Journal of Computational and Applied Mechanics, Vol. 4., No. 2., (2003), pp. 105–117

PARALLEL INCOMPLETE FACTORIZATION PRECONDITIONING OF ROTATED LINEAR FEM SYSTEMS

Gergana Bencheva, Svetozar Margenov
Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bontchev str., Bl. 25A, 1113 Sofia, Bulgaria
{gery,margenov}@parallel.bas.bg

[Received: June 19, 2002]

Abstract. The recent efforts in development of efficient solution methods for non-conforming finite element systems are inspired by their importance for various applications in scientific computations and engineering. This study is focused on the implementation of rotated bilinear elements. A locally modified approximation of the global stiffness matrix is proposed allowing for: a) a stable MIC(0) factorization; and b) a scalable parallel implementation. An optimal condition number estimate is derived for the constructed sparse matrix approximation with respect to the original global stiffness matrix. The estimates of the parallel speed-up and the parallel efficiency as well as the presented parallel numerical tests demonstrate the potential of the PCG algorithm and the MPI code developed.

Mathematical Subject Classification: 65F10, 65N30
Keywords: non-conforming FEM, preconditioning, parallel algorithms

1. Introduction

Consider the elliptic equation

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

Here $\nabla u(x)$ denotes the gradient of u and $\nabla\cdot q$ denotes the divergence of the vector q. Further, we assume that Ω is a convex polygonal domain in $\mathbb{R}^2$, $f(x)$ is a given function in $L^2(\Omega)$, $a(x)=[a_{ij}(x)]_{i,j=1}^2$ is a symmetric matrix, n is the outward unit vector normal to the boundary $\Gamma=\partial\Omega$, and $\Gamma=\bar{\Gamma}_D\cup\bar{\Gamma}_N$. We assume that $a_{ij}(x)$ are piece-wise smooth functions on $\bar{\Omega}$ satisfying the uniform positive definiteness condition of the matrix $a(x)$.

Problem (1.1) can be discretized in various ways. Among the most popular and frequently used methods of approximation are the Galerkin finite element method, the finite volume method and the mixed finite element method. Each of these methods has its advantages and disadvantages when applied to particular engineering problems. For example, for petroleum reservoir problems in geometrically simple domains and

heterogeneous media, the finite volume method is known to be reliable, accurate, and locally mass conservative. Many engineering problems, e.g. petroleum recovery, ground-water contamination, seismic exploration, etc. need very accurate velocity (flux) determination in the presence of heterogeneities, anisotropy and large jumps in the coefficient matrix $a(x)$. More accurate approximation of the velocity can be achieved through the use of the mixed finite element method (see, e.g. [5]). However, the technique of the mixed finite element method leads generally to an algebraic saddle point problem that is more difficult and more expensive to solve.

An alternative approach can be taken by developing hybrid methods where the continuity of the velocity vector normal to the boundary of each element is enforced by Lagrange multipliers. The important discovery of Arnold and Brezzi [2] is that the Schur system for the Lagrange multipliers can be obtained also as a discretization of (1.1) by Galerkin method using linear nonconforming elements. Namely in [2] it is shown that the lowest-order Raviart-Thomas mixed element approximations are equivalent to the usual Crouzeix-Raviart P_1-nonconforming finite element approximations when the classical P_1-nonconforming space is augmented with P_3-bubbles. Further, such a relationship has been studied for a large variety of mixed finite element spaces [1, 6].

Our study is focused on the implementation of rotated bilinear elements. These elements are an attractive discretizing tool since they possess favorable stability properties for the Stokes and the Lamé equations. An additional important feature is the regular sparsity of the stiffness matrices with no more than seven non-zero elements per row even in the case of non-regular meshes.

Two algorithms are presented, where MP and MV stand for the variants of the nodal basis functions corresponding to mid-point and integral mid-value interpolation operators.

