Appendix 2

Application of large-scale FEM computations for design of new extracting technologies in mining

R. Blaheta, P. Byczanski, O. Jakl, R. Kohut, A. Kolcun, K. Krečmer, J. Starý
Institute of Geonics AS CR, Ostrava, Czech Republic
INCO KIT 977006 Report, 2002

SUMMARY

The paper describes an application of large-scale FE computations for the assessment of stress changes induced by mining. For the keynote part of the computations, which is the solution of large linear systems, the use of various space decomposition techniques are suggested and tested. Similar decomposition technique is exploited for locally improved FE analysis and solving multi-scale problems. The tests are performed on a Linux cluster and some other parallel computers with the use of a benchmark formulated from a real-life large-scale modelling of the effects induced by mining.

1 Introduction

The solution of geomechanical problems of underground mining, such as answering the questions about stability and safety of underground openings, requires understanding of the stress changes induced by mining. The stress computation can be done by the finite element method (FEM), which frequently uses very large number of degrees of freedom (DOF). This large-scale feature is due to

- considering large 3D domains, which are used to involve far fields effects and enable to use boundary conditions corresponding to the primary uninfluenced stress state,
- necessity to solve and compare a higher number of construction stages and design variants. Frequently, it is also necessary to consider additional variants corresponding to different choices of input data, which are mostly given with a considerable uncertainty.

The computational requirements can be further increased by modelling more complicated behaviour of geomaterials, involving time effects as well as coupling of the mechanical behaviour with other phenomena such as e.g. underground water flow.

The large scale FEM computations give specific requirements to grid generation, assembling of the system of equations, solution of this systems and visualization of the results.

This paper describes an example of large-scale modelling of the stress changes induced by mining and shows efficiency of FEM computations, which use structured and composite FE meshes and various space decomposition techniques.

Especially, we shall concentrate on the solution of linear systems, which constitutes the most laborious part of the FEM computations. For the solution of large linear systems, we shall use decomposition techniques, which are also suitable for parallel computations.

The paper is organized as follows. In Section 2, we describe an example of large-scale modelling, which will illustrate the requirements of the FEM computations. Section 3 describes the use of structured and composite grids for the finite element analysis. In Section 4, we present a general space decomposition technique and its variants. The described decomposition technique is used for parallelization of solvers based on the conjugate gradient method as well as for the construction of efficient preconditioners. In implementation of these preconditioners, we allow inexact solution of the arising subproblems by inner iterations. This freedom, which can be very helpful for tuning the performance of algorithms and balance of the load of processors, is compensated by the use of generalized conjugate gradients with explicit orthogonalization of the search direction in the outer iterations.

The following sections describe three specific decomposition techniques as displacement decomposition, domain decomposition and composite meshes. All these techniques were implemented in the FE software GEM from the Institute of Geonics AS CR and tested on benchmarks specified in Section 2. The paper is finished by concluding remarks in Section 8.

Note that in the sequel $\langle u,v\rangle = u^T v$ and $\parallel u \parallel = \sqrt{\langle u,u\rangle}$ denote respectively the standard scalar product and norm in the space of n dimensional real algebraic vectors R^n.

2 An example of large scale modelling

As an example of large scale geomechanical problems, which requires an application of high-performance computing, we shall describe the assessment of the stress field changes induced by mining at the uranium ore deposit Rožná in the Czech Republic, cf. [12].

The deposit is situated in metamorphic sedimentary-effusive rocks with uranium mineralization of hydrothermal origin situated in zones of veins arising from longitudinal faults with inclination 45° - 70° to the West, see Figure 1.

The modelling is performed in 3D domain with the dimensions 1430×550×600 meters, which can be seen in Figure 2. For the finite element modelling, this domain is divided into hexahedral bricks, which are subsequently divided, each brick into six tetrahedra. The finite element analysis with linear tetrahedral elements uses 3 873 264 degrees of freedom. The exploited mesh can be seen in Figure 2. The loading is given by the weight of the materials in the computational domain and the action of the surrounding rocks existing under the pre-mining stress state. With respect to the performed measurements of this pre-mining stress by the hydrofracturing technique, the modelling was performed for two extreme cases: the isotropic pre-mining stress $\sigma_{xx} = \sigma_{yy} = \sigma_{zz},\ \sigma_{xy} = \sigma_{yz} = \sigma_{zx} = 0$ and the anisotropic one $\sigma_{xx} = \sigma_{yy} = 1.25\,\sigma_{zz},\ \sigma_{xy} = 0.75\,\sigma_{zz},\ \sigma_{yz} = \sigma_{zx} = 0$. The vertical stress σ_{zz} is given by the weight of the rocks, $\sigma_{zz} = -\gamma\, h$ where γ is the averaged specific weight of the rocks, h is the depth under surface. For a future modelling, an identification procedure was suggested in [8], which allows to reduce the uncertainties concerning the pre-mining stress state.

