# On parallel solution of linear elasticity problems. Part II: Methods and some computer experiments

I. Gustafsson* and G. Lindskog*

*Department of Mathematics, Chalmers University of Technology, S-41296 Göteborg, Sweden*

SUMMARY

This is the second part of a trilogy on parallel solution of the linear elasticity problem. We consider the plain case of the problem with isotropic material, including discontinuous coefficients, and with homogeneous Dirichlet boundary condition. The discretized problem is solved by the preconditioned conjugate gradient (pcg) method.

In the first part of the trilogy block-*diagonal* preconditioners based on the separate displacement component (sdc) part of the elasticity equations were analysed. The preconditioning systems were solved by the pcg-method, i. e. inner iterations were performed. As preconditioner we used modified incomplete factorization MIC(0), where possibly the element matrices were modified in order to give M-matrices i.e. in order to garantee the existence of the MIC(0) factorization.

In the present paper, the second part, *full* block incomplete factorization preconditioners are presented and analysed. In order to avoid inner/outer iterations we also study a variant of the block-diagonal method and of the full block method, where the matrices of the inner systems are just replaced by their MIC(0)-factors. A comparison is made between the various methods with respect to rate of convergence and work per unknown. The fastest methods are implemented by message passing utilizing the MPI system.

In the third part of the trilogy we will focus on the use of higher order finite elements. 

KEY WORDS: elasticity problem; separate displacement component; preconditioned conjugate gradient method; parallel implementation

## 1. INTRODUCTION

In the first part of this trilogy, see Reference [1], the theoretical basis for this study is founded and will not be repeated here. We consider the plain strain case of the linear elasticity problem with isotropic material and Dirichlet boundary conditions.

A finite element approximation of this problem is based on a variational formulation

* Correspondence to: ivar@math.chalmers.se, lindskog@math.chalmers.se

including the bilinear form

$$a(u,v) = \int_\Omega \{\sum_i \frac{\partial u_i}{\partial x_i}\frac{\partial v_i}{\partial x_i} + \sum_{j\neq i}[\frac{1-\tilde{\nu}}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j} + \tilde{\nu}\frac{\partial u_j}{\partial x_j}\frac{\partial v_i}{\partial x_i} + \frac{1-\tilde{\nu}}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial v_j}{\partial x_i}]\}d\Omega \qquad (1)$$

where $\tilde{\nu}$ is a transformed contraction ratio. We recall from Reference [1] that since we consider homogeneous Dirichlet boundary conditions, (1) can be rewritten as

$$a(u,v) = \int_\Omega \{\sum_i \frac{\partial u_i}{\partial x_i}\frac{\partial v_i}{\partial x_i} + \frac{1-\tilde{\nu}}{2}\sum_{j\neq i}\frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j} + \frac{1+\tilde{\nu}}{2}\sum_{j\neq i}\frac{\partial u_j}{\partial x_j}\frac{\partial v_i}{\partial x_i}\}d\Omega.$$

Also recall that in practice $0 \leq \tilde{\nu} \leq 1$ and $\tilde{\nu} \to 1$ represents an incompressible material.

Further, in Reference [1] a preconditioning technique based on the separate displacement component of $a$, i.e.

$$\tilde{a}(u,v) = \int_\Omega \{\sum_i \frac{\partial u_i}{\partial x_i}\frac{\partial v_i}{\partial x_i} + \frac{1-\tilde{\nu}}{2}\sum_{j\neq i}\frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j}\}d\Omega \qquad (2)$$

plays an important part. In particular the following inequalities hold with $\sigma = \frac{1+\tilde{\nu}}{3-\tilde{\nu}}$:

$$(1-\sigma)\ \tilde{a}(u,u) \leq a(u,u) \leq (1+\sigma)\ \tilde{a}(u,u) \qquad (3)$$

For a natural numbering of the basis functions in the finite element method the assembled matrices corresponding to $a$ and $\tilde{a}$ become

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}$$

and

$$C_D = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$$

respectively, i.e. $C_D$ is block-diagonal.

The inequalities (3) lead to the upper bound

$$\kappa(C_D^{-1/2} A C_D^{-1/2}) \leq \frac{1+\sigma}{1-\sigma} = \frac{2}{1-\tilde{\nu}} \qquad (4)$$

for the condition number when $C_D$ is used as a preconditioning matrix for solving the system with matrix $A$ in (for instance) the conjugate gradient method, compare with the formula (6.8) in Reference [1]. Also recall from Reference [1] that this bound is independent of the discretization, in particular it is independent of the mesh size parameter $h$. The same bound is obtained also for a modification where the diagonal blocks in

$$C_D = \begin{bmatrix} C_{11} & 0 \\ 0 & C_{22} \end{bmatrix} \qquad (5)$$

are different from $A_{ii}$. In this modification two parameters, $\tau_1$ and $\tau_2$, play an important part. For $\tau_1 = \tau_2 = \frac{1-\tilde{\nu}}{2}$ we get $C_{ii} = A_{ii}$ i.e. the original separate displacement component preconditioning method.

In this paper we present a full block incomplete factorization preconditioning method. As in part one, see Reference [1] , one variant of the preconditioner involves the separate displacement component of $a$, see (2) and another variant involves a modification of the separate displacement part by two possibly different parameters $\tau_1$ and $\tau_2$. We analyse these methods with respect to the condition number of the preconditioned matrix. In both methods coupled inner/outer iterations are performed.

Other versions of the preconditioners are obtained by replacing the matrices of the inner systems by the modified incomplete MIC(0) factors. In this way we avoid the inner iterations. This idea is tested for the block-diagonal case as well as for the full block case. We compare the rate of convergence and the computational work for all considered methods. The two fastest methods are implemented in parallel on a Sun Enterprise 10000.

As a side-result of this paper, we have demonstrated how well the idea of modification on element level in order to obtain M-matrices, see Reference [2], works in practice. Different techniques to obtain an M-matrix, based on modification of the assembled elasticity matrix, are used for instance in Reference [3].

For the formulation of the elasticity problem and earlier research in this area we refer to the list of references in part one Reference [1]. For later presentations of iterative solvers to elasticity problems, including papers on separate displacement preconditioning, inner-outer iterations, 3D elasticity problems, locking-free finite element techniques, and parallel implementation, see References [4], [5], [6], [7], [8], [9], [3], and [10]. Note that the problem with locking, which is a consequence of ill-conditioning when the elasticity parameter $\tilde{\nu} \to 1$, see for instance Reference [4], is not studied in this paper. Also observe that the full-block preconditioning technique in this paper is not applicable to 3D problems.