There are two general approaches to construct parallel preconditioners, based respectively on: a) domain decomposition, or b) block incomplete/approximate factorization. The second approach does not lead to an optimal preconditioner in terms of problem size, but produces highly parallel and efficient algorithms (see, [3, 4, 7, 8, 9, 10, 11]).

Here we first locally modify the stiffness matrix, and then apply a pointwise incomplete factorization. We get as a result a well parallelizable block structure of the preconditioner, preserving the robustness with respect to the local properties of the matrix.

2. **Finite element discretization**

The domain Ω is partitioned using quadrilaterals $e \in \omega_h$. Our analysis here is concentrated on the isotropic case, and we will assume from now that $a(x)$ is a scalar function. The partitioning ω_h is aligned with the discontinuities of the coefficient $a(x)$ so that over each element $e \in \omega_h$ the function $a(x)$ is smooth. Further, we assume that the partitioning is quasi-uniform with a characteristic mesh-size h.

The Galerkin variational formulation of the above problem reads: given $f \in L^2(\Omega)$ find a function $u \in H_D^1(\Omega) = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$, satisfying

$$\mathcal{A}(u, v) = (f, v) \qquad \forall v \in H_D^1(\Omega), \tag{2.1}$$

where

$$\mathcal{A}(u, v) = \int_\Omega a(x)\nabla u(x) \cdot \nabla v(x) dx\cdot$$

The rotated bilinear non-conforming finite elements on quadrilaterals are implemented for the numerical solution of (2.1). The finite element space V_h corresponds to ω_h. This study is concerned with a comparison analysis of two alternative constructions of V_h, where algorithms MP and MV stand for the variants of the nodal basis functions corresponding to mid-point and integral mid-value interpolation operators (see [12]). In defining the isoparametric rotated bilinear element one uses the unit square (with sides parallel to the coordinate axes) as a reference element $\hat{e}$. For each $e \in \omega_h$, let $\psi_e : \hat{e} \to e$ be the corresponding bilinear transformation. The element nodal basis functions are determined by the relations

$$\{\phi_i\}_{i=1}^4 = \{\hat{\phi}_i \circ \psi_e^{-1}\}_{i=1}^4.$$

For algorithm MP, the reference element basis functions $\{\hat{\phi}_i\}_{i=1}^4$ are determined by the standard nodal interpolation conditions

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

where $\{b_\Gamma^j\}_{j=1}^4$ are the mid-points of the sides $\{\Gamma_{\hat{e}}^j\}_{j=1}^4$ of $\hat{e}$, and then

$$\{\hat{\phi}_i\}_{i=1}^4 = \left\{\left(1 \pm 2\xi_i + \xi_i^2 - \xi_{j+1}^2\right)/4,\ j = 1, 2\right\}.$$

Alternatively, for algorithm MV, an integral mid-value interpolation operator is applied in the form

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

and then

$$\{\hat{\phi}_i\}_{i=1}^4 = \left\{\left(2 \pm 4\xi_j + 3\left(\xi_j^2 - \xi_{j+1}^2\right)\right)/16,\ j = 1, 2\right\}.$$

Then the finite element formulation is: find a function $u_h \in V_h$, satisfying

$$\mathcal{A}_h(u_h, v_h) = (f, v_h) \qquad \forall v_h \in V_h, \tag{2.2}$$

where

$$\mathcal{A}_h(u_h, v_h) = \sum_{e \in \omega_h} \int_e a(e)\nabla u_h \cdot \nabla v_h dx,$$

where $a(e)$ is defined as the averaged value

$$a(e) = \frac{1}{|e|} \int_e a(x) dx$$

over each $e \in \omega_h$. We note that we allow strong coefficient jumps through the interface boundaries between the elements. Now, the standard computational procedure leads to the linear system of equations

$$A\mathbf{u} = \mathbf{f}, \tag{2.3}$$

where A is the corresponding stiffness matrix. The matrix A is sparse, symmetric and positive definite. For large scale problems, the preconditioned conjugate gradient (PCG) method is known to be the best solution method of (2.3).