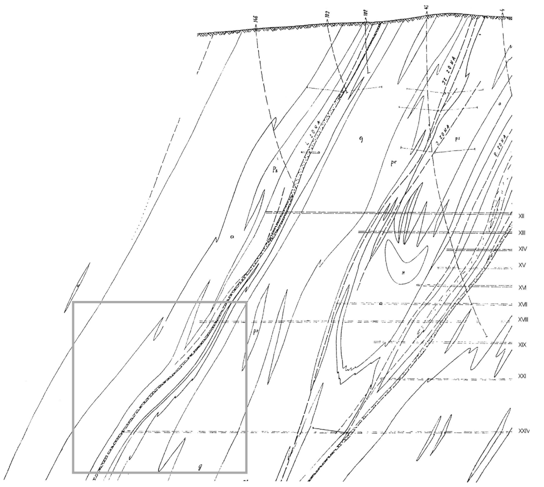

Figure 1: East-West cross-section of the deposit with 4th zone and the considered area in the left down corner.

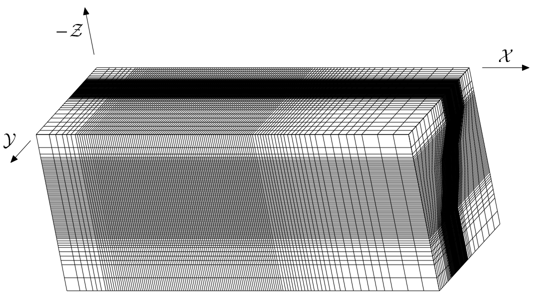

Figure 2: The considered domain of $1430 \times 550 \times 600$ metres with FE mesh with $124{\times}137{\times}76$ nodes.

The modelling itself consists of solving a sequence of boundary value problems with different material distributions, which corresponds to four selected stages of mining. In these stages, we solve the equilibrium equations with the use of effective linear constitutive relations. Such approach requires an estimate of the effective elastic moduli for the rocks, ore as well as for the damaged rocks (the goaf), which appear in the vicinity of the mine openings. Moreover, we have to estimate the extent of the zone of damaged material.

As the pre-mining stress is not compatible with any apriori known displacements, the modelling sequence consists of solving four boundary value problems with pure force (Neumann) boundary conditions. In this case, the use of different weights of intact and damaged materials can lead to some global disbalance of forces, which can be overcome by finding generalized solution of the arising singular systems. Note that stresses, which are subject of our primary interest, are unique.

The above sequence of four Neumann problems (Neumann sequence) can be solved by iterative solvers based on the conjugate gradient method. But more advantageous may be to use another sequence of problems (Dirichlet sequence), which consists of one auxiliary problem with the pure Neumannn boundary conditions for the computation of pre-mining displacements, which are then used as pure displacement (Dirichlet) boundary conditions in the subsequent modelling steps. Note that the maximal difference in the stress computed by these two sequences was found to be less than 7 %, which is acceptable for the performed modelling.

The described modelling concerns the stress changes in the whole considered domain. For detailed modelling of the stress changes in the vicinity of the active slope, we need to use finer grid and consider more details. This local analysis, which can be used e.g. for comparison of

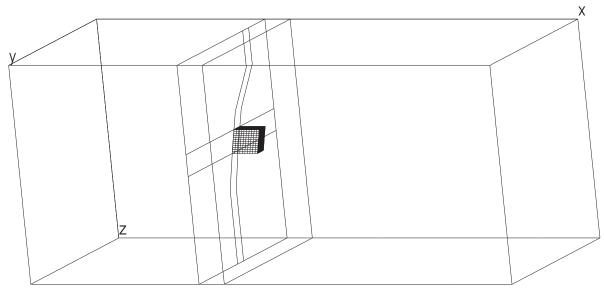

Figure 3: Local subdomain $80 \times 28 \times 60$ metres with refined local FE mesh with $49 \times 113 \times 65$ nodes.

different mining technologies as overhand and underhand stoping [12], can be performed by using quadruple refinement in a local domain with the dimensions $80 \times 28 \times 60$ metres, see Figure 3. Because, we are not able to specify sufficiently accurate boundary conditions for the local problem, we shall solve a composite global–local problem by the iterative composite grid technique described further in the paper. This composite technique may also be very helpful for other coupled far-field and near-field, two–scale or multi–scale problems.

From the described modelling, we shall derive two benchmarks which will be used for testing the solvers in the following sections.