The remainder of the paper is organized as follows: in Section 2 the full block incomplete factorization is presented and analysed. Numerical results concerning the rate of convergence and work per unknown are given in Section 3. In Section 4 we analyse a discontinuous problem. Results from the parallel implementation are given in Section 5 and finally we make some concluding remarks in Section 6.

## 2. FULL BLOCK INCOMPLETE FACTORIZATION

As an alternative to the block-diagonal preconditioning method given in the introduction we will also consider the full block incomplete factorization method represented by the preconditioning matrix

$$C_F = \begin{bmatrix} L_1 & 0 \\ A_{12}^T L_1^{-T} & L_2 \end{bmatrix} \begin{bmatrix} L_1^T & L_1^{-1} A_{12} \\ 0 & L_2^T \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & \bar{A}_{22} \end{bmatrix} \tag{6}$$

where

$$\bar{A}_{22} = A_{12}^T A_{11}^{-1} A_{12} + A_{22} \tag{7}$$

and $L_1 L_1^T = A_{11}$, $L_2 L_2^T = A_{22}$ are Cholesky factorizations. Below we will also consider *incomplete* Cholesky factorizations of $A_{11}$ and $A_{22}$.

Following a technique used in Reference [11] we will derive a bound for the condition number of $C_F^{-1/2} A C_F^{-1/2}$. It becomes clear from (3) that the eigenvalues of $C_D^{-1} A$, i.e. the eigenvalues of $C_D^{-1/2} A C_D^{-1/2}$, are in the interval $[1-\sigma, 1+\sigma]$.

We may write

$$C_D^{-1/2} A C_D^{-1/2} = \begin{bmatrix} A_{11}^{-1/2} & 0 \\ 0 & A_{22}^{-1/2} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} \begin{bmatrix} A_{11}^{-1/2} & 0 \\ 0 & A_{22}^{-1/2} \end{bmatrix} = \begin{bmatrix} I & F \\ F^T & I \end{bmatrix}$$

where $F = A_{11}^{-1/2} A_{12} A_{22}^{-1/2}$. Hence the eigenvalues of $\begin{bmatrix} 0 & F \\ F^T & 0 \end{bmatrix}$ are in the interval $[-\sigma, \sigma]$, or equivalently

$$\|F\|_2 = \| \begin{bmatrix} 0 & F \\ F^T & 0 \end{bmatrix} \|_2 \leq \sigma = \frac{1+\tilde{\nu}}{3-\tilde{\nu}}. \tag{8}$$

We will also use a generalization of the bound (6.5) in Reference [1] obtained by using (6.4) in Reference [1] and the formula $\mid ab \mid \leq \frac{1}{2}(\delta a^2 + \delta^{-1} b^2)$ valid for all real numbers $a$, $b$ and $\delta > 0$. This gives, for $u = (u_1, u_2)$, the lower bound

$$a(u,u) \geq min\{1 - \frac{1+\tilde{\nu}}{2}\theta\delta, 1 - \frac{1+\tilde{\nu}}{1-\tilde{\nu}}\delta(1-\theta)\}\tilde{a}(u_1,u_1)$$

$$+ min\{1 - \frac{1+\tilde{\nu}}{2}\theta\delta^{-1}, 1 - \frac{1+\tilde{\nu}}{1-\tilde{\nu}}\delta^{-1}(1-\theta)\}\tilde{a}(u_2,u_2). \tag{9}$$

If we choose $\theta = \frac{2}{3-\tilde{\nu}}$ and $\delta^{-1} = \frac{1+\tilde{\nu}}{2}\theta$ in (9) we get

$$a(u,u) \geq (1-\sigma^2)\tilde{a}(u_2,u_2) \tag{10}$$

where $\sigma = \frac{1+\tilde{\nu}}{3-\tilde{\nu}}$ (as before).

In the full block incomplete factorization we have, see (7)

$$\bar{A}_{22} = A_{22}^{1/2}(I + F^T F)A_{22}^{1/2}$$

and by (8) we get

$$x_2^T A_{22} x_2 \leq x_2^T \bar{A}_{22} x_2 \leq (1+\sigma^2) x_2^T A_{22} x_2 \tag{11}$$

for all vectors $x_2$

Finally, by using (10) and (11) we derive the following bounds for the quotient between the quadratic forms with $C_F$ and $A$:

$$\begin{aligned} 1 \leq \frac{x^T C_F x}{x^T A x} &= \frac{x_1^T A_{11} x_1 + 2x_1^T A_{12} x_2 + x_2^T \bar{A}_{22} x_2}{x_1^T A_{11} x_1 + 2x_1^T A_{12} x_2 + x_2^T A_{22} x_2} \\ &= 1 + \frac{x_2^T \bar{A}_{22} x_2 - x_2^T A_{22} x_2}{x_1^T A_{11} x_1 + 2x_1^T A_{12} x_2 + x_2^T A_{22} x_2} \\ \leq 1 + \frac{x_2^T \bar{A}_{22} x_2 - x_2^T A_{22} x_2}{(1-\sigma^2) x_2^T A_{22} x_2} &\leq 1 + \frac{\sigma^2}{1-\sigma^2} = \frac{1}{1-\sigma^2}. \end{aligned}$$

Hence, for the condition number we get the upper bound

$$\kappa(C_F^{-1/2} A C_F^{-1/2}) \leq \frac{1}{1-\sigma^2} = \frac{(3-\tilde{\nu})^2}{8(1-\tilde{\nu})} \tag{12}$$

and this bound is independent of the discretization, in particular it is independent of the mesh size parameter $h$.

We notice that when $\tilde{\nu}$ goes to 1 then the bound (12) goes to $\frac{1}{2(1-\tilde{\nu})}$ i.e. for almost incompressible materials the bound for this condition number is a factor 4 smaller than for the block-diagonal method, compare with (4).

We note that a system of linear equations with the preconditioning matrix $C_F$ in (6) i.e. a system of the form

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{12}^T A_{11}^{-1} A_{12} + A_{22} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

can readily be reformulated into the system

$$\begin{aligned} A_{22} r_2 &= g_2 - A_{12}^T A_{11}^{-1} g_1 \\ A_{11} r_1 &= g_1 - A_{12} r_2 \end{aligned} \tag{13}$$

Thus the solution can be calculated by solving three inner systems, two with matrix $A_{11}$ and one with matrix $A_{22}$. If we use an iterative method for these systems as well we get a method with coupled inner/outer iterations.

