Parallel DD-MIC(0) Preconditioning of Rotated Trilinear FEM Elasticity Systems

Yavor Vutov

Institute for Parallel Processing, Bulgarian Academy of Sciences

Abstract. A new parallel preconditioning algorithm for 3D nonconforming FEM elasticity systems is presented. The preconditioner is constructed in two steps. First displacement decomposition of the stiffness matrix is used. Than MIC(0) factorization is applied to a proper auxiliary M-matrix to get an approximate factorization of the obtained block-diagonal matrix. The auxiliary matrix has a special block structure – its diagonal blocks are diagonal matrices themselves. This allows the solution of the preconditioning system to be performed efficiently in parallel. Estimates for the parallel times, speedups and efficiencies are derived. The performed parallel tests are in total agreement with them. The robustness of the proposed algorithm is confirmed by the presented experiments solving problems with strong coefficient jumps.

Key words: NC FEM, PCG, MIC(0), parallel algorithms

1 Introduction

We consider the weak formulation of the linear elasticity problem in the form: find $\mathbf{u} \in [H^1_E(\Omega)]^3 = \{\mathbf{v} \in [H^1(\Omega)]^3 : \mathbf{v}_{\Gamma_D} = \mathbf{u}_S\}$ such that

$$\int_\Omega [2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) + \lambda \operatorname{div}\mathbf{u} \operatorname{div}\mathbf{v}]d\Omega = \int_\Omega \mathbf{f}^t\mathbf{v}d\Omega + \int_{\Gamma_N} \mathbf{g}^t\mathbf{v}d\Gamma, \qquad (1)$$

$\forall \mathbf{v} \in [H^1_0(\Omega)]^3 = \{\mathbf{v} = [H^1(\Omega)]^3 : \mathbf{v}_{\Gamma_D} = 0\}$, with the positive constants λ and μ of Lamé, the symmetric strains $\varepsilon(\mathbf{u}) := 0.5(\nabla\mathbf{u} + (\nabla\mathbf{u})^t)$, the volume forces $\mathbf{f}$, and the boundary tractions $\mathbf{g}$, $\Gamma_N \cup \Gamma_D = \partial\Omega$. Nonconforming rotated trilinear elements of Rannacher-Turek [1] are used for the discretization of (1).

To obtain a stable saddle-point system one usually uses a mixed formulation for $\mathbf{u}$ and $\operatorname{div}\mathbf{u}$. By the choice of non-continuous finite elements for the dual variable, it can be eliminated at the macroelement level, and we get a symmetric positive definite FEM system in displacement variables. This approach is known as *reduced and selective integration*(RSI) technique, see [2].

Let $\Omega^H = w_1^H \times w_2^H \times w_3^H$ be a regular coarser decomposition of the domain $\Omega \subset \mathbb{R}^3$ into hexahedrons, and let the finer decomposition $\Omega^h = w_1^h \times w_2^h \times w_3^h$ be obtained by a regular refinement of each macro element $E \in \Omega^H$ into eight similar hexahedrons. The cube $\hat{e} = [-1, 1]^3$ is used as a reference element in the parametric definition of the rotated trilinear elements. For each $e \in \Omega^h$, let

$\psi_e : \hat{e} \to e$ be the trilinear 1–1 transformation. Then the nodal basis functions are defined by the relations $\{\phi_i\}_{i=1}^6 = \{\hat{\phi}_i \circ \psi_e^{-1}\}_{i=1}^6$, where $\hat{\phi}_i \in span\{1, \xi_j, \xi_j^2 - \xi_{j+1}^2, j = 1, 2, 3\}$. Mid-point (MP) and integral mid-value (MV) interpolation conditions can be used for determining the reference element basis functions $\{\hat{\phi}_i\}_{i=1}^6$. This leads to two different FEM spaces V^h, referred as Algorithm MP and Algorithm MV.

