Available online at www.sciencedirect.com

ScienceDirect

Computers and Mathematics with Applications 55 (2008) 2197–2211

An International Journal
computers & mathematics
with applications

www.elsevier.com/locate/camwa

Parallel *MIC*(0) preconditioning of 3D elliptic problems discretized by Rannacher–Turek finite elements

P. Arbenz[a,1], S. Margenov[b,*], Y. Vutov[b,2]

[a] *Institute of Computational Science, ETH Zurich, Universitätstrasse 6, 8092 Zurich, Switzerland*
[b] *Institute for Parallel Processing, Bulgarian Academy of Sciences, Acad. G. Bonchev Bl. 25A, 1113 Sofia, Bulgaria*

Abstract

Novel parallel algorithms for the solution of large FEM linear systems arising from second order elliptic partial differential equations in 3D are presented. The problem is discretized by rotated trilinear nonconforming Rannacher–Turek finite elements. The resulting symmetric positive definite system of equations $A\mathbf{x} = \mathbf{f}$ is solved by the preconditioned conjugate gradient algorithm. The preconditioners employed are obtained by the modified incomplete Cholesky factorization *MIC*(0) of two kinds of auxiliary matrices B that both are constructed as locally optimal approximations of A in the class of M-matrices. Uniform estimates for the condition number $\kappa(B^{-1}A)$ are derived. Two parallel algorithms based on the different block structures of the related matrices B are studied. The numerical tests confirm theory in that the algorithm scales as $\mathcal{O}(N^{7/6})$ in the matrix order N.

Keywords: Nonconforming FEM; *MIC*(0) preconditioning; Parallel algorithms

1. Introduction

Consider the elliptic boundary value problem

$$\begin{aligned}
&-\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}) \quad \text{in } \Omega,\\
&u = 0 \quad \text{on } \Gamma_D,\\
&(a(\mathbf{x})\nabla u(\mathbf{x})) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_N,
\end{aligned} \tag{1.1}$$

where Ω is a polyhedral domain in $\mathbb{R}^3$ with axially parallel faces, $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N \neq \emptyset$, and $a(\mathbf{x}) > 0$ is a piecewise smooth function on Ω. The problem (1.1) can be discretized in various ways. Among the most popular and frequently used methods of approximation are the finite volume method, the Galerkin finite element method (FEM), and the mixed FEM. Many engineering problems need very accurate velocity (flux) determinations in the presence

[*] Corresponding author. Fax: +359 2 8722 349.
E-mail addresses: arbenz@inf.ethz.ch (P. Arbenz), margenov@parallel.bas.bg (S. Margenov), yavor@parallel.bas.bg (Y. Vutov).

[1] Fax: +41 44 632 1374.

[2] Fax: +359 2 8722 349.

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.11.013

of heterogeneities and large jumps in the coefficient $a(\mathbf{x})$. This can be achieved through the mixed FEM. However, the technique of the mixed FEM leads generally to an algebraic saddle point problem that is more difficult and more expensive to solve. An important discovery of Arnold and Brezzi [1] is that the Schur system for the Lagrange multipliers can be obtained also as a discretization of (1.1) by a Galerkin method using nonconforming elements. The application of rotated trilinear hexahedral FEs is studied in this paper.

There are two general approaches to constructing parallel preconditioners, based respectively on (a) domain decomposition, and (b) (block) incomplete factorization. The second approach does not lead to optimal preconditioners in terms of the problem size, but produces highly parallel and efficient preconditioners. Here, we first modify the stiffness matrix A locally to obtain an auxiliary matrix B. Then, the preconditioner is obtained by means of a point-wise modified incomplete Cholesky factorization ($MIC(0)$) of the matrix B. This is a 3D extension of the preconditioning technique first proposed in 2D for skewed five point stencil discretization (see e.g. [2]). Related methods for nonconforming Crouzeix–Raviart and for rotated bilinear FE systems were introduced in [3,4], and in [5,6], respectively. Some earlier ideas and numerical results about 3D case can be found in [7,8].

The reminder of the paper is organized as follows. A brief description of the rotated trilinear nonconforming FEs and the related $MIC(0)$ preconditioning strategies is given in Section 2. The construction of locally optimized parallel $MIC(0)$ preconditioners and the corresponding condition number estimates are presented in Section 3. The next two sections contain respectively the analysis of the parallel algorithms and the related numerical tests.

2. FEM discretization and preconditioning strategies

The weak formulation of problem (1.1) reads as follows: given $f \in L^2(\Omega)$, find $u \in \mathcal{V} \equiv H_D^1(\Omega) = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$, satisfying

$$\mathcal{A}(u, v) = (f, v) \quad \forall v \in H_D^1(\Omega), \text{ where } \mathcal{A}(u, v) = \int_\Omega a(\mathbf{x})\nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})\mathrm{d}\mathbf{x}. \tag{2.1}$$

Let the domain Ω be discretized by the hexahedral partition $\mathcal{T}_h$. We assume that $\mathcal{T}_h$ is aligned with the discontinuities of the coefficient $a(\mathbf{x})$ such that in each element $e \in \mathcal{T}_h$, $a(\mathbf{x})$ is a smooth function. The variational problem (2.1) is then discretized using the finite element method, i.e., the continuous space $\mathcal{V}$ is replaced by a finite dimensional subspace $\mathcal{V}_h$, leading to the following finite element formulation: find $u_h \in \mathcal{V}_h$, satisfying

$$\mathcal{A}_h(u_h, v_h) = (f, v_h) \quad \forall v_h \in \mathcal{V}_h, \text{ where } \mathcal{A}_h(u_h, v_h) = \sum_{e\in\mathcal{T}_h} a(e) \int_e \nabla u_h \cdot \nabla v_h \mathrm{d}\mathbf{x}. \tag{2.2}$$

Here, $a(e) > 0$ is a piecewise constant function, defined by the integral averaged value of $a(\mathbf{x})$ over each element from $\mathcal{T}_h$. We note that this approach admits strong coefficient jumps across element interfaces. The resulting discrete problem to be solved is then a linear system of equations

$$A_h\mathbf{u}_h = \mathbf{f}_h, \tag{2.3}$$

where A_h, $\mathbf{u}_h$, and $\mathbf{f}_h$ denote the global stiffness matrix, the vector of unknown degrees of freedom, and the global right hand side, respectively. Here h indicates the discretization parameter (mesh size) for the underlying partition $\mathcal{T}_h$ of Ω. The aim of this paper is to investigate scalable parallel preconditioners for solving (2.3).

2.1. Rotated trilinear nonconforming FEM

Nonconforming finite elements based on rotated multilinear shape functions were introduced by Rannacher and Turek [9] as a class of simple elements for the Stokes problem. More generally, the recent activities in the development of efficient solution methods for nonconforming finite element systems are inspired by their attractive properties as a stable discretization tool for ill-conditioned problems. The cube $[-1, 1]^3$ (see Fig. 1) is used as a reference element $\hat{e}$ to define the isoparametric rotated trilinear element $e \in \mathcal{T}_h$.