If we replace $A_{11}$ and $A_{22}$ in (13) by incomplete Cholesky factors $L_1 L_1^T \approx A_{11}$, $L_2 L_2^T \approx A_{22}$ we get only outer iterations, in general of course at the expense of a larger number of iterations. Note that this is equivalent to replacing $L_1$ and $L_2$ by incomplete Cholesky factors already in (6).

Like in the block-diagonal method in Reference [1] we also consider preconditioning based on the *modified* separate displacement component part

$$\hat{a}(u,v) = \int_\Omega \{ \sum_i \frac{\partial u_i}{\partial x_i} \frac{\partial v_i}{\partial x_i} + \tau \sum_{j \neq i} \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} \} d\Omega. \tag{14}$$

with $\tau$ possibly different from $\frac{1-\tilde{\nu}}{2}$.

In Reference [1] the following upper and lower bounds for $a(u,u)$ are derived with $\frac{1-\tilde{\nu}}{1+\tilde{\nu}} \leq \tau \leq 1$, $\eta = \frac{1-\tau}{1+\tau}$ and $\sigma = \frac{\tilde{\nu}+\tau}{1+\tau}$:

$$a(u,u) \leq max\{\frac{2}{1+\tau}, \frac{1-\tilde{\nu}}{2\tau}\} \hat{a}(u,u) \leq \frac{2}{1+\tau} \hat{a}(u,u) = (1+\eta) \hat{a}(u,u) \tag{15}$$

$$a(u,u) \geq \frac{1-\tilde{\nu}}{1+\tau} \hat{a}(u,u) = (1 - \frac{\tilde{\nu}+\tau}{1+\tau}) \hat{a}(u,u) = (1-\sigma) \hat{a}(u,u) \geq (1-\sigma) \hat{a}(u_2, u_2). \tag{16}$$

We recall from Reference [1] that these bounds imply that the bound (4) for the condition number for the block-diagonal preconditioning method still holds if $C_D$ is based on $\hat{a}$ instead of $\tilde{a}$ for $\frac{1-\tilde{\nu}}{3+\tilde{\nu}} \leq \tau \leq 1$.

The analysis for the full block preconditioning method with $\hat{a}$ instead of $\tilde{a}$ is now carried through similarly with the bounds (15) and (16) instead of (8) and (10). We get the following bound for the condition number, compare with (12):

$$\kappa(C_F^{-1/2} A C_F^{-1/2}) \leq 1 + \frac{max(\sigma,\eta)^2}{1-\sigma}. \tag{17}$$

Observe that $A_{11}$ and $A_{22}$ in $C_F$, see (6) and (7), are modified in this method.

We also give a result when different values at $\tau$ is used in $C_{11}$ ($\tau = \tau_1$) and $C_{22}$ ($\tau = \tau_2$), see also Section 7 in Reference [1]. Let $\tau_{min} = min(\tau_1, \tau_2)$, $\tau_{max} = max(\tau_1, \tau_2)$, $\eta = \frac{1-\tau_{min}}{1+\tau_{min}}$, $\sigma = \frac{\tilde{\nu}+\tau_{max}}{1+\tau_{max}}$. Then for $\frac{1-\tilde{\nu}}{3+\tilde{\nu}} \leq \tau_{min} \leq \tau_{max} \leq 1$, the bound (17) holds also in this case.

## 3. COMPARISON BETWEEN PRECONDITIONING METHODS

In this section we compare some preconditioning techniques for the conjugate gradient method, based on the theory in Section 2. The methods considered are:

*method 1 (m1)* The block-diagonal preconditioning method based on separate displacement component i.e. on the form $\tilde{a}(u, v)$

*method 2 (m2)* The block-diagonal preconditioning method based om modified separate displacement component i.e. on the form $\hat{a}(u, v)$, with parameters $\tau_1$ and $\tau_2$ in $C_{11}$ and $C_{22}$, respectively

*method 3 (m3)* The approximate block-diagonal preconditioning method i.e. with $L_1$ and $L_2$ being incomplete Cholesky factors of $A_{11}$ and $A_{22}$, respectively

*method 4 (m4)* The full block preconditioning method based on separate displacement component i.e. on the form $\tilde{a}(u, v)$

*method 5 (m5)* The full block preconditioning method based on modified separate displacement component i.e. on the form $\hat{a}(u, v)$, with parameters $\tau_1$ and $\tau_2$ in $C_{11}$ and $C_{22}$, respectively

*method 6 (m6)* The approximate full block preconditioning method i.e. with $L_1$ and $L_2$ being incomplete Cholesky factors of $A_{11}$ and $A_{22}$, respectively

We note that m1 and m2 are the methods described in the first part of this trilogy, Reference [1], where also some preliminary computer experiments are presented for these methods.

As mentioned in Section 2, we assume that the inner systems arising in the methods m1, m2, m4, and m5 are solved iteratively by the preconditioned conjugate gradient method too i.e. these methods involve coupled inner/outer iterations, while m3 and m6 involve just one level of iterations.

In order to obtain a high degree of parallelism with almost equal processor load we assume a proper node-numbering of a regular triangular mesh, see for instance Figure 1. This kind of node-numbering is motivated in Reference [1], where also other kinds of node-numberings are discussed. With the considered node-numbering we can (theoretically) utilize a number of parallel processors equal to the number of nodes along one side of the square in Figure 1.

We use the modified incomplete Cholesky MIC(0) method as incomplete factorization. In order to garantee existence of the MIC(0) factorization the element matrices are possibly modified by the technique presented in Reference [2] before the assembling and incomplete factorization. Whether this modification has to be performed depends both on the orientation $\theta$, the angle between the $x_1$-axis and the hypotenuses in the triangulation, and on the elasticity parameter $\tilde{\nu}$, for details see Reference [12]. An alternative to the modification on element level could be to use the generalized stable incomplete factorization in Reference [13]. This latter method, however, does not preserve the faster rate of convergence with respect to the mesh size parameter $h$.