The goal of this study is to present a **robust and parallelizable** preconditioning algorithm for solving (2.3). The proposed method will be based on incomplete factorization of sparse matrices. The construction of the preconditioner is based on a MIC(0) factorization of a locally modified approximation of the original stiffness matrix. We have shown in our model analysis that the condition number corresponding to the applied local modification is independent of possible coefficient jumps (see Section 5). The proposed construction is aimed to improve the parallel features of the algorithm, which are discussed in Section 7. A set of numerical tests illustrating the robustness of the method, and the efficiency of the parallel implementation are shown in Sections 6, 8, respectively. Some concluding remarks are given in the last section.

3. MIC(0) preconditioning

We present here some background remarks about the modified incomplete Cholesky factorization MIC(0) preconditioner. Our presentation at this point follows those in [3], see also [9]. Let us rewrite the real $N \times N$ matrix $A = (a_{ij})$ in the form

$$A = D - L - L^t\,, \tag{3.1}$$

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:

$$\mathcal{C}_{MIC(0)}(A) = \mathcal{C}_{MIC(0)} = (X - L)X^{-1}(X - L)^t\,, \tag{3.2}$$

where $X = diag(x_1, \cdots, x_N)$ is a diagonal matrix determined by the condition of equal row sums:

$$\mathcal{C}_{MIC(0)}\underline{e} = A\underline{e},\ \underline{e} = (1, \cdots, 1)^t \in \mathcal{R}^N\,.$$

For the purpose of preconditioning, we are interested in the case when $X > 0$ and thus $\mathcal{C}_{MIC(0)}$ is positive definite. If this holds, we speak about *stable* $MIC(0)$ factorization. Concerning stability of $MIC(0)$ factorization, the following theorem holds.

Theorem 1. *Let $A = (a_{ij})$ be a symmetric real $N \times N$ matrix and let $A = D - L - L^t$ be the splitting (3.1) of A. Let us assume that*

$$\begin{aligned} L &\geq 0 \\ A\underline{e} &\geq 0 \\ A\underline{e} + L^t\underline{e} &> 0 \quad \underline{e} = (1, \cdots, 1)^t \in \mathcal{R}^N, \end{aligned}$$

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

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

gives the positive values x_i and the diagonal matrix $X = diag(x_1, \cdots, x_N)$ defines stable $MIC(0)$ factorization of A.

Remark 1. *The numerical tests presented in this paper 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 $\tilde{D} = \tilde{D}(\xi) = diag(\tilde{d}_1, \ldots \tilde{d}_N)$ is defined as follows:*

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

where

$$w_i = -\sum_{j>i} a_{ij}.$$

Here $0 < \xi < 1$ is a constant of the same order as the minimal eigenvalue of A. The computations for the model problems considered are done with $\xi = h^2$.

It is readily seen from (3.2) that the computational cost of one MIC(0) PCG iteration is proportional to the size of the matrix A. More precisely, the complexity $\mathcal{N}(\mathcal{C}^{-1}_{MIC(0)}\mathbf{v})$ is almost the same as $\mathcal{N}(A\mathbf{v})$. This will be discussed in some more details later. Unfortunately, the method is based on recursive computations, and therefore is inherently sequential. The idea of our algorithm is to apply MIC(0) factorization to a modified sparse matrix the special block structure of which allows for a scalable parallel implementation.

4. **The preconditioning algorithm**

The studied preconditioner $\mathcal{C}$ is constructed by a proper local modification of the stiffness matrix A. Following the standard FEM assembling procedure we write A in the form

$$A = \sum_{e \in \omega_h} L_e^T A_e L_e,$$

where A_e is the element stiffness matrix, L_e stands for the restriction mapping of the global vector of unknowns to the local one corresponding to the current quadrilateral element e. We now introduce the approximation B_e of A_e as follows

$$A_e = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad B_e = \begin{bmatrix} b_{11} & a_{12} & 0 & a_{14} \\ a_{21} & b_{22} & a_{23} & 0 \\ 0 & a_{32} & b_{33} & a_{34} \\ a_{41} & 0 & a_{43} & b_{44} \end{bmatrix}, \tag{4.1}$$

where

$$b_{11} = a_{11} + a_{13},\ b_{22} = a_{22} + a_{24},\ b_{33} = a_{33} + a_{31},\ b_{44} = a_{44} + a_{42},$$

that is A_e and B_e have equal row sums.