Let $\Psi_e : \hat{e} \to e$ be the corresponding trilinear one-to-one transformation, and let the nodal basis functions be determined by the relation

$$\{\phi_i\}_{i=1}^6 = \{\hat{\phi}_i \circ \Psi_e^{-1}\}_{i=1}^6, \qquad \{\hat{\phi}_i\} \in \operatorname{span}\{1, x, y, z, y^2 - x^2, x^2 - z^2\},$$

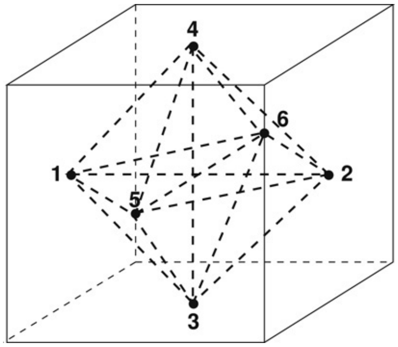

Fig. 1. Node numbering of the reference rotated trilinear hexahedral element $\hat{e}$ and connectivity pattern of the related element stiffness matrix A_e.

where $(x_1, x_2, x_3) \equiv (x, y, z)$, '$\circ$' denotes the composition of functions $\hat{\phi}_i$ and Ψ_e^{-1}. For the variant *MP* (mid point), $\{\hat{\phi}_i\}_{i=1}^6$ are defined by the point-wise interpolation condition

$$\hat{\phi}_i(b_\Gamma^j) = \delta_{ij},$$

where b_Γ^j, $j = 1, 6$ are the centers of the faces of the cube $\hat{e}$. So,

$$\hat{\phi}_1(x,y,z) = \left(1 - 3x + 2x^2 - y^2 - z^2\right)/6, \qquad \hat{\phi}_2(x,y,z) = \left(1 + 3x + 2x^2 - y^2 - z^2\right)/6,$$
$$\hat{\phi}_3(x,y,z) = \left(1 - x^2 - 3y + 2y^2 - z^2\right)/6, \qquad \hat{\phi}_4(x,y,z) = \left(1 - x^2 + 3y + 2y^2 - z^2\right)/6,$$
$$\hat{\phi}_5(x,y,z) = \left(1 - x^2 - y^2 - 3z + 2z^2\right)/6, \qquad \hat{\phi}_6(x,y,z) = \left(1 - x^2 - y^2 + 3z + 2z^2\right)/6.$$

Alternatively, for the variant MV, the face average interpolation operator is applied in the form

$$|\Gamma_{\hat{e}}^j|^{-1} \int_{\Gamma_{\hat{e}}^j} \hat{\phi}_i \mathrm{d}\Gamma_{\hat{e}}^j = \delta_{ij},$$

where $\Gamma_{\hat{e}}^j$, $j = 1, 6$ are faces of the cube $\hat{e}$, leading to

$$\hat{\phi}_1(x,y,z) = \left(2 - 6x + 6x^2 - 3y^2 - 3z^2\right)/12, \qquad \hat{\phi}_2(x,y,z) = \left(2 + 6x + 6x^2 - 3y^2 - 3z^2\right)/12,$$
$$\hat{\phi}_3(x,y,z) = \left(2 - 3x^2 - 6y + 6y^2 - 3z^2\right)/12, \qquad \hat{\phi}_4(x,y,z) = \left(2 - 3x^2 + 6y + 6y^2 - 3z^2\right)/12,$$
$$\hat{\phi}_5(x,y,z) = \left(2 - 3x^2 - 3y^2 - 6z + 6z^2\right)/12, \qquad \hat{\phi}_6(x,y,z) = \left(2 - 3x^2 - 3y^2 + 6z + 6z^2\right)/12.$$

2.2. *MIC(0) preconditioning*

In this section, we recall some facts about the modified incomplete factorization, see [10,11], or [12] for an alternative approach. Let us decompose the real $N \times N$ matrix $A = (a_{ij})$ into the form

$$A = D - L - L^T,$$

where D is the diagonal and $(-L)$ is the strictly lower triangular part of A. Then we consider the approximate factorization of A which has the following form:

$$C_{MIC(0)} = (X - L)X^{-1}(X - L)^T,$$

where $X = \mathrm{diag}(x_1, \dots, x_N)$ is a diagonal matrix determined such that A and $C_{MIC(0)}$ have equal row sums. For the purpose of preconditioning, we restrict ourselves to the case when $X > 0$, i.e., when $C_{MIC(0)}$ is positive definite. In this case, the *MIC*(0) factorization is called *stable*. Concerning the stability of the *MIC*(0) factorization, we have the following theorem [10].

Theorem 2.1. *Let $A = (a_{ij})$ be a symmetric real $N \times N$ matrix and let $A = D - L - L^T$ be the splitting of A. Let us assume that (in an elementwise sense)*

$$\begin{aligned} &L \geq 0,\\ &A\mathbf{e} \geq 0,\\ &A\mathbf{e} + L^T\mathbf{e} > 0, \quad \mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^N, \end{aligned}$$

i.e., that A is a weakly diagonally dominant matrix with nonpositive off-diagonal entries and that $A + L^T = D - L$ is strictly diagonally dominant. Then the relation

$$x_i = a_{ii} - \sum_{k=1}^{i-1} \frac{a_{ik}}{x_k} \sum_{j=k+1}^{N} a_{kj} > 0 \tag{2.4}$$

holds and the diagonal matrix $X = \mathrm{diag}(x_1, \ldots, x_N)$ defines a stable MIC(0) factorization of A.

Remark 2.2. The numerical tests presented in the last section are performed using the perturbed version of *MIC*(0) algorithm, where the incomplete factorization is applied to the matrix $\tilde{A} = A + \tilde{D}$. The diagonal perturbation $\tilde{D} = \tilde{D}(\xi) = \mathrm{diag}(\tilde{d}_1, \ldots \tilde{d}_N)$ is defined as follows:

$$\tilde{d}_i = \begin{cases} \xi a_{ii}, & \text{if } a_{ii} \geq 2w_i, \\ \xi^{1/2} a_{ii}, & \text{if } a_{ii} < 2w_i, \end{cases}$$

where $0 < \xi < 1$ is a constant and $w_i = -\sum_{j>i} a_{ij}$.

It is well known that the number of iteration steps the Preconditioned Conjugate Gradient (*PCG*) method takes until convergence is $k = O(\kappa(C^{-1}A)^{1/2})$ [13]. For the case when $C = C_{MIC(0)}(A)$ is used with second order model elliptic problems, Gustafsson [14] and Blaheta [10] give the estimate

$$\kappa(C^{-1}A) = \kappa(A)^{1/2}. \tag{2.5}$$

Many numerical experiments indicate that (2.5) holds for a wider range of problems. For the problem (1.1) we have $\kappa(A) = O(n^2) = O(N^{2/3})$ and thus, we expect that the iteration count $k = O(N^{1/6})$.