Below we present computer experiments for our methods applied on the testproblem in Reference [1] i.e. the plain isotropic elasticity problem with homogeneous Dirichlet boundary conditions and constant body force on a unit square with various orientations $\theta$ and the parallel node-numbering indicated in Figure 1. We use zero starting vectors in outer as well as inner iterations and a relative residual stopping criterion $\|r^{(p)}\|_2 \leq \epsilon \|r^{(0)}\|_2$ with $\epsilon_o = 10^{-4}$ in the outer iterations and various values of $\epsilon_i$ for the inner iterations, see below.

Recall from Reference [1] that the number of outer iterations for m1 or m2 with

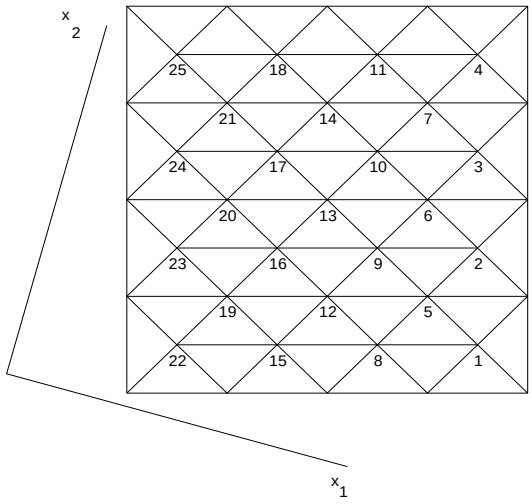


Figure 1. The test problem

$\frac{1-\tilde{\nu}}{3+\tilde{\nu}} \leq \tau_1 = \tau_2 \leq 1$ is at most

$$\frac{1}{2}\sqrt{\frac{2}{1-\tilde{\nu}}}\ \ln\frac{2}{\epsilon_o} + 1, \tag{18}$$

for m2 with $0 \leq \tau_1 = \tau_2 \leq \frac{1-\tilde{\nu}}{3+\tilde{\nu}}$ at most

$$\frac{1}{2}\sqrt{\frac{\tau+1}{2\tau}}\ \ln\frac{2}{\epsilon_o} + 1 \tag{19}$$

and for m2 with $\tau_1 \neq \tau_2$ and $\tau_{min} \geq \frac{1-\tilde{\nu}}{3+\tilde{\nu}}$ at most

$$\frac{1}{2}\sqrt{\frac{2(1+\tau_{max})}{(1+\tau_{min})(1-\tilde{\nu})}}\ \ln\frac{2}{\epsilon_o} + 1. \tag{20}$$

Similarly, from the bounds for the condition numbers (12) and (17) we conclude that the number of iterations for the method m4 is at most

$$\frac{3-\tilde{\nu}}{4}\sqrt{\frac{1}{2(1-\tilde{\nu})}}\ \ln\frac{2}{\epsilon_o} + 1 \tag{21}$$

and for m5 at most

$$\frac{1}{2}\sqrt{1+\frac{max(\sigma,\eta)^2}{1-\sigma}}\ \ln\frac{2}{\epsilon_o} + 1 \tag{22}$$

with $\sigma$ and $\eta$ given in Section 2.

In Table I we present the number of iterations for our methods when solving the model problem with $\tilde{\nu} = 0.995$ and $\theta = 0$. The inner systems are solved accurately, $\epsilon_i = 10^{-3}$. In

| method | m=15 | m=31 | m=63 | m=127 | m=255 | bound |
|---|---|---|---|---|---|---|
| m1 | 34 | 63 | 78 | 88 | 92 | 101 |
| m2,$\tau_1 = \tau_2 = 1$ | 51 | 67 | 75 | 80 | 83 | 101 |
| m2,$\tau_1 = \tau_2 = 0.1$ | 37 | 58 | 76 | 84 | 87 | 101 |
| m2,$\tau_1 = 0.2, \tau_2 = 0.5$ | 40 | 58 | 76 | 82 | 86 | 112 |
| m3 | 42 | 80 | 140 | 232 | 376 | - |
| m4 | 18 | 33 | 41 | 45 | 47 | 51 |
| m5,$\tau_1 = \tau_2 = 1$ | 44 | 52 | 68 | 74 | 79 | 100 |
| m5,$\tau_1 = \tau_2 = 0.1$ | 20 | 31 | 40 | 43 | 45 | 75 |
| m5,$\tau_1 = 0.05, \tau_2 = 0.5$ | 24 | 37 | 47 | 51 | 57 | 87 |
| m6 | 27 | 50 | 99 | 171 | 310 | - |

Table I. The number of outer iterations for various methods applied to the model problem of various size and the theoretical upper bound for this number

order to compare with the theory we also give the upper bounds for the number of iterations calculated from (18) - (22) in the various cases. The size of the problem is modelled by the value $m$, which is the number of diagonal blocks in the matrices $A_{ii},\ i = 1,\ 2$; $m = 7$ in Figure 1. The total number of unknowns in the system is then $m^2 + 1$.

The results indicate that the bounds (20) and (22) may be a bit too pessimistic, otherwise the results agree well with the theory.

Regarding the total amount of work it is not advisable to solve the inner iterations with such a high accuracy as $\epsilon_i = 10^{-3}$. If we relax the inner accuracy to $\epsilon_i = 10^{-1}$ the number of outer iterations increases but the total work becomes smaller. In Table II we give the total number of floating point operations (flops) per unknown for the test problem and various methods. Here, one flop consists of one multiplication plus one addition. The problem size is $m = 127,\ \theta = 0,\ \epsilon_i = 10^{-1}$, and $\tilde{\nu} = 0,\ 0.9$ or $0.995$.

In 3D the elasticity stiffness matrix has many more nonzero elements per row, thus the multiplication with it becomes more expensive. Then one should solve more accurately with the diagonal blocks in order to decrease the number of costly outer iterations.

We see that the simple methods m3 and m6 with just one level of iterations are preferable and about equal effective with m3 i.e. the approximate block-diagonal preconditioning is slightly better.

