCGSTAB as the iterative solver with the same exit test of Section 5.1.

Table 4. Combinations of parameters and results for problem PO-878 on 128 processors

Run	δ_1	d_K	ϵ_1	ϵ_2	δ_S	d_S	ϵ_S	ρ	iter	T_{P1}	T_{P2}	T_{sol}	T_{tot}
ICP 1	0	1	0	0.01	0.01	1	0	1.23	> 10000	1.4	0.2	> 200.0	> 200.0
ICP 2	0.01	2	0.01	0.1	0.01	1	10^{-3}	1.36	4945	2.9	1.4	127.8	129.2
ICP 3	0.1	4	0.1	0.1	0.01	1	10^{-3}	0.72	1254	2.3	2.2	24.3	26.6
TICP	0.1	4	0.1	0.1	0.01	1	10^{-3}	0.72	3669	3.6	2.3	66.1	68.4

In Table 4 we report for each run the parameters related to the three FSAI approximations as described in the previous sections. We also provide a measure ρ of the density of the preconditioner matrices as:

$$\rho = \rho_1 + \rho_2 = \frac{\mathtt{nnz}(G_1^{-1})}{\mathtt{nnz}(\mathcal{A})} + \frac{\mathtt{nnz}(G_S^{-1})}{\mathtt{nnz}(\mathcal{A})}$$

Parameter ρ gives an indication of the additional core memory needed for computing and storing the preconditioner. We present the following timings, all given in seconds: T_{P1} is the preprocessing time needed to construct G_1^{-1}, G_2^{-1} and S_0, T_{P2} refers to the construction of G_S^{-1} and T_{sol} to the CPU time required by the iterative solver. Finally, $T_{tot} = T_{P2} + T_{sol}$ is the total CPU time.

We report in Table 4 the results of three ICP and one TICP runs employing the three different patterns for the FSAI preconditioner in the approximation of K (with $p = 128$). Using $d_K = 1$ no convergence is attained within 10000 iterations, $d_K = 2$ yields 4945 iterations while with $d_K = 4$ the iterative method obtains convergence after 1254 iterations. From the table we see that only a sparsity pattern for the block K which uses nonzeros far away from the diagonal ($d_K = 4$) allows for a (relatively) fast convergence. We note on passing that the TICP with the same parameters as the third ICP run yields more than twice the ICP iterations and roughly twice CPU time. This is again a consequence of the ill conditioning of this problem.

Table 5. Parallel performance of FSAI-ICP (TICP) code for problem PO-878

run	p	T_{P1}	$S_P^{(2)}$	iter	T_{P2}	T_{sol}	T_{tot}	$S_p^{(2)}$	$E_p^{(2)}$
	2	99.7		1409	83.1	1667.4	1750.5		
	4	42.8	**4.7**	1521	32.3	693.0	725.3	**4.8**	**1.21**
	8	23.6	**8.5**	1518	17.6	350.5	368.1	**9.5**	**1.19**
ICP 3	16	13.6	14.7	1407	10.1	171.8	181.9	**19.3**	**1.20**
	32	7.9	25.2	1397	5.7	92.9	98.6	**35.5**	**1.11**
	64	4.3	46.7	1521	3.4	55.3	58.7	59.7	0.93
	128	2.3	86.4	1254	2.2	24.3	26.6	**131.8**	**1.03**
	2	86.5		3726	67.9	2998.5	3066.4		
	4	42.3	**4.1**	3916	32.2	1523.7	1556.0	3.9	0.99
	8	23.6	7.3	3754	17.5	767.6	783.1	7.8	0.98
TICP	16	13.6	12.7	3842	10.0	397.1	407.1	15.1	0.94
	32	7.9	21.9	3737	5.7	206.8	212.5	28.9	0.90
	64	4.2	41.2	3834	3.4	115.5	118.9	51.6	0.81
	128	2.3	75.2	3669	2.2	57.5	59.7	102.7	0.80

We present in the sequel the results of the scalability study carried out with the FSAI–ICP code when used to solve the PO-878 test problem. We show in Table 5 the results obtained running our FSAI-ICP code using $p = 2$ to $p = 128$, regarding the two preconditioners ICP3 and TICP of Table 4. These results show that our code exhibits almost perfect scalability both on the preprocessing stage and the iterative part. As before, superspeedups can occur due to cache effects and also to the variable number of iterations with different processor number p.