2.3. Parallel preconditioning strategies

The standard *MIC*(0) preconditioning is inherently sequential due to the recursive solution of the involved triangular systems. The idea of our approach is to apply *MIC*(0) factorization to a modified sparse matrix B, the special block structure of which allows for a scalable parallel implementation. This is based on the experience in developing *MIC*(0) algorithms for 2D problems (see [5,6,2,3]). Following the standard FEM assembling procedure, we write A in the form

$$A = \sum_{e \in \omega_h} T_e^T A_e T_e, \tag{2.6}$$

where A_e is the element stiffness matrix, T_e stands for the restriction mapping of the global vector of unknowns to the local one corresponding to the current element e. The matrix A_e is dense and can be written in the form

$$A_e = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix},$$

where the node numbering and connectivity pattern is displayed in Fig. 1.

Lemma 2.1. *Let us introduce the auxiliary global matrix B in the form*

$$B = \sum_{e \in \omega_h} \lambda_e^{(1)} T_e^T B_e T_e, \tag{2.7}$$

where B_e is a symmetric positive semidefinite matrix with nonpositive off-diagonal entries such that $B_e \mathbf{e} = A_e \mathbf{e}$, $\mathbf{e}^T = (1, 1, 1, 1, 1, 1)$, and where $\{\lambda_e^{(i)}\}_{i=1}^5$ are the nontrivial eigenvalues of $B_e^{-1} A_e$ ordered to be monotonically increasing. Then

(i) *the matrix B satisfies the conditions of Theorem 2.1 for a stable MIC(0) factorization, and*
(ii) *the local condition number analysis is applicable,*

$$\kappa(B^{-1}A) \leq \max_e \kappa(B_e^{-1} A_e).$$

Proof. The auxiliary matrix B is constructed as an M-matrix. Then, the proof of condition (i) follows straightforwardly, subject to a standard lexicographic global node numbering.

For the second statement, let $\mathbf{v} \in \mathbb{R}^N$, N is the size of the global matrix, and let $\mathbf{v}_e \in \mathbb{R}^6$ be the restriction of $\mathbf{v}$ on the current element $e \in \mathcal{T}_h$. Then,

$$(B\mathbf{v}, \mathbf{v}) = \sum_e \lambda_e^{(1)} (B_e \mathbf{v}_e, \mathbf{v}_e) \leq \sum_e (A_e \mathbf{v}_e, \mathbf{v}_e) = (A\mathbf{v}, \mathbf{v}),$$

and therefore the minimal eigenvalue λ_m is bounded below by

$$\lambda_m(B^{-1}A) \geq 1.$$

Similarly,

$$(B\mathbf{v}, \mathbf{v}) = \sum_e \lambda_e^{(1)} (B_e \mathbf{v}_e, \mathbf{v}_e) \geq \sum_e \frac{\lambda_e^{(1)}}{\lambda_e^{(5)}} (A_e \mathbf{v}_e, \mathbf{v}_e) \geq \min_e \frac{\lambda_e^{(1)}}{\lambda_e^{(5)}} \sum_e (A_e \mathbf{v}_e, \mathbf{v}_e) = \min_e \frac{\lambda_e^{(1)}}{\lambda_e^{(5)}} (A\mathbf{v}, \mathbf{v}),$$

and thus, for the maximal eigenvalue of $B^{-1}A$ we have

$$\lambda_M(B^{-1}A) \leq \max_e \frac{\lambda_e^{(5)}}{\lambda_e^{(1)}},$$

which completes the proof. □

Now we will introduce the structure of two variants for local approximations B_e under the assumption that the conditions of Lemma 2.1 are satisfied. They will later be referred to as Variant $B1$ and Variant $B2$.

Variant B1 *Variant B2*

$$B_e = \begin{bmatrix} b_{11} & & b_{13} & b_{14} & b_{15} & b_{16} \\ & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & & b_{35} & b_{36} \\ b_{41} & b_{42} & & b_{44} & b_{45} & b_{46} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & \\ b_{61} & b_{62} & b_{63} & b_{64} & & b_{66} \end{bmatrix} \qquad B_e = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & & & \\ b_{41} & b_{42} & & b_{44} & & \\ b_{51} & b_{52} & & & b_{55} & \\ b_{61} & b_{62} & & & & b_{66} \end{bmatrix}.$$

The definitions of the introduced sparse approximations B_e correspond to the local numbering shown in Fig. 2. Here the dashed lines represent the connectivity pattern of Variant $B1$ (left) and Variant $B2$ (right). Notice that the condition $B_e \mathbf{e} = A_e \mathbf{e}$ prevents the selection of a diagonal B_e.

At this point we introduce the preconditioner $\mathcal{C}$ of A which is defined as the $MIC(0)$ factorization of B; that is,

$$\mathcal{C} = \mathcal{C}_{MIC(0)}(B).$$

Recall that the definition of B ensures a stable $MIC(0)$ factorization. The locally optimized construction of the element matrices B_e for both variants $B1$ and $B2$, and for both cases of Rannacher–Turek elements, MV and MP, will be studied in the next section.

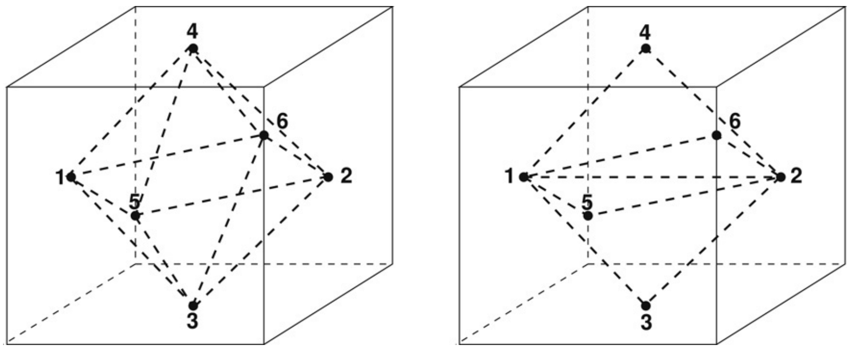

Fig. 2. Node numbering of a rotated trilinear hexahedral element and connectivity pattern of B_e.

The sparsity patterns of the B_e will ease the parallelization of the $MIC(0)$ preconditioned PCG considerably. The auxiliary matrices B will have a special block structure where the diagonal blocks are diagonal matrices. These blocks correspond to nodal lines and plains for variants $B1$ and $B2$, respectively.

3. Locally optimized approximation of the stiffness matrix

Let us consider the finite element problem (2.2) in the case of mesh isotropy. Then, the related element stiffness matrices for the Rannacher–Turek elements MP and MV read as follows,

$$A_e^{MP} = \frac{2ha_e}{9}\begin{bmatrix} 17 & -1 & -4 & -4 & -4 & -4 \\ -1 & 17 & -4 & -4 & -4 & -4 \\ -4 & -4 & 17 & -1 & -4 & -4 \\ -4 & -4 & -1 & 17 & -4 & -4 \\ -4 & -4 & -4 & -4 & 17 & -1 \\ -4 & -4 & -4 & -4 & -1 & 17 \end{bmatrix}, \qquad A_e^{MV} = 2ha_e \begin{bmatrix} 3 & 1 & -1 & -1 & -1 & -1 \\ 1 & 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & 3 & 1 & -1 & -1 \\ -1 & -1 & 1 & 3 & -1 & -1 \\ -1 & -1 & -1 & -1 & 3 & 1 \\ -1 & -1 & -1 & -1 & 1 & 3 \end{bmatrix}. \tag{3.1}$$

The problem we state in this section is to construct element matrices which satisfy the assumptions of Lemma 2.1 such that the relative condition numbers

$$\kappa\left(\left(B_e^{MP}\right)^{-1} A_e^{MP}\right) \quad \text{and} \quad \kappa\left(\left(B_e^{MV}\right)^{-1} A_e^{MV}\right)$$

are minimal for both variants $B1$ and $B2$.