Assembling the locally defined matrices B_e we get the global one

$$B = \sum_{e \in \omega_h} L_e^T B_e L_e, \tag{4.2}$$

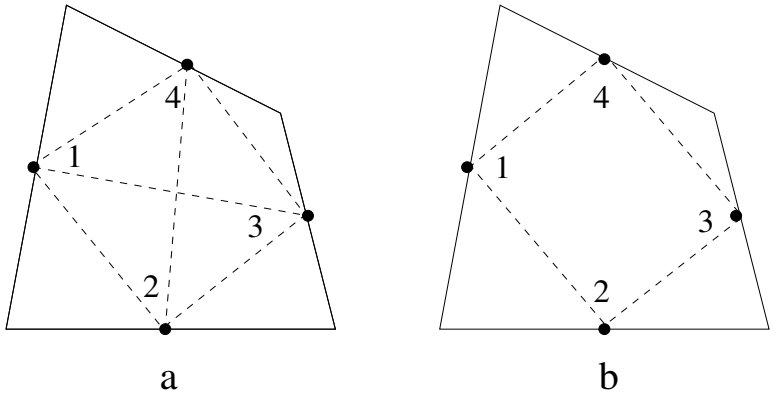

Figure 1. (a) Node numbering of a rotated bilinear quadrilateral element e; (b) Connectivity pattern of B_e.

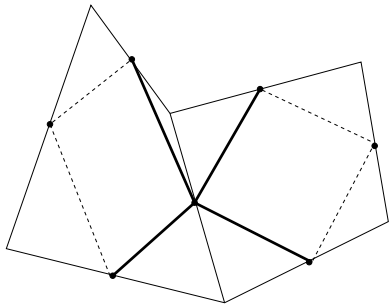

Figure 2. Skewed five point stencil

The definition of B_e corresponds to the node numbering as shown in Figure 1. Here the dash lines represent the connectivity pattern of (a) the dense element stiffness matrix A_e and (b) its locally modified sparse approximation B_e. The structure of B could be interpreted as a skewed five point stencil (see Figure 2) whereas in a very general setting A and B are spectrally equivalent.

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

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

This needs naturally B to allow for a stable MIC(0) factorization, which in particular will be discussed in the next section.

5. **Condition number model analysis**

The model problem we analyze in this section is set on a uniform square mesh. Then the element stiffness matrices corresponding to the square element $e \in \omega_h$ in the cases MP and MV have the form:

$$A_e^{MP} = \frac{a(e)}{3} \begin{bmatrix} 5 & -1 & -2 & -2 \\ -1 & 5 & -2 & -2 \\ -2 & -2 & 5 & -1 \\ -2 & -2 & -1 & 5 \end{bmatrix}, \tag{5.1}$$

$$A_e^{MV} = \frac{a(e)}{8} \begin{bmatrix} 5 & 1 & -3 & -3 \\ 1 & 5 & -3 & -3 \\ -3 & -3 & 5 & 1 \\ -3 & -3 & 1 & 5 \end{bmatrix}. \tag{5.2}$$

We consider now the local eigenvalue problem

$$A_e^{MP}\mathbf{w} = \lambda B_e^{MP}\mathbf{w}. \tag{5.3}$$

Obviously $Ker(A_e^{MP}) = Ker(B_e^{MP}) = Span\{\mathbf{e}\}$ where $\mathbf{e}^t = (1,1,1,1)$, and it is therefore enough to consider a reduced 3×3 eigenvalue problem instead of (5.3). Then the simplification using the substitution

$$\mu = 1 - \lambda$$

leads to the following characteristic equation for μ

$$det \begin{bmatrix} 1+4\mu & -1 & -2\mu \\ -1 & 1+4\mu & -2\mu \\ -2\mu & -2\mu & 1+4\mu \end{bmatrix} = 0. \tag{5.4}$$