In order for comparison with the computer experiments in Reference [1] we also consider the test problem with $\theta = \pi/8$ and $m = 63$ for various values of $\tilde{\nu}$. Figure 2 gives the total number of flops for our methods applied to this problem. By the transformation $\sqrt{1/(1-\tilde{\nu})}$ for the 'x-axis' we discern the behaviour for $\tilde{\nu}$ close to one. We see that for this problem the method m6 i.e. the approximate full block preconditioning method is faster than the approximate block-diagonal preconditioning method (m3) for values of $\tilde{\nu}$ close to one. Thus, it may pay off to use the former more complicated method in such cases.

| method | $\tilde{\nu} = 0$ | $\tilde{\nu} = 0.9$ | $\tilde{\nu} = 0.995$ |
|---|---|---|---|
| m1 | 1270 | 4430 | 18900 |
| m2,$\tau_1 = \tau_2 = 1$ | 1120 | 2730 | 9320 |
| m2,$\tau_1 = \tau_2 = 0.1$ | 3190 | 4100 | 12920 |
| m2,$\tau_1 = 0.2, \tau_2 = 0.5$ | 1840 | 3020 | 9410 |
| m3 | 783 | 2133 | 6264 |
| m4 | 1122 | 3256 | 18012 |
| m5,$\tau_1 = \tau_2 = 1$ | 1835 | 4732 | 15046 |
| m5,$\tau_1 = \tau_2 = 0.1$ | 3706 | 3084 | 10546 |
| m5,$\tau_1 = 0.05, \tau_2 = 0.5$ | 3852 | 3096 | 7332 |
| m6 | 1238 | 2850 | 6412 |

Table II. The total work per unknown for various methods applied to the model problem with $m = 127,\ \theta = 0$ and $\epsilon_i = 10^{-1}$

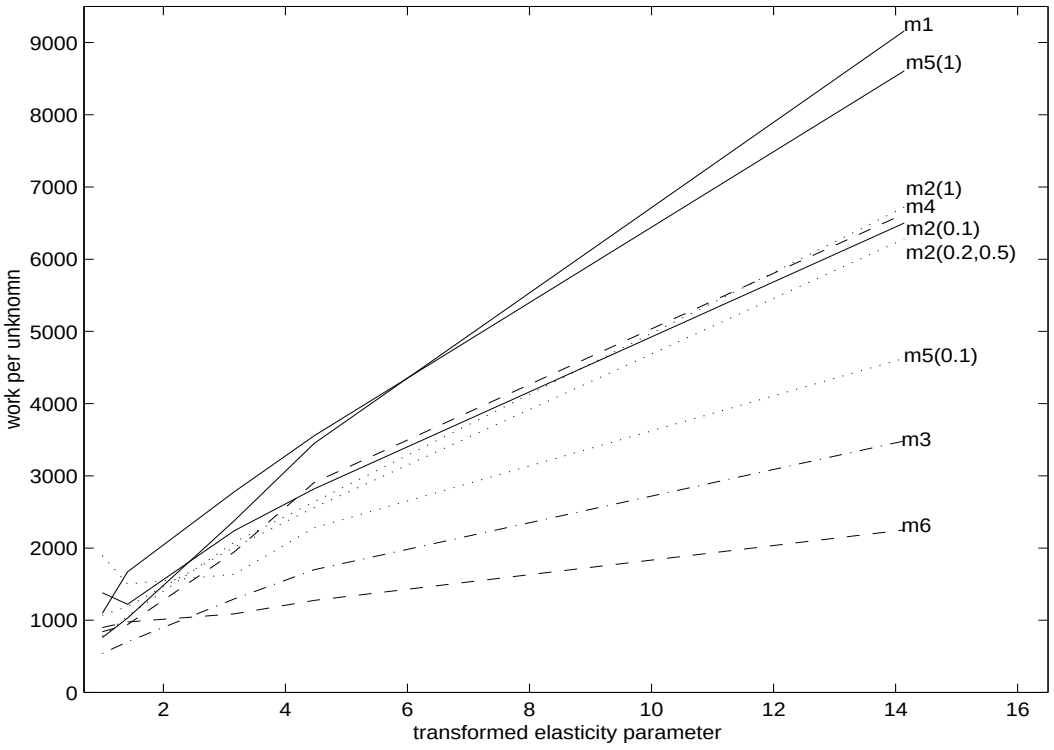


Figure 2. The work per unknown as a function of the elasticity parameter $\tilde{\nu}$ for various methods applied to a model problem

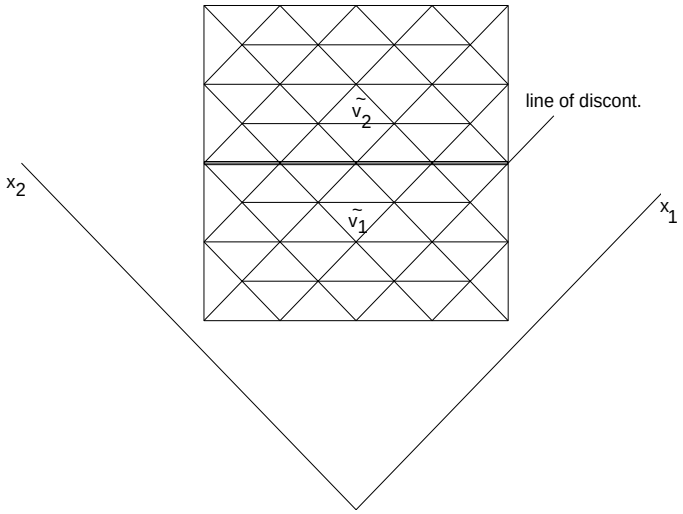


Figure 3. A discontinuous elasticity problem

## 4. A DISCONTINUOUS ELASTICITY PROBLEM

In this section we consider an elasticity problem with discontinuous material. The parameter $\tilde{\nu}$ is discontinuous along some interior boundary with $\tilde{\nu} = \tilde{\nu}_1$ in $\Omega_1$ and $\tilde{\nu} = \tilde{\nu}_2$ in $\Omega_2$.

At first we prove that the discontinuity does not effect our upper bound for the condition number in the block-diagonal preconditioning method i.e. the bound (4) holds with $\tilde{\nu}$ replaced by $\max(\tilde{\nu}_1, \tilde{\nu}_2)$. Assume, with no loss of generality, that $\tilde{\nu}_2 > \tilde{\nu}_1$.