3.1. Theoretical background

The following two simple lemmas will be used in our analysis.

Lemma 3.1. *Let A and B be symmetric and positive semidefinite matrices such that* $\ker(A) = \ker(B)$*, and let P be a symmetric permutation matrix, i.e., $P^T P = P^2 = I$, such that $PAP = A$. Let us assume that B has a minimal relative condition number $\kappa(B^{-1}A) = \kappa_m$ in the class of matrices with nonpositive off-diagonal elements. Then, there exists a matrix $\tilde{B}$ from the same class, such that $P\tilde{B}P = \tilde{B}$ and $\kappa(\tilde{B}^{-1}A) = \kappa_m$.*

Proof. If $A\mathbf{v} = \lambda B\mathbf{v}$, then assumption $PAP = A$ implies that $A\hat{\mathbf{v}} = \lambda(PBP)\hat{\mathbf{v}}$, $\hat{\mathbf{v}} = P\mathbf{v}$. Let us denote by λ_m and λ_M respectively the minimal and maximal eigenvalues of the generalized eigenproblem

$$A\mathbf{v} = \lambda B\mathbf{v}, \quad \mathbf{v} \notin \ker(A).$$

Then the following relations hold,

$$\lambda_m B \le A \le \lambda_M B,$$
$$\lambda_m PBP \le A \le \lambda_M PBP,$$
$$\lambda_m \tilde{B} \le A \le \lambda_M \tilde{B},$$

where

$$\tilde{B} = \frac{B + PBP}{2}.$$

Here the inequalities are understood in the sense of positive semidefinite matrices. The last from the above inequalities implies

$$\kappa(\tilde{B}^{-1}A) = \frac{\lambda_M}{\lambda_m} = \kappa_m.$$

It is readily seen that the off-diagonal entries of $\tilde{B}$ are nonpositive. To complete the proof we note that

$$P\tilde{B}P = \frac{PBP + PPBPP}{2} = \frac{B + PBP}{2} = \tilde{B}. \quad \square$$

Lemma 3.2. *Let*

$$A_e = \begin{bmatrix} 2b_{12} + a_1 & -a_1 & -b_1 & -b_1 & -b_2 & -b_2 \\ -a_1 & 2b_{12} + a_1 & -b_1 & -b_1 & -b_2 & -b_2 \\ -b_1 & -b_1 & 2b_{13} + a_2 & -a_2 & -b_3 & -b_3 \\ -b_1 & -b_1 & -a_2 & 2b_{13} + a_2 & -b_3 & -b_3 \\ -b_2 & -b_2 & -b_3 & -b_3 & 2b_{23} + a_3 & -a_3 \\ -b_2 & -b_2 & -b_3 & -b_3 & -a_3 & 2b_{23} + a_3 \end{bmatrix} \tag{3.2}$$

with $b_{ij} = b_i + b_j$. Then

$$A_e V_e = V_e L_e, \tag{3.3}$$

with

$$V_e = \begin{bmatrix} 1 & 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 2 & -1 & 0 & 0 \\ 1 & \sqrt{3} & -1 & 0 & 1 & 0 \\ 1 & \sqrt{3} & -1 & 0 & -1 & 0 \\ 1 & -\sqrt{3} & -1 & 0 & 0 & 1 \\ 1 & -\sqrt{3} & -1 & 0 & 0 & -1 \end{bmatrix}$$

and

$$L_e = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_{12} + 4b_3 & \sqrt{3}(b_2 - b_1) & 0 & 0 & 0 \\ 0 & \sqrt{3}(b_2 - b_1) & 3b_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(b_{12} + a_1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(b_{13} + a_2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(b_{23} + a_3) \end{bmatrix}.$$

Proof. The proof is by verification. □

Remark 3.1. We have chosen the basis of the invariant subspace spanned by the second and third columns of V_e such that the nonzero off-diagonal elements in L_e vanish if $b_1 = b_2$. This will be the case in our applications such that L_e is diagonal and has the eigenvalues of A_e in its diagonal. In particular, by setting $b_1 = b_2 = b_3 = 4$ and $a_1 = a_2 = a_3 = 1$, we have

$$A_e^{MP} V_e = V_e\, L_e^{MP} \qquad L_e^{MP} = \frac{2ha_e}{9}\,\mathrm{diag}(0, 24, 24, 18, 18, 18). \tag{3.4}$$

and, by setting $b_1 = b_2 = b_3 = 1$ and $a_1 = a_2 = a_3 = -1$, we get

$$A_e^{MV} V_e = V_e\, L_e^{MV} \qquad L_e^{MV} = 2ha_e\,\mathrm{diag}(0, 6, 6, 2, 2, 2). \tag{3.5}$$

3.2. Model local analysis

Lemma 3.1 states that it suffices to investigate locally optimal approximations B_e preserving the isotropy of the element stiffness matrix A_e if applicable.

Variant $B1$: In this case, a straightforward application of Lemma 3.2 leads to the following

Lemma 3.3. *For Variant* $B1$*, the matrix*

$$B_e^{MP} = B_e^{MV} = \begin{bmatrix} 4b & & -b & -b & -b & -b \\ & 4b & -b & -b & -b & -b \\ -b & -b & 4b & & -b & -b \\ -b & -b & & 4b & -b & -b \\ -b & -b & -b & -b & 4b & \\ -b & -b & -b & -b & & 4b \end{bmatrix}, \quad b > 0,$$

is a locally optimal approximation of A_e^{MP} *and* A_e^{MV}*. For any positive b, we get the bounds*

$$\kappa\left(\left(B_e^{MP}\right)^{-1} A_e^{MP}\right) = 9/8, \quad \textit{and} \quad \kappa\left(\left(B_e^{MV}\right)^{-1} A_e^{MV}\right) = 2. \tag{3.6}$$

Proof. With $a_1 = a_2 = a_3 = 0$ and $b_1 = b_2 = b_3 = b$, we get

$$B_e V_e = V_e \operatorname{diag}(0, 6b, 6b, 4b, 4b, 4b).$$

Therefore, with (3.4), the eigenvalues of the stencil (A_e^{MP}, B_e) restricted to $\operatorname{span}(\mathbf{e})^\perp$, $\mathbf{e} = [1, 1, 1, 1, 1, 1]^T$, are (up to the factor $2ha_e/9$)

$$4/b, 4/b, 9/2b, 9/2b, 9/2b$$

and, by consequence, $\kappa(A_e^{MP}, B_e) = 9/8$.

Similarly, the eigenvalues of the stencil (A_e^{MV}, B_e) restricted to $\operatorname{span}(\mathbf{e})^\perp$ are (up to the factor $2ha_e$)

$$1/b, 1/b, 1/2b, 1/2b, 1/2b.$$

Thus, $\kappa(A_e^{MV}, B_e) = 2$. □

Variant $B2$: Here, we will apply Lemma 3.2, together with the isotropy arguments. The result is given by the next lemma.