The RSI FEM discretization reads as follows: find $\mathbf{u}^h \in V_E^h$ such that

$$\sum_{e \in \Omega^h} \int_e \left[2\mu \varepsilon^*(\mathbf{u}^h) : \varepsilon^*(\mathbf{v}^h) + \lambda \, \mathrm{div}\, \mathbf{u}^h \, \mathrm{div}\, \mathbf{v}^h \right] de = \int_\Omega \mathbf{f}^t \mathbf{v}^h d\Omega + \int_{\Gamma_N} \mathbf{g}^t \mathbf{v}^h d\Gamma, \tag{2}$$

$\forall \mathbf{v}^h \in V_0^h$, where $\varepsilon^*(\mathbf{u}) := \nabla \mathbf{u} - 0.5 I_L^{Q^H} [\nabla \mathbf{u} - (\nabla \mathbf{u})^t]$, V_0^h is the FEM space, satisfying (in nodalwise sense) homogeneous boundary conditions on Γ_D, the operator $I_L^{Q^H}$ denotes the L^2–orthogonal projection onto Q^H, the space of piecewise constant functions on the coarser decomposition Ω^H of Ω. Than a standard computational procedure leads to a system of linear equations

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^1 \\ \mathbf{u}_h^2 \\ \mathbf{u}_h^3 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h^1 \\ \mathbf{f}_h^2 \\ \mathbf{f}_h^3 \end{bmatrix}. \tag{3}$$

Here the stiffness matrix K is written in block form corresponding to a separate displacements components ordering of the vector of nodal unknowns. Since K is sparse, symmetric and positive definite, we use the preconditioned conjugate gradient method (PCG) to solve the system (3). PCG is known to be the best solution method for such systems [3].

2 DD MIC(0) Preconditioning

Let us first recall some well known facts about the modified incomplete factorization MIC(0). Let us split the real $N \times N$ matrix $A = (a_{ij})$ in 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:

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

where $X = \mathrm{diag}(x_1, \ldots, x_N)$ 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 [4].

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

$$L \geq 0,\ A\mathbf{e} \geq 0,\ A\mathbf{e} + L^T\mathbf{e} > 0, \qquad \mathbf{e} = (1, \cdots, 1)^T \in \mathbb{R}^N,$$

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

$$x_i = a_{ii} - \sum_{k=1}^{i-1} \frac{a_{ik}}{x_k} \sum_{j=k+1}^{N} a_{kj} > 0 \tag{4}$$

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

Remark 1. The numerical tests presented in this work 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: $\tilde{d}_i = \xi a_{ii}$ if $a_{ii} \geq 2w_i$, and $\tilde{d}_i = \xi^{1/2} a_{ii}$ otherwise, where $0 < \xi < 1$ is a constant and $w_i = -\sum_{j>i} a_{ij}$.

We use PCG with a isotropic displacement decomposition (DD) MIC(0) factorization preconditioner in the form:

$$C_{DDMIC(0)}(K) = \begin{bmatrix} C_{MIC(0)}(B) & & \\ & C_{MIC(0)}(B) & \\ & & C_{MIC(0)}(B) \end{bmatrix}$$

Matrix B is a modification of the stiffness matrix A corresponding to the bilinear form

$$a(u^h, v^h) = \sum_{e \in \Omega^h} \int_e E \left(\sum_{i=1}^{3} \frac{\partial u^h}{\partial x_i} \frac{\partial v^h}{\partial x_i} \right) de.$$

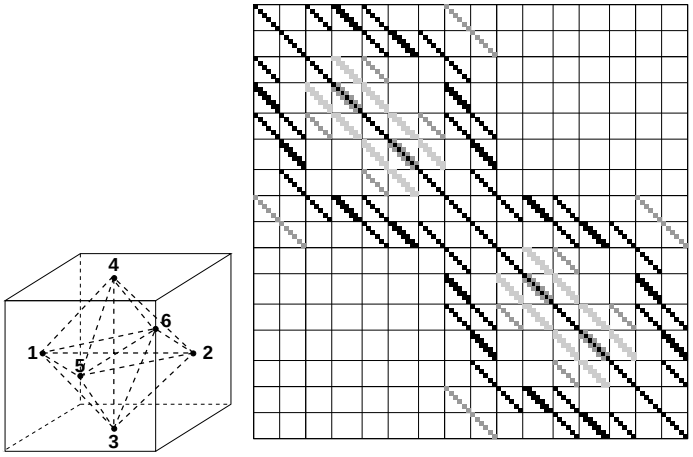
Here E is the modulus of elasticity. Such DD preconditioning for the coupled matrix K is theoretically motivated by the Korn's inequality which holds for the RSI FEM discretization under consideration [5]. The auxiliary matrix B is constructed element-by-element: Following the standard FEM assembling procedure we write A in the form $A = \sum_{e \in \Omega^h} L_e^T A_e L_e$, where L_e stands for the restriction mapping of the global vector of unknowns to the local one corresponding to the current element e and $A_e = \{a_{ij}\}_{i,j=1}^{6}$ is the element stiffness matrix The local node numbering and connectivity pattern is displayed in Fig. 1 (a). Now we will introduce the structure of two variants for local approximations B_e. They will later be referred to as Variant B1 and Variant B2.