Following Reference [1] we write

$$a(u,u) = \int_\Omega \{\sum_i (\frac{\partial u_i}{\partial x_i})^2 + \frac{1-\tilde{\nu}}{2} \sum_{j\neq i} (\frac{\partial u_i}{\partial x_j})^2 + \frac{1+\tilde{\nu}}{2} \sum_{j\neq i} \frac{\partial u_j}{\partial x_j}\frac{\partial u_i}{\partial x_i}\} d\Omega$$

$$\leq \int_\Omega \{\sum_i (\frac{\partial u_i}{\partial x_i})^2 + \frac{1-\tilde{\nu}}{2} \sum_{j\neq i} (\frac{\partial u_i}{\partial x_j})^2 + \frac{1+\tilde{\nu}_2}{2} \sum_{j\neq i} |\, \frac{\partial u_j}{\partial x_j}\frac{\partial u_i}{\partial x_i}\, |\} d\Omega$$

$$= \int_{\Omega_1} \{\sum_i (\frac{\partial u_i}{\partial x_i})^2 + \frac{1-\tilde{\nu}_1}{2} \sum_{j\neq i} (\frac{\partial u_i}{\partial x_j})^2\} d\Omega_1 + \int_{\Omega_2} \{\sum_i (\frac{\partial u_i}{\partial x_i})^2 + \frac{1-\tilde{\nu}_2}{2} \sum_{j\neq i} (\frac{\partial u_i}{\partial x_j})^2\} d\Omega_2$$

$$+ \frac{1+\tilde{\nu}_2}{2} \int_\Omega \sum_{j\neq i} |\, \frac{\partial u_j}{\partial x_j}\frac{\partial u_i}{\partial x_i}\, |\} d\Omega = \tilde{a}(u,u) + \frac{1+\tilde{\nu}_2}{2} \int_\Omega \sum_{j\neq i} |\, \frac{\partial u_j}{\partial x_j}\frac{\partial u_i}{\partial x_i}\, |\, d\Omega.$$

Using (6.5) in Reference [1] on each subdomain $\Omega_1$ and $\Omega_2$ we get, similarly to (6.6) in Reference

[1]:

$$a(u,u) \leq max\{1 + \frac{1+\tilde{\nu}_2}{2}\theta,\ 1 + \frac{1+\tilde{\nu}_2}{1-\tilde{\nu}_2}(1-\theta)\}\tilde{a}(u,u) \tag{23}$$

valid for all $u$ (in the space considered, see Reference [1]) and all $\theta \in [0,1]$. By chosing $\theta = 3/(3-\tilde{\nu}_2)$ in (23) we get

$$a(u,u) \leq \frac{4}{3-\tilde{\nu}_2}\tilde{a}(u,u). \tag{24}$$

Similarly we derive the lower bound

$$a(u,u) \geq \frac{2(1-\tilde{\nu}_2)}{3-\tilde{\nu}_2}\tilde{a}(u,u). \tag{25}$$

By combining (24) and (25) we conclude that the bound (4) for the condition number is still valid with $\tilde{\nu}$ replaced by $\tilde{\nu}_2$, i.e. the largest value of $\tilde{\nu}$ determines the upper bound for the condition number.

It is straightforward to carry through the analysis for the condition number of the other variants of preconditionings in the discontinuous case. For instance, the bound (12) holds for the full block incomplete factorization with $\tilde{\nu}$ replaced by $\tilde{\nu}_2$.

We now present some computer experiments for the discontinuous model problem where the parameter $\tilde{\nu}$ is discontinuous along an interior boundary with $\tilde{\nu} = \tilde{\nu}_1$ in $\Omega_1$ and $\tilde{\nu}_2$ in $\Omega_2$, see Figure 3. Apart from the discontinuity this is the model problem in Section 3 with $\theta = -\pi/4$. In this case the matrices $C_{11}$ and $C_{22}$ become M-matrices, see Reference [1], and hence there is no need for modification on element level in this problem. At first we would like to confirm that the number of (outer) iterations is fairly independent of the discontinuity as the obtained upper bound for the condition number indicates.

For our discontinuous model problem with $\tilde{\nu}_1 = 0,\ \tilde{\nu}_2 = 0.995,\ m = 255$ and inner system accuracy $\epsilon_i = 10^{-3}$, the block-diagonal method (m1) required 75 (outer) iterations and the full block method (m4) required 38 iterations. This should be compared with 74 and 37 iterations, respectively, for the corresponding continuous problem with $\tilde{\nu} = 0.995$. However, as is expected for a more ill-conditioned problem, the number of inner iterations increases with the degree of discontinuity.

For the fastest methods i.e. the approximate block-diagonal (m3) and approximate full block (m6) methods we thus get an increase in the number of iterations in the discontinuous problem compared with the continuous one. Recall that in these methods the inner systems are solved with low accuracy.

In Table III we compare the continuous model problem with $\tilde{\nu} = 0.995$ and the discontinuous model problem with $\tilde{\nu}_1 = 0,\ \tilde{\nu}_2 = 0.995$. We give the number of iterations for m3 and m6 and various sizes of the problem.

For the approximate block-diagonal method (m3) the increase in the number of iterations for the discontinuous problem is not dramatic but for the approximate full block method (m6) it is more pronounced. Also note that for large enough problems the method m3 is faster than the method m6 in particular for the discontinuous problem. The method m3 requires 27 operations per iteration per unknown while the method m6 requires 37.5 operations per iteration per unknown (for large enough problems).

| problem size | continuous | | discontinuous | |
|---|---|---|---|---|
| m | m3 | m6 | m3 | m6 |
| 15 | 19 | 11 | 29 | 20 |
| 31 | 31 | 21 | 51 | 44 |
| 63 | 53 | 31 | 81 | 82 |
| 127 | 74 | 59 | 140 | 154 |
| 255 | 151 | 124 | 222 | 316 |

Table III. The number of iterations for the methods m3 and m6 applied to a continuous and a discontinuous model problem

## 5. PRALLEL IMPLEMENTATION

For the parallel implementation we use Sun Enterprise 10000, where 16 processors have been at our disposal. The parallelization is performed by message passing implemented by the MPI, Message Passing Interface, system.

We consider the testproblem in Figure 1 with the node-numbering indicated. We have used the orientation $\theta = 0$ and the elasticity parameter $\tilde{\nu} = 0.995$.

We have implemented the conjugate gradient method without preconditioning, method 0 (m0), the approximate block-diagonal (m3) and the approximate full block (m6) preconditioning methods.

The goal for the parallelization is to get as equal processor load as possible and also to make effective message transfers. The main principle for realizing this is to make an equal decomposition of the domain between the processors and to let each processor calculate only the data elements corresponding to meshnodes in its own area of the domain. Data elements here refer to the vector elements in all types of calculations in the algorithms and to the parts of the scalar products calculated in each processor. In Figure 4 the problem size parameter $m = 11$ and the domain is partitioned between three processors, P1, P2 and P3.