Lemma 3.4. *For Variant* $B2$*, the locally optimal approximations* B_e^{MP} *and* B_e^{MV} *can be found in the form*

$$B_e = \begin{bmatrix} 4b+a & -a & -b & -b & -b & -b \\ -a & 4b+a & -b & -b & -b & -b \\ -b & -b & 2b & & & \\ -b & -b & & 2b & & \\ -b & -b & & & 2b & \\ -b & -b & & & & 2b \end{bmatrix}, \quad a \geq 0, b > 0, \tag{3.7}$$

with the following uniform estimates for the condition numbers,

$$\kappa\left(\left(B_e^{MP}\right)^{-1} A_e^{MP}\right) \leq 3, \quad \textit{and} \quad \kappa\left(\left(B_e^{MV}\right)^{-1} A_e^{MV}\right) \leq 6. \tag{3.8}$$

Proof. For the version MP the approximate matrix B_e corresponds to the matrix in (3.3) with $b_1 = b_2 = b$, $a_1 = a$, and $b_3 = a_2 = a_3 = 0$. The values b and a shall be determined such that the condition number of $B_e^{-1} A_e^{MP}$ is minimized. We have

$$B_e V_e = V_e \operatorname{diag}(0, 2b, 6b, 4b + 2a, 2b, 2b).$$

Thus, the eigenvalues of the stencil (A_e^{MP}, B_e) restricted to span$(\mathbf{e})^{\perp}$ are (up to the factor $2ha_e/9$)

$$\frac{12}{b}, \frac{9}{a+2b}, \frac{4}{b}, \frac{9}{b}, \frac{9}{b}.$$

Hence,

$$\kappa(B_e^{-1}A_e) = \max\left(3, \frac{4(a+2b)}{3b}\right) = \max\left(3, \frac{8}{3} + \frac{4a}{3b}\right).$$

In particular, if we set $a = 1$ and $b = 4$ as in (3.1) then the condition number equals 3. Choosing $a = 0$ is also possible. This does not change the condition number but the nonzero pattern of the global matrix B. This completes the case *MP*.

A similar consideration is applied to the *MV* version of the Rannacher–Turek element. The eigenvalues of the stencil (A_e^{MV}, B_e) restricted to span$(\mathbf{e})^{\perp}$ are (up to the factor $2ha_e$)

$$\frac{3}{b}, \frac{1}{b}, \frac{1}{2b+a}, \frac{1}{b}, \frac{1}{b}.$$

Thus,

$$\kappa(B_e^{-1}A_e^{MV}) = \max\left(3, \frac{3(a+2b)}{b}\right) = 6 + \frac{3a}{b}.$$

Setting $a = 0$, we get the minimal $\kappa(B_e^{-1}A_e^{MV}) = 6$ for any positive b, in particular $b = 1$ as in (3.1) which completes the proof of the lemma. □

Now, we assume that the global matrices B^{MP} and B^{MV} are defined by (2.7), and applying Lemmas 2.1, 3.3 and 3.4 we get the global condition number estimates

$$\kappa\left(\left(B^{MP}\right)^{-1}A\right) \le 3, \qquad \kappa\left(\left(B^{MV}\right)^{-1}A\right) \le 6,$$

for both variants $B1$ and $B2$. These bounds are uniform with respect to the mesh size h as well as to possible jumps of the coefficients $a(e)$.

4. Parallel *MIC*(0) algorithms

4.1. Description of the algorithms

The *PCG* algorithm is used for the solution of the linear system (2.3). The *MIC*(0) factorization of B is chosen as the preconditioner. Let us assume that the rectangular domain Ω is decomposed into $n_1 \times n_2 \times n_3$ nonconforming hexahedral elements. The degrees of freedom in both discretization types *MV* and *MP* can be associated with the midpoints of the finite element faces. The structure of B is shown in Fig. 4 for both variants $B1$ and $B2$. Lexicographic node numbering is used. See Fig. 3 for a comparison with the structure of the original matrix A. Notice that for both variants $B1$ and $B2$, the diagonal blocks of the matrix B are diagonal. For the Variant $B1$ each diagonal block corresponds to a line of nodes; for the Variant $B2$ each such block corresponds to a plane of nodes. The sizes of these blocks vary. They are n_3 or $n_3 + 1$ for Variant $B1$ and n_2n_3 or $n_2(n_3 + 1) + (n_2 + 1)n_3$ for Variant $B2$.

To handle the system with the preconditioner

$$C_{MIC(0)}(B)\mathbf{w} \equiv (X - L)\,X^{-1}\left(X - L^T\right)\mathbf{w} = \mathbf{v} \tag{4.1}$$

one has to solve systems $\tilde{L}\mathbf{y} \equiv (X - L)\mathbf{y} = \mathbf{v}$, $X^{-1}\mathbf{z} = \mathbf{y}$ and $\tilde{L}^T\mathbf{w} = \mathbf{z}$, where L is the strictly lower triangular part of the matrix B. The triangular systems are solved using standard forward or backward recurrences. This can be done in $k_{B1} = n_1(2n_2 + 1) + n_2$ and $k_{B2} = 2n_1 + 1$ stages for variants $B1$ and $B2$, respectively. Within stage i the block $\mathbf{y}_i$ is computed. Since the blocks $\tilde{L}_{ii}$, are diagonal, the computations of each component of $\mathbf{y}_i$ can be performed in parallel. Let the $p \le n_3 + 1$ processors be denoted by $P_1, P_2, \dots, P_p$. We distribute the entries of the vectors corresponding to each diagonal block of B among all processors. This can be done as shown in Fig. 5 for both variants $B1$ and $B2$. A 2D projection on each "layer" of finite elements is used. Since the "layers" have common interfaces, the nodes that belong

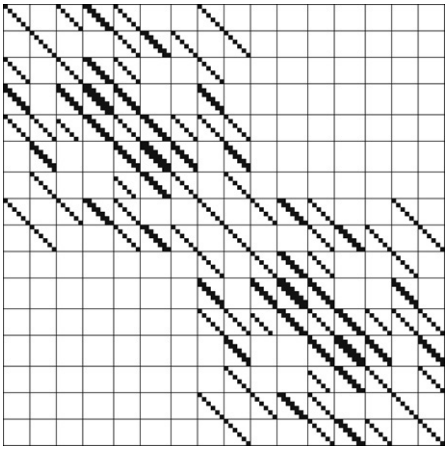

Fig. 3. Sparsity pattern of the original matrix A for a division of Ω into $2 \times 2 \times 6$ hexahedrons. Non-zero elements are indicated by small squares.