Further computation shows that $\mu_1 = 0$ and $\mu_{2,3} = -1/2$, and therefore

$$\lambda_1 = 1, \qquad \lambda_{2,3} = 3/2.$$

The global condition number estimate directly follows from the local analysis presented. Namely, we have

$$\mathbf{v}^T A^{MP}\mathbf{v} = \sum_{e\in\omega_h} \mathbf{v}_e^T L_e^T A^{MP} L_e \mathbf{v}_e \le 3/2 \sum_{e\in\omega_h} \mathbf{v}_e^T L_e^T B_e^{MP} L_e \mathbf{v}_e = 3/2 \mathbf{v}^T B^{MP}\mathbf{v}$$

and, similarly,

$$\mathbf{v}^T A^{MP}\mathbf{v} \ge \mathbf{v}^T B^{MP}\mathbf{v}.$$

The same approach is directly applied to the matrices A^{MV} and B^{MV} where $\mu_1 = 0$, $\mu_{2,3} = 1/3$, and therefore $\lambda_1 = 1$, $\lambda_{2,3} = 2/3$.

The result of our local analysis is summarized in the next theorem.

Theorem 2. *Let us consider the non-conforming FEM problem (2.2) defined on a square mesh. Then:*

(i) *the sparse approximation B of the stiffness matrix A satisfies the conditions of Theorem 1 for a stable MIC(0) factorization;*

(ii) *the matrices B and A are spectrally equivalent where the next relative condition number estimate holds uniformly with respect to any possible jumps of the diffusion coefficients.*

$$\kappa\left(B^{-1}A\right) \leq 2 \tag{5.5}$$

The above result holds for both MP and MV cases.

6. **Numerical tests I**

The model problem $-\Delta u = f$ in the unit square is considered, where homogeneous Dirichlet boundary conditions are assumed at the bottom side. The presented numerical tests illustrate the PCG convergence rate of the studied MIC(0) preconditioners when the size of the discrete problem is varied. A relative stopping criterion $(\mathcal{C}^{-1}\, r^{n_{it}}, r^{n_{it}})/(\mathcal{C}^{-1}\, r^0, r^0) < \varepsilon$ is used in the PCG algorithm, where r^i stands for the residual at the i-th iteration step, $(\cdot,\cdot)$ is the standard Euclidean inner product, and $\varepsilon = 10^{-6}$. A uniform mesh is used, where $h = 1/n$, and the size of the discrete problem is $N = 2n(n+1)$.

Table 1. *PCG* iterations: $MIC(0)$ preconditioning in the cases MP and MV

Meshsize		n_{it}^{AA}		n_{it}^{AB}	
n	N	MP	MV	MP	MV
63	8064	51	48	34	39
127	32512	82	70	50	56
255	130560	133	101	71	81
511	523264	214	144	104	114
1023	2095104	292	208	149	167

The obtained numbers of iterations are reported in Table 1. Here again, MP and MV stand, respectively for the cases of mid-point and integral mid-value interpolation operators used to construct the nodal basis of the related non-conforming FEM basis. We denote here by n_{it}^{AA} and n_{it}^{AB} the number of iterations obtained when MIC(0) factorizations of A and B are used as preconditioners of A.

The following observations are derived from the presented numerical results:

- The number of iterations in all cases is $O(\sqrt{n}) = O(N^{1/4})$.
- The PCG convergence rate is better for the case MV. This is one more advantage of this variant of the rotated bilinear elements (see [12] for more details about the approximation properties).
- $n_{it}^{AB} < n_{it}^{AA}$. Note, that this is considerably better than what we have as a prediction from the uniform estimate from Theorem 2.

- A stable MIC(0) factorization of the matrix A^{MV} for the considered model problem has been obtained. It is important to note that the related conditions from Theorem 1 are not satisfied in this case.