The various calculations in our algorithms are scalar products, vector updates, matrix-vector multiplications and the solution of the preconditioning systems. The scalar products and the vector updates are easily parallelized since vector elements are needed only in the processors where they are calculated. Hence no vector elements are needed to be transfered to other processors. The scalar products are calculated by summing up the contributions from each processor. Matrix-vector multiplications are involved in the multiplication by the matrix $A$, see Section 1, and in method m6 also in the solution of the preconditioning system. The submatrices $A_{11}$, $A_{12}$ and $A_{22}$ involved in the matrix-vector multiplication have a non-zero structure given by the nonzero couplings shown in Figure 5. Because of the nonzero couplings between node $a$ and the nodes $c$, $d$, $f$ and $g$, see Figure 5, the parallel implementation of, say $A_{11}d$, $d$ a vector, by the main principle mentioned above, utilizes that elements of $d$ corresponding to nodes being nearest neighbours to the subdomain boundaries have to be transfered to the nearest neighbouring processor. The actual nodes are denoted by $\Box$ in Figure 4. The data transfers can all be made in parallel.

Concerning the parallel implementation of the preconditioning systems, the methods m3 and m6 both involve the solution of the systems $L_i L_i^T r_i = g_i$, $i = 1, 2$, see Section 2. We consider for simplicity only $L_1 h_1 = g_1$, where $h_1 = L_1^T r_1$. As is pointed out in Reference [1],

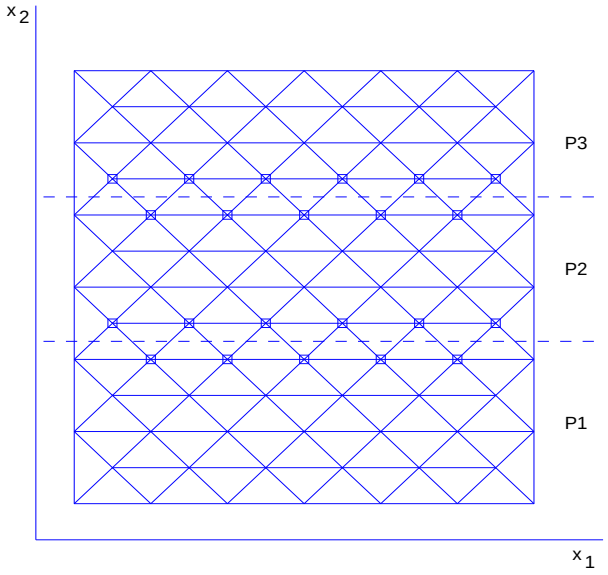


Figure 4. Decomposition of the domain between three processors. Data in the node s □ need to be transfered in the pcg-algorithm.

the proper nodenumbering for parallel solution of the preconditioning system and the non-zero couplings of a node to its neighbours give a block structure of $L_1$ with diagonal blocks in the diagonal. Hence the vector $h_1$ can be calculated in parallel on each vertical line of nodes in Figure 4. Because of the nonzero couplings between node $a$ and the nodes $c$ and $g$ in Figure 5, an assumption for the parallel solution is that the elements of $h_1$ corresponding to nodes □ in Figure 4 are transfered to the nearest neighbouring processor before the next vertical line of nodes is treated.

The numerical tests are performed for $m = 31, 63, 127, 255$ and 511. The stopping criterion for the iterations is $\|r^{(p)}\|_2 \leq 10^{-2}\|r^{(0)}\|_2$ . The programs are run in parallel in batch mode on $4, 8$ and 16 processors. In Table IV we give the number of iterations, the time for sequential execution and the speedup for various number of processors. The results show a quite good speedup for the larger problems. For lagre enough executions the times include some waiting periods due to other programs. Here this will occur in the largest sequential executions, resulting in a somewhat larger speedup than the number of processors in some cases. For small enough problems the use of an increased number of parallel processors will not cause an increased speedup. A reason for this is that the arithmetic operations are not enough to compensate the communication times.

An analysis of the parallel algorithm corresponding to method m3 shows that the amount of arithmetic operations per iteration per processor in terms including $m$ is $(27m^2 - 24m)/k$, where $k$ is the number of processors. The amount of communication per iteration is $12m$ data transports in each processor. Hence, the computational time will increase by a factor about 4 and the time for communication will increase by a factor 2 when $m$ is increased by a factor 2. This is in agreement with the numerical results shown in Table V for method m3 and the use

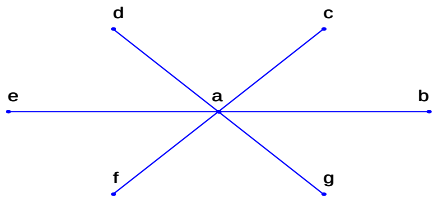


Figure 5. The nonzero couplings of a node to its neighbours

| method | | m=31 | m=63 | m=127 | m=255 | m=511 |
|---|---|---|---|---|---|---|
| m0 | Number of iterations | 145 | 321 | 732 | 1965 | 4253 |
| | Sequential time | 0.46s | 4.26s | 42.77s | 561.00s | 5821.93s |
| | Speedup: | | | | | |
| | 4 processors | 1.6 | 2.7 | 4.0 | 5.2 | 4.8 |
| | 8 processors | 1.2 | 3.0 | 5.3 | 8.7 | 11.6 |
| | 16 processors | - | 2.7 | 5.1 | 10.8 | 13.2 |
| m3 | Number of iterations | 41 | 68 | 107 | 183 | 299 |
| | Sequential time | 0.19s | 1.19s | 8.02s | 67.71s | 520.26s |
| | Speedup: | | | | | |
| | 4 processors | 1.0 | 1.7 | 2.5 | 3.8 | 4.6 |
| | 8 processors | 1.0 | 1.7 | 3.1 | 6.2 | 8.0 |
| | 16 processors | - | 1.5 | 3.0 | 7.0 | 10.7 |
| m6 | Number of iterations | 27 | 50 | 80 | 137 | 226 |
| | Sequential time | 0.16s | 0.75s | 8.81s | 71.53s | 603.42s |
| | Speedup: | | | | | |
| | 4 processors | 1.0 | 1.7 | 2.9 | 3.5 | 4.7 |
| | 8 processors | 1.0 | 2.0 | 3.4 | 5.3 | 7.1 |
| | 16 processors | - | 1.1 | 2.5 | 5.8 | 10.4 |

Table IV. Results from the parallel implementation.

of 4 and 8 processors. We give the computational time per iteration and the communication time per iteration for various $m$. The growing factors for computation and communication mentioned above and the fact that the time for communicaton is a large percent of the total time per iteration for the smaller problems explain the small speedup values for these problems.

| Work | m=31 | m=63 | m=127 | m=255 | m=511 |
|---|---|---|---|---|---|
| Computational time per iteration: | | | | | |
| 4 processors | 0.0012s | 0.0042s | 0.0171s | 0.0730s | 0.3214s |
| 8 processors | 0.00078s | 0.0028s | 0.0108s | 0.0376s | 0.1587s |
| Communication time per iteration: | | | | | |
| 4 processors | 0.0034s | 0.0060s | 0.0128s | 0.0243s | 0.0567s |
| 8 processors | 0.0038s | 0.0074s | 0.0133s | 0.0261s | 0.0587s |

Table V. Times in seconds for one iteration with method m3.