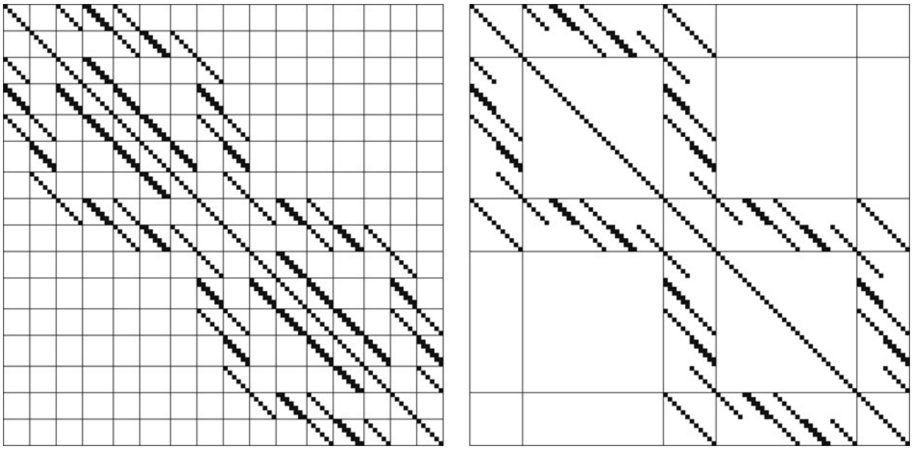

Fig. 4. Sparsity pattern of the matrix B for a division of Ω into $2 \times 2 \times 6$ hexahedrons. Variant $B1$ on the left and Variant $B2$ on the right. Non-zero elements are indicated by small squares.

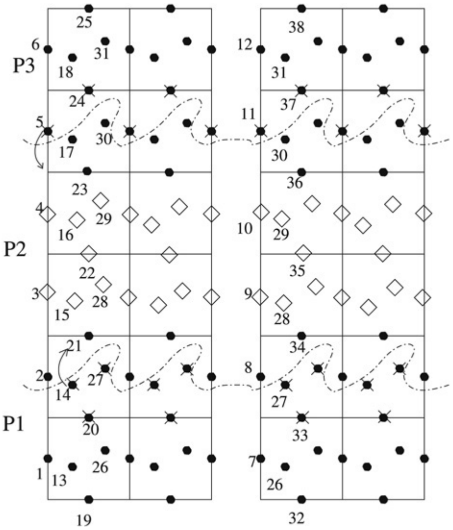

Fig. 5. Data distribution: $p = 3, n_1 = 2, n_2 = 2, n_3 = 6$. Communication scheme for the matrix–vector multiplication.

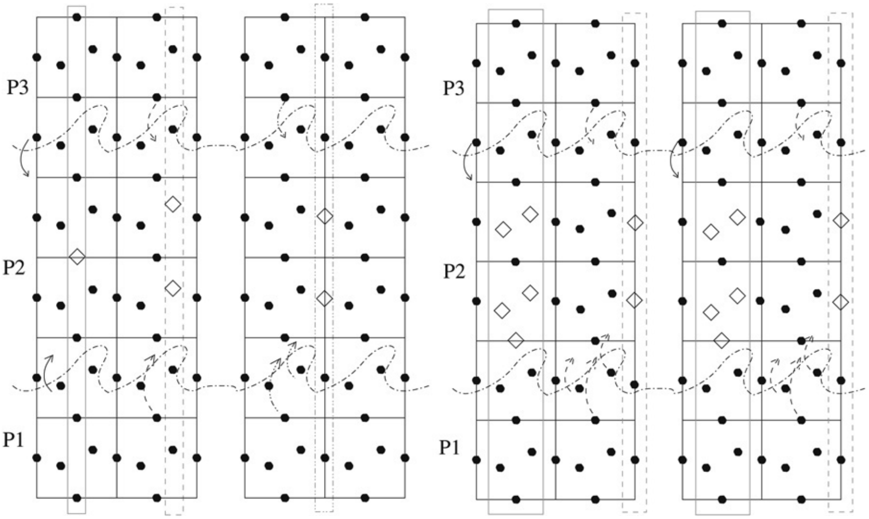

Fig. 6. Communication scheme for the solution of lower triangular systems in the preconditioner. Variant $B1$ on the left and Variant $B2$ on the right.

to those sides appear twice (once on each 2D projection). Each processor P_j receives a strip of the computational domain. These strips have almost equal size. Elements of all vectors and rows of all matrices that participate in the *PCG* algorithm are distributed in the same manner. The processor P_j takes care of the local computations on the j-th strip.

Further, we briefly describe how the operations in the *PCG* algorithm are performed, and what kind of communications are required.

In each *PCG* iteration, besides solving a system with the preconditioning matrix $C_{MIC(0)}(B)$, one matrix vector multiplication with the original matrix A, two inner products, and three linked vector triads of the form $\mathbf{v} := \alpha\mathbf{v} + \mathbf{u}$ have to be executed. The number of operations per iteration for the *PCG* algorithm is $\mathcal{N}_{it}^{PCG} \approx 27N$ for the Variant $B1$, and $\mathcal{N}_{it}^{PCG} \approx 25N$ for the Variant $B2$. Here, $N = 3n_1n_2n_3+n_1n_2+n_1n_3+n_2n_3$ is the total number of unknowns.

For the triads, each processor calculates its part of the vector $\mathbf{v}$. No communication is required. After computing the inner products corresponding to their parts of the vectors, the processors have to perform a global reduction operation to sum up the final result.

To obtain the components of the matrix–vector multiplication $A\mathbf{v}$ for which the processor P_i is responsible, it needs to receive from the processors P_{i-1} and P_{i+1} some components of the vector $\mathbf{v}$. The number of these components is $4n_1n_2 + n_1 + n_2$. Because of the even distribution of nodes that lie on the splitting planes between the strips, half of that number is to be received from each of the processors P_{i-1} and P_{i+1}. On Fig. 5, the elements to be transferred to P_2 are marked with the sign ×. While these communications are in progress, the components of $A\mathbf{v}$ which do not depend on the components of $\mathbf{v}$ in the neighbouring processors can be computed. These components are marked by the sign $\diamond$.

Let us return to the solving with the preconditioner (4.1). The solution of a system with a diagonal matrix is trivial and does not require any communication. As we mentioned, the solution of the triangular systems can be done in k_{B1} or k_{B2} stages. During each stage, the part of the solution corresponding to one vertical line of nodes for Variant $B1$ and corresponding to the related plane of nodes for Variant $B2$ is computed. Before each stages, the processors have to exchange some components in order the computations to be performed. Three distinct patterns of transfer are required for Variant $B1$ and two for Variant $B2$. For the case with a lower triangular matrix they are depicted in Fig. 6. The rectangles dashed with the same pattern bound the elements to be computed on a single stage. The transfers shown with the arrows in that pattern are to be performed prior the computations. For Variant $B1$ a transfer of one or two components between each pair of nodes in both directions is required per each vertical line. For Variant $B2$ transfers of either n_2 or $3n_2 + 2$ elements are needed between the stages. Again, computations for the inner components, marked with sign $\diamond$, can be overlapped with the communications.