Variant B1

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

Variant B2

$$B_e = \begin{bmatrix} b_{11} & & a_{13} & a_{14} & a_{15} & a_{16} \\ & b_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & b_{33} & & & \\ a_{41} & a_{42} & & b_{44} & & \\ a_{51} & a_{52} & & & b_{55} & \\ a_{61} & a_{62} & & & & b_{66} \end{bmatrix}$$



(a) (b)

Fig. 1. (a) Local node numbering and connectivity pattern; (b) Sparsity pattern of the matrices A and B (both variants) for a division of Ω into 2x2x6 hexahedrons. Non-zero elements are drawn with boxes: ■ non-zero in A and B (both variants), ■ non-zero in A and B Variant B1, ■ non-zero only in A. With thicker lines are bordered blocks in the matrix B Variant B2.

The matrices B_e are symmetric and positive semidefinite, with nonpositive off-diagonal entries, such that $B_e\mathbf{e} = A_e\mathbf{e}$, $\mathbf{e}^T = (1,1,1,1,1,1)$. Then we construct the global matrix $B = \sum_{e\in\Omega_h} \lambda_e^{(1)} L_e^T B_e L_e$, where $\{\lambda_e^{(i)}\}_{i=1}^5$ are the nontrivial eigenvalues of $B_e^{-1}A_e$ in ascending order. The matrix B is a M-matrix and has a special block structure with diagonal blocks being diagonal matrices, see Fig. 1(b). These blocks correspond to nodal lines and plains for variants B1 and B2, respectively. Lexicographic node numbering is used. This allows a stable MIC(0) factorization and efficient parallel implementation. It is important, that A and B are spectrally equivalent, and the relative condition number $\kappa(B^{-1}A)$ is uniformly bounded [6].

3 Parallel Algorithm

3.1 Description

The PCG algorithm is used for the solution of the linear system (3). Let us assume that the parallelogram domain Ω is decomposed into $n \times n \times n$ equal nonconforming hexahedral elements. The size of the resulting nonconforming FEM system is $N = 9n^2(n+1)$. To handle the systems with the preconditioner one has to solve three times 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 $k_{B1} = 2n^2 + 2n$ and $k_{B2} = 2n + 1$ 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 $p \leq n/2$ processors be denoted by P_1, P_2, ..., P_p. We distribute the entries of the vectors corresponding to each diagonal block of B among the processors. 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. Thus the processor P_j takes care of the local computations on the j-th strip.

3.2 Parallel Times

On each iteration in the PCG algorithm one matrix vector multiplication $K\mathbf{v}$, one solution of the preconditioner system $C\mathbf{x} = \mathbf{y}$, two inner products and three linked triads of the form $\mathbf{x} = \mathbf{y} + \alpha\mathbf{z}$ are computed. The matrix vector multiplication can be performed on the macroelement level. In the case of rectangular brick mesh, the number of non-zero elements in the macroelement stiffness matrix is 1740. The number of operations on each PCG iteration is:

$$\mathcal{N}^{it} = \mathcal{N}(K\mathbf{x}) + \mathcal{N}(C^{-1}\mathbf{x}) + 2\mathcal{N}(< .,. >) + 3\mathcal{N}(\mathbf{x} = \mathbf{y} + \alpha\mathbf{z})$$
$$\mathcal{N}^{it} \approx 24N + \mathcal{N}(C^{-1}\mathbf{x}) + 2N + 3N,\ \mathcal{N}^{it}_{B1} \approx 40N,\ \mathcal{N}^{it}_{B2} \approx 38N$$

An operation is assumed to consist of one addition and one multiplication. Estimations of the parallel execution times are derived with the following assumptions: a) executing M arithmetical operations on one processor lasts $T = Mt_a$, b) the time to transfer M data items between two neighboring processors can be approximated by $T^{comm} = t_s + Mt_c$, 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 neighboring processors can be done in parallel. We get the following expressions for the communication times :

$$T^{comm}(K\mathbf{v}) \approx 2t_s + \frac{4}{3}N^{2/3}t_c,$$
$$T^{comm}(C^{-1}_{B1}\mathbf{v}) \approx \frac{2}{3}N^{2/3}t_s + \frac{8}{3}N^{2/3}t_c, \quad T^{comm}(C^{-1}_{B2}\mathbf{v}) \approx \frac{2}{9}N^{1/3}t_s + \frac{8}{3}N^{2/3}t_c.$$

Two communication steps for the matrix vector multiplication are performed to avoid duplication of the computations or extra logic. For the solution of the triangular systems, after each nodal column (variant B1) or each nodal plain (variant B2) of unknowns is commputed some vector components must be exchnged.