## 6. CONCLUSIONS

We have compared block-diagonal and full block incomplete factorization preconditioners for the solution of the linear elasticity problem by the preconditioned conjugate gradient method. For problems with almost incompressible material we have shown a condition number for the full block incomplete factorization method, which is a factor 4 smaller than for the block-diagonal method. Numerical tests on the rate of convergence confirm the theory. Furthermore, for large problems the theoretical upper bounds for the number of iterations are close to the number of iterations from our computer experiments.

The approximate block-diagonal and approximate full block preconditioners,where no inner iterations are performed, give a slower rate of convergence. However, the smaller amount of computational work per iteration results in the lowest work per unknown for these methods. For the testproblem with $\theta = \pi/8$ and $\tilde{\nu}$ close to one, the approximate full block preconditioning method is faster than the approximate block-diagonal preconditioning method. The approximate block-diagonal method is however slightly better than the approximate full block method for $\theta = 0$ and various $\tilde{\nu}$.

For a discontinuous elasticity problem it is shown that the condition number is independent of the discontinuity, which also is confirmed by numerical results. The number of outer iterations is almost independent of the discontinuity. The approximate block-diagonal method is the fastest method for this problem. However, the rate of convergence is slower than for the continuous problem.

The parallel implementation of the fastest methods, i.e. the approximate block-diagonal and the approximate full block methods show a good speedup for larger problems. We have used 4, 8 and 16 processors and the implementation is made by message passing using the MPI system.

It is obvious how to generalize the (approximate) block-diagonal preconditioning method to three space dimensions, see Reference [1]. The (approximate) full-block method, however, is not applicable to 3D problems. We note that the computer experiments have been performed on very regular domains and meshes. In Reference [12] our idea of node-numbering giving a high degree of partial parallelism and equal processor load is generalized to more general triangulations of more general domains and to 3D.

It is apparent how to solve the three subproblems corresponding to different space directions in 3D, appearing in the block-diagonal method, in parallel utilizing just three processors. This

low degree of parallelism is considered in Reference [4] for instance. Parallelism can also be achieved by dividing a mesh into subdomains. Our type of methods can then be applied to each subdomain to get a higher degree of parallelism.

In the third part of this trilogy Reference [14], we generalize our theory and methods to higher order approximation by using hierarchical finite element technique. Promising computer tests have been performed also for this generalization.

## ACKNOWLEDGEMENTS

We thank the two referees for many helpful comments and suggestions.

## REFERENCES

1. Gustafsson I, Lindskog G. On Parallel Solution of Linear Elasticity Problems, Part I: Theory. *Numerical Linear Algebra with Applications* 1998; **5**:123–139.
2. Gustafsson I. An incomplete factorization preconditioning method based on modification of element matrices. *BIT* 1996; **36**(1):86–100.
3. Saint-Georges P, Warzee G, Beauwens R, Notay Y. High-performance PCG solvers for FEM structural analysis. *International Journal for Numerical Methods in Engineering* 1996; **39**(8):1313–1340.
4. Axelsson O. On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods. *Mathematics and Computers in Simulations* 1999; **50**:11–30.
5. Axelsson O, Padiy A. On a robust and scalable linear elasticity solver based on a saddle point formulation. *Int. J. Numer. Methods Eng.* 1999; **44**(6):801-818.
6. Blaheta R, Jakl O, Stary J. A parallel CG solver for FE analysis of 3D problems in geomechanics. In *Proceedings of the international conference Geomechanics '96,* Rakowski Z, Balkema A A (eds). Brookfield: Rotterdam, 1997; 159–164.
7. Georgiev A, Margenov S. BSR BILU preconditioning for 3D simulation of pile foundation systems in nonline ar media. In *Notes Numer. Fluid Mechaniics. Large-scale scientific computations of engineering and environmental problems. Proceedings of the 1st workshop, Varna , Bulgaria, June 7-11, 1997,* Griebel et al. (eds). Vieweg: Wiesbaden, 1998; **62**:219–228.
8. Lirkov I, Margenov S, Owens R, Paprzycki M. A shared memory parallel implementation of block-circulant preconditioners. In *Notes Numer. Fluid Mechanics. Large-scale scientific computations of engineering and environmental problems. Proceedings of the 1st workshop, Varna, Bulgaria, June 7 -11, 1997,* Griebel et al. (eds). Vieweg: Wiesbaden, 1998 **62**:319–327.
9. Padiy A. On a robust multilevel method applied for solving large-scale linear elasticity problems. *Commun. Numer. Methods Eng.* 1999; **15**(3):153–165.
10. Valkenberg R, Warzee G, Saint-Georges P, Beauwens R, Notay Y. Parallel iterative solution of 3D elasticity equations. In *Proceedings of the international conference Geomechanics '96,* Rakowski Z, Balkema A A (eds). Brookfield: Rotterdam, 1997; 147–152.
11. Axelsson O, Gustafsson I. Preconditioning and two-level multigrid methods of arbitrary degree of approxima tion. *Mathematics of Computation* 1983; **40**:219–242.
12. Gustafsson I, Lindskog G. Parallel algorithms for orthotropic problems. *Numerical Linear Algebra with Applications,* 1996; **3**(3):1-23.
13. Kaporin I E. High Quality Preconditioning of a General Symmetric Positive Definite Matrix Bas ed on its $U^TU + U^TR + R^TU$-Decomposition. *Numerical Linear Algebra with Applications* 1998; **5**:483–509.
14. Gustafsson I, Lindskog G. On Parallel Solution of Linear Elasticity Problems, Part III: Higher order finite elements. (In progress).