4.2. Parallel execution times

Estimations of the parallel execution times are derived with the following assumptions: (a) executing M arithmetical operations on one processor lasts $T = Mt_a$, where t_a is the average unit time to perform one arithmetical operation on a single processor, (b) the time to transfer M data items between two neighbouring processors can be approximated by $T^{\text{comm}} = t_s + Mt_c$, where t_s is the startup time and t_c is the incremental time for each of the M elements to be transferred, and (c) send and receive operations between each pair of neighbouring processors can be done in parallel. We get the following expressions for communication times:

$$\begin{aligned}
&T^{\text{comm}}(A\mathbf{v}) \approx t_s + 2n_1n_2t_c,\\
&T^{\text{comm}}(C_{B1}^{-1}\mathbf{v}) \approx 6n_1n_2t_s + 8n_1n_2t_c,\\
&T^{\text{comm}}(C_{B2}^{-1}\mathbf{v}) \approx 2n_1t_s + 8n_1n_2t_c.
\end{aligned}$$

The above communications are completely local and do not depend on the number of processors. The inner product needs one broadcasting and one gathering global communication, but they do not contribute to the leading terms of the total parallel time. The parallel properties of the algorithm do not depend on the number of iterations, so it is enough to evaluate the parallel time per iteration, and use it in the speedup and efficiency analysis. As the computations are almost equally distributed among the processors, assuming there is no overlapping of the communications and computations one can write for the total time per iteration on p processors the following estimates:

Variant $B1$:

$$\begin{aligned}
T_p^{it} &\approx \frac{27Nt_a}{p} + T_{B1}^{\text{comm}}(C^{-1}\mathbf{v}) + T^{\text{comm}}(A\mathbf{v})\\
&= \frac{27Nt_a}{p} + (6n_1n_2 + 1)t_s + 10n_1n_2t_c
\end{aligned}$$

Variant $B2$:

$$\begin{aligned}
T_p^{it} &\approx \frac{25Nt_a}{p} + T_{B2}^{\text{comm}}(C^{-1}\mathbf{v}) + T^{\text{comm}}(A\mathbf{v})\\
&= \frac{25Nt_a}{p} + (2n_1 + 1)t_s + 10n_1n_2t_c.
\end{aligned}$$

The relative speedups $S_p = T_1/T_p$ are thus

Variant $B1$:

$$S_p \approx \frac{27Nt_a}{\frac{27Nt_a}{p} + (6n_1n_2 + 1)t_s + 10n_1n_2t_c} \approx \frac{p}{1 + \frac{2t_sp}{27n_3t_a} + \frac{10t_cp}{81n_3t_a}},$$

and Variant $B2$:

$$S_p \approx \frac{25Nt_a}{\frac{25Nt_a}{p} + (2n_1 + 1)t_s + 10n_1n_2t_c} \approx \frac{p}{1 + \frac{2t_sp}{75n_2n_3t_a} + \frac{2t_cp}{15n_3t_a}},$$

respectively. Here $N \approx 3n_1n_2n_3$ is used. The speedup, and therefore the efficiency $E_p = S_p/p$, will grow with n_3 in both variants up to their theoretical limits $S_p = p$ and $E_p = 1$. Since on a real computer $t_s \gg t_c$ and $t_s \gg t_a$, we can expect good efficiencies only when $n_3 \gg p\,t_s/t_a$. The speedup of Variant $B2$ is expected to be much better than the speedup of Variant $B1$, because about $3n_2$ times fewer messages are sent.

Generally, when computations and communications overlap, parallel execution times decrease. Overlapping computation and communication can notably reduce the influence of high network latencies and low bandwidths. Of course this requires a large amount of computation per communication.

In Variant $B2$, about $3n_2$ times fewer messages are sent than in Variant $B1$. So, during the application of the preconditioner, n_2 times more arithmetic operations can be overlapped with communication, thus vastly improving the parallel performance.

Table 1
Variant $B1$

p	$\frac{n}{\text{Iter}}$	C1			C2			C3		
		T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1	$\frac{31}{22}$	1.08			1.28			0.70		
2		0.94	1.15	0.57	0.93	1.38	0.69	0.65	1.07	0.54
4		0.91	1.19	0.30	0.83	1.55	0.39	0.45	1.57	0.39
8		1.14	0.95	0.12	1.12	1.15	0.14	0.37	1.90	0.24
16		1.28	0.84	0.05	1.11	1.15	0.07	0.37	1.89	0.11
1	$\frac{63}{31}$	9.87			12.90			7.07		
2		6.98	1.41	0.71	8.56	1.51	0.75	4.44	1.59	0.79
4		5.91	1.67	0.42	6.39	2.02	0.50	3.54	2.00	0.50
8		5.46	1.81	0.23	7.12	1.81	0.23	2.77	2.55	0.32
16		5.83	1.69	0.11	11.9	1.08	0.07	2.40	2.94	0.18
1	$\frac{127}{44}$	95.3			134.2			78.1		
2		63.2	1.51	0.75	84.2	1.59	0.80	43.4	1.80	0.90
4		48.5	1.96	0.49	54.2	2.47	0.62	29.2	2.67	0.66
8		36.9	2.59	0.32	45.0	2.98	0.37	20.3	3.84	0.48
16		34.2	2.79	0.17	74.8	1.79	0.11	16.2	4.83	0.30
1	$\frac{255}{64}$	1147			1474[a]			859[a]		
2		659	1.74	0.87						
4		361	3.18	0.79	551	2.67[a]	0.67[a]	277	3.10[a]	0.78[a]
8		287	4.00	0.50	432	3.41[a]	0.42[a]	173	4.97[a]	0.62[a]
16		264	4.34	0.27	397	3.71[a]	0.23[a]	123	6.98[a]	0.43[a]

[a] Estimated.

5. Parallel numerical tests

The variants $B1$ and $B2$ for parallel preconditioning were implemented in C++ using the *MP*I message passing library. The communications and the computations were overlapped wherever possible.

We present parallel execution times, speedups and efficiencies from experiments performed on three parallel computing platforms, referred to below as C1, C2 and C3. Platform C1 is an "IBM SP Cluster 1600" consisting of 64 p5-575 nodes interconnected with a pair of connections to the Federation HPS (High Performance Switch). Each p5-575 node contains 8 Power5 S*MP* processors at 1.9 GHz and 16 GB of RAM. The network bandwidth is 16 Gb/s. Platform C2 is an IBM Linux Cluster 1350, made of 512 dual-core IBM X335 nodes. Each node contains 2 Xeon Pentium IV processors and 2 GB of RAM. Nodes are interconnected with a 1Gb Myrinet network. Platform C3 is a "Cray XD1" cabinet, fully equipped with 72 2-way nodes, totalling 144 AMD Opteron processors at 2.4 GHz. Each node has 4 GB of memory. The CPUs are interconnected with the Cray RaidArray network with a bandwidth of 5.6 Gb/s. The key feature of C3 is the extremely small network latency. The processors on C1 are capable of executing 4 double precision floating point operations per clock cycle, while those of C2 and C3 are capable only of 2.