The three systems of the preconditioner (one for each displacement) are solved simultaneously. Thus no extra communication steps for different displacements are required. 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

Table 1. Algorithm MP

		$J=1$			$J=10$			$J=100$			$J=1000$		
n	N	B0	B1	B2	B0	B1	B2	B0	B1	B2	B0	B1	B2
32	304 128	161	147	113	186	173	130	227	253	189	361	343	247
64	2 396 160	264	223	162	284	262	186	428	391	271	565	523	357
128	19 021 824	367	331	230	424	389	264	638	581	385	843	780	509
256	151 584 768		486	327		570	377		852	542		1 148	725

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:

$$T^{it}_{B1}(p) = \frac{40N}{p} + \frac{2}{3}N^{2/3}t_s + 4N^{2/3}t_c, \quad T^{it}_{B2}(p) = \frac{38N}{p} + \frac{2}{9}N^{1/3}t_s + 4N^{2/3}t_c$$

The relative speedup $S(p) = T(1)/T(p)$ and efficiency $E(p) = S(p)/p$, will grow with n 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 $n \gg p\,t_s/t_a$. The efficiency of Variant B2 is expected to be much better than the one of Variant B1, because about $3n$ times fewer messages are sent.

4 Benchmarking

4.1 Convergence Tests

The presented numerical tests illustrate the PCG convergence rate of the studied displacement decomposition algorithms when the size of the discrete problem and the coefficient jumps are varied.

The computational domain is $\Omega = [0,1]^3$ where homogeneous Dirichlet boundary conditions are assumed at the bottom face. An uniform mesh is used. The number of intervals in each of the coordinate directions for the finer grid is n.

A relative stopping criterion $(C^{-1}\mathbf{r}^i, \mathbf{r}^i)/(C^{-1}\mathbf{r}^0, \mathbf{r}^0) < \varepsilon^2$ is used in the PCG algorithm, where $\mathbf{r}^i$ stands for the residual at the i-th iteration step, and $\varepsilon = 10^{-6}$. The interaction between a soil media and a foundation element with varying elasticity modulus is considered. The foundation domain is $\Omega_f = [3/8, 5/8] \times [3/8, 5/8] \times [1/2, 1]$. The mechanical characteristics are $E_s = 10$MPa, $\nu_s = 0.2$ and $E_f = J10$MPa, $\nu_f = 0.2$ for the soil and foundation respectively. Experiments with jump $J \in \{1, 10, 100, 1000\}$ are performed. The force acting on the top of the foundation is 1MN. In Table 1 and Table 2 are collected the number of iterations for both variants B1 and B2 for Algorithms MP and MV respectively. In Table 1 also is added Variant B0 corresponding to the application of the MIC(0) factorization directly to the matrix A. Note that

Table 2. Algorithm MV

		$J=1$		$J=10$		$J=100$		$J=1000$	
n	N	B1	B2	B1	B2	B1	B2	B1	B2
32	304 128	173	255	197	280	313	348	405	411
64	2 396 160	295	648	310	744	486	904	630	1069
128	19 021 824	471	916	536	1 053	778	1 281	1 013	1 517
256	151 584 768	730	1 282	857	1 486	1 198	1 813	1 600	2 154

this is possible only for the Algorithm MP (because of the positive offdiagonal entries in A in algorithm MV) and only in a sequential program. One can clearly see the robustness of the proposed preconditioners. The number of iterations is of order $O(n^{1/2}) = O(N^{1/6})$. It is remarkable that for Algorithm MP, the number of iterations for Variants B2 are less than that number for Variant B1 which are even less than the iterations obtained without the modification of the matrix A.