7. **Parallel preconditioning algorithm**

We study in this section the parallel properties of the proposed algorithm. Our analysis is focused on the PCG solution of the linear algebraic system obtained. The preconditioner was introduced as $\mathcal{C} = \mathcal{C}_{MIC(0)}(B)$. Each PCG iteration consists of one solution of a system with the matrix $\mathcal{C}$, one matrix vector multiplication with the original matrix A, two inner products, and three linked vector triads of the form $\mathbf{v} := \alpha\mathbf{v} + \mathbf{u}$. Therefore the computational complexity of one PCG iteration is given by

$$\mathcal{N}_{PCG}^{it} \approx \mathcal{N}(\mathcal{C}^{-1}\mathbf{v}) + \mathcal{N}(A\mathbf{v}) + 10N \approx 34N.$$

In the general case, the solution of triangular systems with matrices $(X - L)$ and $(X - L)^t$ is typically recursive, see (3.2). This is the reason for considering MIC(0) an inherently sequential algorithm. We will show now how this disadvantage has been overcome by the sparse matrix B introduced.

To illustrate the basic idea, we will analyze in a more detailed form the model problem where $\Omega = (0, 1)^2$, the square mesh ω_h corresponds to the mesh size $h = 1/n$. In this case the size of the discrete problem is $N = 2n(n + 1)$. The structures of the matrices A and B are illustrated in Figure 3, where each of the diagonal blocks corresponds to one vertical line of the mesh if a column-wise numbering of the unknowns has been used (see also [11]). The important advantage of the matrix B is that all of its diagonal blocks are diagonal. In this case, the implementation of the PCG solution step $\mathcal{C}^{-1}\mathbf{v}$ is fully parallel within each of these blocks. One can see at this point how the construction of B has been inspired by the properties of the conforming linear FEM stiffness matrix corresponding to a skewed triangulation (see [10] for some more details). Following [13], we will assume that the computations and communications are not overlapping, and therefore, the parallel execution time is the sum of the computation and communication times. We will also assume that: a) the execution of M arithmetic operations on one processor takes time $T_a = Mt_a$, where t_a is the average unit time to perform one arithmetic operation on one processor (no vectorization); and b) the communication time to transfer M data elements from one processor to another can be approximated by $T_{com} = \ell(t_s + Mt_c)$, where t_s is the start-up time and t_c is the incremental time necessary for each of the M elements to be sent, and ℓ is the graph distance between the processors.

Let us consider a distributed memory parallel algorithm where the number of processors is p ($p > 2$), and let $n = mp$ with some natural number m. The computational domain is split in p equally sized strips. The processor P_k is responsible for the local computations corresponding to the k-th strip. Then, we get the following expressions for the communication times related to $\mathcal{C}^{-1}\mathbf{v}$ and $A\mathbf{v}$

$$T_{com}(C^{-1}\mathbf{v}) = 8n(t_s + t_c),$$

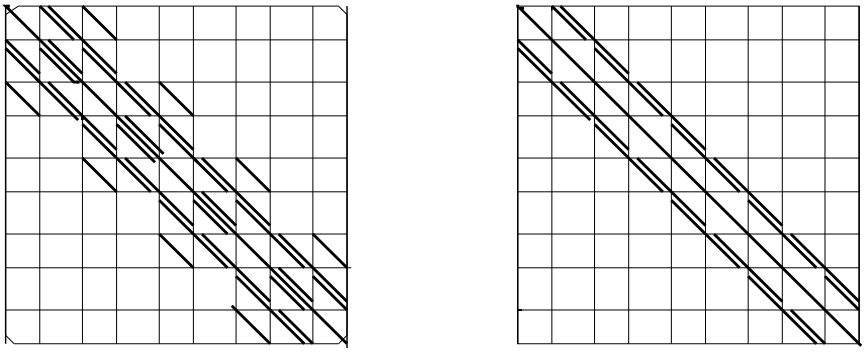

Figure 3. Sparsity pattern of the matrices A and B, $\Omega = (0,1)^2$.