Acknowledgments. We acknowledge the CINECA Iscra Award PARPSEA (2010) for the availability of HPC resources and support. We also thank the four anonymous reviewers who helped improve the overall quality of the paper.

6 Conclusions

This paper describes a parallel block preconditioner for saddle point type linear systems based on an FSAI preconditioner with variable sparsity pattern. We first show that our FSAI-PCG code is efficient and scalable for the solution of $K\boldsymbol{x} = \boldsymbol{b}$. Then the FSAI preconditioner is used to develop a parallel fully explicit ICP within the BiCGSTAB Krylov subspace solver. We have presented a portable parallel code implemented in Fortran 90 using MPI for interprocessor communications. This ensures portability on a whole range of supercomputers. The efficiency of our code is evaluated on realistic engineering applications arising from 3D FE discretization of a coupled consolidation problem exhibiting almost perfect scalability both on the preprocessing stage and the iterative part as well as satisfactory computational efficiency.

References

1. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. 14, 1–137 (2005)
2. Bergamaschi, L.: Eigenvalue distribution of constraint preconditioned saddle point matrices. Numer. Lin. Alg. Appl (submitted) (2011)
3. Bergamaschi, L., Ferronato, M., Gambolati, G.: Mixed constraint preconditioners for the solution to FE coupled consolidation equations. J. Comp. Phys. 227(23), 9885–9897 (2008)
4. Bergamaschi, L., Gondzio, J., Venturin, M., Zilli, G.: Inexact constraint preconditioners for linear systems arising in interior point methods. Comput. Optim. Appl. 36(2–3), 136–147 (2007)
5. Bergamaschi, L., Gondzio, J., Zilli, G.: Preconditioning indefinite systems in interior point methods for optimization. Comput. Optim. Appl. 28(2), 149–171 (2004)
6. Bergamaschi, L., Martínez, Á.: Parallel acceleration of Krylov solvers by factorized approximate inverse preconditioners. In: Daydé, M., Dongarra, J., Hernández, V., Palma, J.M.L.M. (eds.) VECPAR 2004. LNCS, vol. 3402, pp. 623–636. Springer, Heidelberg (2005)
7. Bergamaschi, L., Martínez, A., Pini, G.: An efficient parallel MLPG method for poroelastic models. CMES: Computer and Modeling in Engineering & Sciences 49(3), 191–216 (2009)
8. Biot, M.A.: General theory of three-dimensional consolidation. J. Appl. Phys. 12(2), 155–164 (1941)
9. Castelletto, N., Ferronato, M., Gambolati, G., Janna, C., Teatini, P., Marzorati, D., Cairo, E., Colombo, D., Ferretti, A., Bagliani, A., Mantica, S.: 3D geomechanics in UGS projects: a comprehensive study in northern Italy. In: Proceedings of the 44th US Rock Mechanics Symposium, Salt Lake City, UT (2010)
10. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 1969 24th National Conference, pp. 157–172. ACM, New York (1969)
11. D'Apuzzo, M., De Simone, V., di Serafino, D.: On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods. Comput. Optim. Appl. 45(2), 283–310 (2010)
12. Ferronato, M., Janna, C., Gambolati, G.: Mixed constraint preconditioning in computational contact mechanics. Comp. Methods App. Mech. Engrg. 197(45-48), 3922–3931 (2008)
13. Janna, C., Comerlati, A., Gambolati, G.: A comparison of projective and direct solvers for finite elements in elastostatics. Adv. Engrg. Soft. 40(8), 675–685 (2009)
14. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indefinite linear systems. SIAM J. Matrix Anal. Appl. 21, 1300–1317 (2000)
15. Kolotilina, L.Yu., Nikishin, A.A., Yeremin, A.Yu.: Factorized sparse approximate inverse preconditionings IV. Simple approaches to rising efficiency. Numer. Lin. Alg. Appl. 6, 515–531 (1999)
16. Kolotilina, L.Yu., Yeremin, A.Yu.: Factorized sparse approximate inverse preconditionings I. Theory. SIAM J. Matrix Anal. Appl. 14, 45–58 (1993)
17. Martínez, A., Bergamaschi, L., Caliari, M., Vianello, M.: A massively parallel exponential integrator for advection-diffusion models. J. Comput. Appl. Math. 231(1), 82–91 (2009)
18. Teatini, P., Ferronato, M., Gambolati, G., Bau, D., Putti, M.: Anthropogenic Venice uplift by seawater pumping into a heterogeneous aquifer system. Water Resour. Res. 46 (2010)