4.2 Parallel Tests

Here we present execution times, speedups and efficiencies from experiments performed on three parallel computing platforms, referred to further 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 SMP processors at 1.9GHz and 16GB of RAM. The network bandwidth is 16Gb/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 2GB of RAM. Nodes are interconnected with a 1Gb Myrinet network. Platform C3 is a "Cray XD1" cabinet, fully equipped with 72 2-way nodes, totaling in 144 AMD Opteron processors at 2.4GHz. Each node has 4GB of memory. The CPUs are interconnected with the Cray RaidArray network with a bandwidth of 5.6Gb/s.

Since the parallel properties of the algorithm do not depend on the discretization type and the number of iterations , experiments only for Algorithm MP and for the case with the strongest coefficient jumps are performed. In Table 3 are shown sequential execution times $T(p)$ in seconds. The relative speedups $S(p)$ and efficiencies $E(p)$ different n and different number of processors p are collected in Table 4. Results both for Variant B1 and for Variant B2 are included. For a given number of processors the speedup and efficiency grow with the problem size. Conversely for fixed n, the efficiency decrease 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.

Table 3. Sequential Times

	Variant B1			Variant B2		
n	C1	C2	C3	C1	C2	C3
32	52.18	30.87	29.47	28.16	18.61	21.18
64	578.4	336.8	347.6	336.1	228.4	224.2
128	6596	3793	3556	3887	2556	2610

Table 4. Parallel Speedups and Efficiencies

		Variant B1						Variant B2					
		C1		C2		C3		C1		C2		C3	
n	p	$S(p)$	$E(p)$	$S(p)$	$E(p)$	$S(p)$	$E(p)$	$S(p)$	$E(p)$	$S(p)$	$E(p)$	$S(p)$	$E(p)$
32	2	1.49	0.74	1.31	0.66	1.77	0.88	1.93	0.96	1.33	0.66	1.97	0.99
	4	1.83	0.45	1.49	0.37	2.40	0.60	3.53	0.88	2.08	0.51	3.25	0.81
	8	2.11	0.26	1.22	0.15	3.34	0.42	5.78	0.72	3.07	0.38	5.20	0.65
	16	1.61	0.10	0.92	0.06	3.22	0.20	9.45	0.59	3.93	0.25	7.63	0.48
64	2	1.68	0.84	1.38	0.69	2.02	1.01	2.02	1.01	1.35	0.68	1.77	0.88
	4	2.46	0.61	1.98	0.49	3.17	0.79	3.92	0.98	2.49	0.62	3.50	0.87
	8	3.27	0.41	1.93	0.24	4.26	0.53	7.38	0.92	4.21	0.52	5.91	0.73
	16	3.78	0.23	2.06	0.13	6.03	0.38	12.83	0.81	6.53	0.40	8.64	0.54
128	2	1.82	0.91	1.51	0.76	1.56	0.78	2.00	1.00	1.49	0.74	1.93	0.96
	4	2.96	0.74	2.40	0.60	2.73	0.68	3.90	0.98	2.54	0.63	3.72	0.93
	8	4.50	0.56	2.70	0.34	5.34	0.67	7.33	0.92	4.59	0.57	7.30	0.91
	16	5.83	0.36	3.64	0.23	7.64	0.48	12.73	0.80	7.51	0.47	12.21	0.76

Acknowledgments

The numerical tests are supported via EC Project HPC-EUROPA RII3-CT-2003-506079. The author also gratefully acknowledges the support provided via EC INCO Grant BIS-21++ 016639/2005.

References

1. R. Rannacher, S. Turek: *Simple nonconforming quadrilateral Stokes Element*, Numer. Methods Partial Differential Equations 8(2) (1992), pp. 97–112.
2. D. Malkus, T. Hughes. Mixed finite element methods. Reduced and selective integration techniques: an uniform concepts. *Comp. Meth. Appl. Mech. Eng.,*15 ,1978.
3. O. Axelsson: *Iterative Solution Methods*. Cambridge University Press, 1994.
4. R. Blaheta: *Displacement Decomposition - incomplete factorization preconditioning techniques for linear elasticity problems*, Numer. Lin. Alg. Appl., **1** (1994), 107–126
5. O.Axelsson, I. Gustafsson. Iterative methods for the Navier equation of elasticity. *Comp. Meth. Appl. Mech. Engin.,* 15: 241-258, 1978.
6. P. Arbenz, S. Margenov, Y. Vutov: *Parallel MIC(0) Preconditioning of 3D Elliptic Problems Discretized by Rannacher–Turek Finite Elements*, Comput. Math. Appl. (to appear)