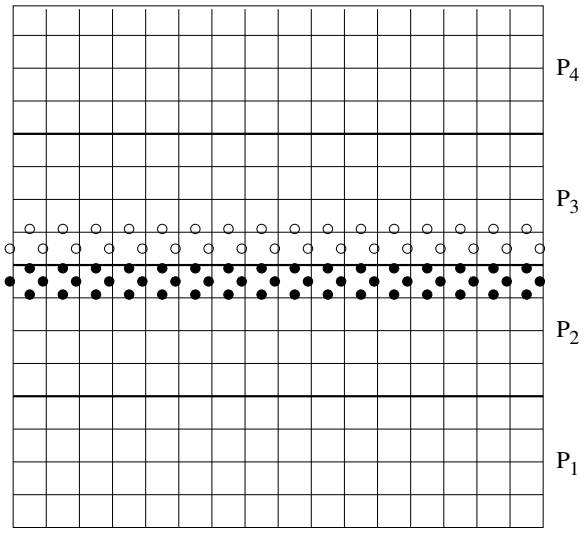

Figure 4. Stripwise data distribution between the processors in parallel implementation

$$T_{com}(A\mathbf{v}) = 4t_s + 2(3n+1)t_c.$$

Note that the above communications are completely local. 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 and will not be considered in our analysis. This setting leads to the following expression for the parallel time per one PCG iteration

$$T_p = T_p^{it} \approx 34\frac{2n(n+1)}{p}t_a + 8nt_s + 14nt_c. \tag{7.1}$$

What we conclude from (7.1) is that the parallel algorithm is asymptotically optimal. At the same time we should emphasize that the real speed-up is strongly dependent on the relations between t_s and t_c which means that in the general case good parallel efficiency could be achieved, if and only if, the size of the problem is large enough. This is readily seen from the results reported in the next section.

8. **Numerical tests II**

The parallel implementation of our C code is developed using the MPI (Message Passing Interface) standard. A set of numerical tests have been performed on a Beowulf type cluster. It consists of four dual processor Power Macintosh computers connected with a Bay Stack 350 Switch. Each node has 512 MB RAM and two processors Power PC G4/450MHz. The same model problem is used to illustrate the properties of the proposed parallel PCG algorithm and the related code. The results obtained are given in Table 2 in terms of the speed-up S_p and the efficiency E_p where

$$S_p = \frac{T_1}{T_p}, \qquad E_p = \frac{S_p}{p},$$

and where T_p stands for the parallel time to perform the code on p processors. A well

Table 2. Parallel performance of $PCG/MIC(0)$: $S_p = \frac{T_1}{T_p}$, $E_p = \frac{S_p}{p}$

MP					MV				
$\frac{n}{n_{it}}$	p	cpu	S_p	E_p	$\frac{n}{n_{it}}$	p	cpu	S_p	E_p
	1	6.02				1	6.82		
128	2	3.32	1.81	0.91	128	2	3.70	1.84	0.92
49	4	3.91	1.54	0.39	56	4	4.29	1.59	0.40
	8	3.67	1.64	0.21		8	4.18	1.63	0.20
	1	35.44				1	40.40		
256	2	19.52	1.82	0.91	256	2	22.21	1.82	0.91
71	4	15.83	2.24	0.56	81	4	18.11	2.23	0.56
	8	11.96	2.96	0.37		8	13.60	2.97	0.37
	1	208.95				1	238.94		
512	2	114.47	1.83	0.92	512	2	130.97	1.82	0.91
104	4	75.85	2.75	0.69	119	4	86.89	2.75	0.69
	8	49.48	4.22	0.53		8	56.56	4.22	0.53
	1	1198.45				1	1335.60		
1024	2	654.51	1.83	0.92	1024	2	762.31	1.75	0.88
148	4	382.87	3.13	0.78	167	4	432.61	3.09	0.77
	8	227.52	5.27	0.66		8	255.75	5.22	0.65

expressed asymptotic scalability of the algorithm is demonstrated by the test data presented. The parallel efficiency $E_8 \approx 0.65$ for $n = 1024$ can be evaluated as a good achievement for the problem considered.