We consider the model Poisson equation in a unit cube with homogeneous Dirichlet boundary conditions on the right side of the domain. The partitioning is uniform; let $n_1 = n_2 = n_3 = n$. The size of the discrete problem is $N = 3(n^3 + n^2)$. A relative stopping criterion,

$$\frac{(C^{-1}\mathbf{r}^i, \mathbf{r}^i)}{(C^{-1}\mathbf{r}^0, \mathbf{r}^0)} < 10^{-9},$$

is used in the *PCG* algorithm, where $\mathbf{r}^i$ stands for the residual at the i-th iteration step. Since the parallel properties of the algorithm do not depend on the discretization type, results from runs of only the Variant *MP* are presented. In Table 1, the mesh size parameter n, the iteration count *Iter*, execution times T_p in seconds, speedups S_p and efficiencies E_p are presented for the Variant $B1$ on platforms C1, C2 and C3. Here p is the number of processors. The corresponding quantities for Variant $B2$ are collected in Table 2.

Tests with $n = 255$ could not be run on platforms C2 and C3 with $p = 1$ and $p = 2$ due to lack of main memory available on a single node. For the same reason, experiments on C2 with $n = 255$ were run on 4 nodes, utilizing only one CPU per computing node. The presented sequential execution times are based on a simple extrapolation. They are

Table 2
Variant $B2$

p	$\frac{n}{\text{Iter}}$	C1			C2			C3		
		T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1	$\frac{31}{22}$	0.60			1.00			0.65		
2		0.30	2.00	1.00	0.65	1.54	0.77	0.39	1.68	0.84
4		0.20	3.05	0.76	0.44	2.27	0.57	0.27	2.37	0.59
8		0.17	3.58	0.45	0.31	3.29	0.41	0.16	4.02	0.50
16		0.08	7.09	0.44	0.27	3.65	0.23	0.11	5.78	0.36
1	$\frac{63}{31}$	6.03			11.14			7.48		
2		3.09	1.95	0.97	6.67	1.67	0.83	3.88	1.92	0.96
4		1.68	3.58	0.89	3.97	2.80	0.70	2.35	3.18	0.79
8		0.94	6.42	0.80	2.58	4.32	0.54	1.49	5.00	0.63
16		0.57	10.57	0.66	2.55	4.36	0.27	1.09	6.87	0.42
1	$\frac{127}{44}$	74.0			127.0			95.9		
2		38.1	1.94	0.97	74.5	1.70	0.85	50.0	1.92	0.96
4		20.8	3.56	0.89	41.1	3.10	0.77	25.0	3.83	0.96
8		10.7	6.94	0.86	23.9	5.31	0.66	13.8	6.97	0.87
16		5.63	13.14	0.82	18.8	6.74	0.42	8.6	11.11	0.69
1	$\frac{255}{61}$	910			1397[a]			1055[a]		
2		458	1.99	0.99						
4		206	4.4	1.10	470	2.97[a]	0.74[a]	291	3.62[a]	0.90[a]
8		111	8.21	1.02	253	5.52[a]	0.69[a]	154	6.85[a]	0.85[a]
16		61	14.88	0.92	149	9.38[a]	0.58[a]	85	12.41[a]	0.77[a]

[a] Estimated.

used for the estimation of the parallel speedups and efficiencies. Programs for Variants $B1$ and $B2$ access the memory in a different pattern. This explains the different behaviours of sequential times, comparing Variants $B1$ and $B2$ on different machines. The tables well illustrate the different properties of the computing platforms used. The network on platform C3 has extremely small latencies, and we see that Variant $B1$ performs better on C3 than on C1, although C1 has 8 processors per node — and so no messages have to be passed if $p \leq 8$.

As expected, one can observe that the iteration count is of order $O(n^{1/2}) = O(N^{1/6})$, and the total time grows as $O(n^{7/2}) = O(N^{7/6})$. As a general rule, for a given number of processors the speedup and efficiency grow with the problem size. Conversely for fixed n, the efficiency decreases with the number of processors. This is true for all platforms and confirms our analysis.

For Variant $B1$, reasonable efficiencies are obtained only when n/p is sufficiently large. And again, as we expected, for a given p and n Variant $B2$ performs far better even for smaller ratios n/p. It is clearly seen how reducing the number of communication steps in the solution of the preconditioner improves the parallel performance. The advantages of overlapping the computations and communications are also better expressed in Variant $B2$.

The efficiencies larger than 1 could be explained by the somehow slower sequential run. Possible reasons for this slowdown might be a worse cache utilization or a heavy load on the memory subsystem caused by other processes running on the same node.

Acknowledgments

Partial support for this work through EC INCO Grant BIS-21++ 016639/2005 and Bulgarian NSF Grant I-1402/2004 is gratefully acknowledged. The development of the numerical tests was supported via EC Project HPC-EUROPA RII3-CT-2003-506079.

References

[1] D.N. Arnold, F. Brezzi, Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates, RAIRO Model. Math. Anal. Numer. 19 (1985) 7–32.

[2] I. Gustafsson, G. Lindskog, On parallel solution of linear elasticity problems: Part I: Theory, Numer. Linear Algebra Appl. 5 (2) (1998) 123–139.

[3] R.D. Lazarov, S. Margenov, On a two-level parallel *MIC*(0) preconditioning of Crouzeix–Raviart non-conforming FEM systems, in: I. Dimov, I. Lirkov, S. Margenov, Z. Zlatev (Eds.), Numerical Methods and Applications, in: LNCS, vol. 2542, Springer, 2003, pp. 192–201.

[4] G. Bencheva, I. Georgiev, S. Margenov, Two-level preconditioning of Crouzeix–Raviart anisotropic FEM systems, in: Lecture Notes in Computer Science, vol. 2907, Springer, 2004, pp. 76–84.

[5] G. Bencheva, S. Margenov, Parallel incomplete factorization preconditioning of rotated linear FEM systems, J. Comp. Appl. Mech. 4 (2) (2003) 105–117.

[6] G. Bencheva, S. Margenov, Performance analysis of parallel *MIC*(0) preconditioning of rotated bilinear nonconforming FEM systems, Math. Balkanica 18 (2004) 319–336.

[7] P. Arbenz, S. Margenov, Parallel *MIC*(0) preconditioning of 3D nonconforming FEM systems, in: Iterative Methods, Preconditioning and Numerical PDEs (Proceedings of IMET Conference, Prague), 2004, pp. 12–15.

[8] Y. Vutov, Parallel incomplete factorization of 3D NC FEM elliptic systems, in: LNCS, vol. 4310, Springer, 2007, pp. 114–121.

[9] R. Rannacher, S. Turek, Simple nonconforming quadrilateral stokes element, Numer. Methods Partial Differential Equations 8 (2) (1992) 97–112.

[10] R. Blaheta, Displacement decomposition–incomplete factorization preconditioning techniques for linear elasticity problems, Numer. Linear Algebra Appl. 1 (1994) 107–126.

[11] I. Gustafsson, Modified incomplete Cholesky (MIC) factorization, in: D.J. Evans (Ed.), Preconditioning Methods; Theory and Applications, Gordon and Breach, 1983, pp. 265–293.

[12] P. Saint-George, G. Warzee, R. Beauwens, Y. Notay, High performance *PCG* solvers for FEM structural analyses, Internat. J. Numer. Meth. Engrg. 39 (1996) 1313–1340.

[13] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.

[14] I. Gustafsson, Stability and rate of convergence of modified incomplete Cholesky factorization methods, Chalmers University of Technology, Report Nr. 79.02R, 1979.