Remark 2. *The cluster we have performed the reported numerical tests on has in fact a heterogeneous architecture, which is not the case for the best performance of a standard MPI code. For such a case, a combination of Open MP (within each of the shared memory two-processor node) and MPI could be recommended to improve total parallel efficiency.*

9. Concluding remarks

In this paper we have proposed a new MIC(0) preconditioner for the rotated linear non-conforming finite element systems for second order elliptic equations. We have proved that the introduced locally modified approximation of the original stiffness matrix has a relative condition number that is bounded uniformly with respect to both the problem size and the possible jumps of the coefficients. The algorithm has been analyzed in the cases of coefficient and mesh isotropy. Further, the derived estimates for the parallel time show that a good parallel scalability can be achieved for large scale problems. The presented numerical results show that the proposed scalable parallel preconditioner preserves and even improves the robustness and the computational efficiency of the standard MIC(0) factorization algorithm.

Our further plans include generalizations to 3-D case including modifications allowing for efficient treatment of coefficient and mesh anisotropy.

Acknowledgement. This work has been supported in part by MES Grant MM 801/89 and by the Center of Excellence BIS-21 Grant ICA1-2000-70016.

REFERENCES

1. Arbogast, T. and Chen, Z.: *On the implementation of mixed methods as nonconforming methods for second order elliptic problems*, Math. Comp., **64**, (1995), 943–972.
2. Arnold, D.N. and Brezzi, F.: *Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates*, RAIRO, Model. Math. Anal. Numer., **19**, (1985), 7–32.
3. Blaheta, R.: *Displacement decomposition - incomplete factorization preconditioning techniques for linear elasticity problems*, Numer. Lin. Alg. Appl., **1**, (1994), 107–126.
4. Bencheva, G. and Margenov, S.: *On a preconditioning strategy for rotated linear FEM elliptic systems*, Proceedings, PRISM'01, University of Nijmegen, 2001, 87-90.
5. Brezzi F. and Fortin M.: *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, Berlin, Heidelberg, 1991.
6. Chen Z.: *Analysis of mixed methods using conforming and nonconforming finite element methods*, RAIRO, Math. Model. Numer. Anal., **27**, (1993), 9–34.
7. Chow, E.: *Parallel implementation and practical use of sparse approximate inverses with a priori sparsity patterns*, Int'l J. High Perf. Comput. Appl., **15**, (2001), 56–74. (`http://www.llnl.gov/CASC/linear_solvers/pubs.html`)
8. Chow, E.: *A priori sparsity patterns for parallel sparse approximate inverse preconditioners*, SIAM Journal on Scientific Computing, **21**(5), (2000), 1804–1822. (`http://www.llnl.gov/CASC/linear_solvers/pubs.html`)

9. Gustafsson, I.: *Modified incomplete Cholesky (MIC) factorization*, In Preconditioning Methods; Theory and Applications, D.J. Evans, ed., Gordon and Breach, (1984), 265–293.

10. Gustafsson, I. and Lindskog, G.: *On parallel solution of linear elasticity problems: Part I Theory*, Numer. Lin. Alg. Appl., **5**, (1998), 123–139.

11. Lazarov, R. and Margenov, S.: *On a two-level parallel MIC(0) preconditioning of Crouzeix-Raviart non-conforming FEM systems*, in: I. Dimov, I. Lirkov, S. Margenov and Z. Zlatev, eds., Numerical Methods and Applications, Springer LNCS, **2542**, (2003) 191–200.

12. Rannacher, R. and Turek, S.: *Simple nonconforming quadrilateral Stokes Element*, Numerical Methods for Partial Differential Equations, **8**(2), (1992), 97-112.

13. Saad, Y. and Schultz, V.: *Data Communication in Parallel Architectures*, Parallel Comput., **11**, (1989), 131–150.