Benchmark **BM1** – the linear system arising from the FE discretization of the elasticity problem corresponding to the fourth stage of mining at the Rožná deposit. As boundary conditions, we use the pre–mining displacements computed in a preliminary stage. Note that the *dimension of the system is about four million.*

Benchmark **BM2** – composite global - local problem, with the global problem specified as in BM1 and the local problem showed in Figure 3 with *an additional one million degrees of freedom.*

Note that the zero initial guess will be used in the benchmarks, although a better initial guess given by the solution from the previous stage of mining in the case of BM1 or by the solution of the global problem in the case of BM2 can be available.

3 Structured and composite FE meshes

For the large–scale finite element computations, we use structured meshes, which can be viewed as an adaptation of a regular (reference) mesh to the solved problem. More precisely, we shall define the reference mesh as tetrahedral mesh arising from any conforming partition of hexahedral mesh with the nodes (i, j, k), $i = 1, \dots, n_x$, $j = 1, \dots, n_y$ and $k = 1, \dots, n_z$. Then the structured mesh arises by adaptation, which means suitable displacement of the reference mesh nodes such that the tetrahedral elements remain to create a conforming FE partition. The freedom in the partition of the hexahedra into tetrahedra as well as the possibility that some tetrahedra remain void increase the flexibility of the considered structured meshes.

The restriction to structured meshes gives the following benefits:

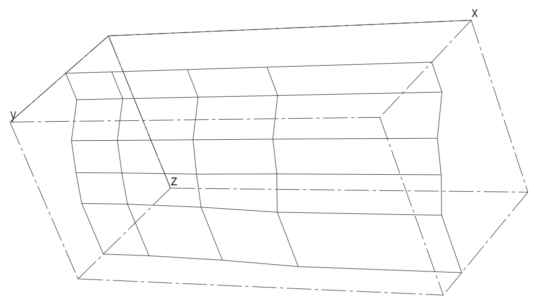

Figure 4: Initial mesh with $5 \times 6 \times 11$ nodes for construction of the mesh from Figure 2.

- simple construction of the mesh generator based on subsequent refinement of the mesh and interpolation of the data,
- efficient storage of the stiffness matrices exploiting the regular stencil or diagonal formats, which both are suitable for the construction of fast matrix handling procedures,
- possibility of simple balanced mesh partitioning as well as construction of transfer operators for one-level and two–level Schwarz methods,
- simplification of the visualization techniques.

The construction of the structured meshes starts from an initial very coarse approximation of the modelled situation. For the modelling described in the previous section, such initial mesh can be seen from Figure 4. This mesh is then gradually refined, with the nodal coordinates and material distribution generated by interpolation. In this refinement procedure, some details are modified by supplying data corresponding to the smaller and smaller details of the modelled situation.

Worse flexibility of the structured meshes can be further compensated by the use of *composite meshes* consisting of a global structured mesh and one or more local structured meshes in the case of a two-level refinement. The hierarchy of the refinement can be extended to more levels. More details about the use of composite meshes will be given in Section 7, see also [11]. The use of composite meshes avoids the not easy task of generation of locally refined tetrahedral meshes. Moreover, the use of composite meshes is more advantageous for mesh adaptation according to aposteriori error estimators.

4 General space decomposition (SD) technique

Let us focus on the solution of an elliptic linear boundary value problem, which is variationally formulated in a space of functions $\mathcal{V}$ by the finite element method, see e.g. [3], [13] and [15] for the details. Let V be a finite element subspace of $\mathcal{V}$ with the FE basis $\{\phi_i\}_{i=1}^n$, which defines isomorphism of V and the space of real algebraic vectors R^n. The FE approximation to the solution of the boundary value problem then takes the form

$$u_V = \sum_{i=1}^{n} u_i \phi_i \tag{4.1}$$

where u_i are components of a vector $u \in R^n$, which can be computed from the solution of the linear system

$$Au = b\,, \quad u, b \in R^n \tag{4.2}$$

where A is the stiffness matrix, which is symmetric positive definite, sparse and large, see again [3], [13] and [15]. For 3D analysis of elasticity problems arising from the mathematical modelling of the stress changes induced by mining, the dimension n of the system (4.2) can be very high, typically in the range from 10^5 to 10^7.

The symmetry and positive definiteness of the matrix A allow to solve (4.2) by the following preconditioned conjugate gradient (PCG) method.

Algorithm CG

given u^0
compute $r^0 = b - Au^0$, $g^0 = G(r^0)$, $v^0 = g^0$
for $i = 0, 1, \ldots$ **until** $\| r^i \| \le \varepsilon \| b \|$ **do**

$$\begin{aligned}
w^i &= Av^i \\
\alpha_i &= \langle r^i, g^i \rangle / \langle w^i, v^i \rangle \\
u^{i+1} &= u^i + \alpha_i v^i \\
r^{i+1} &= r^i - \alpha_i w^i \\
g^{i+1} &= G(r^{i+1}) \\
\beta_{i+1} &= \langle g^{i+1}, r^{i+1} \rangle / \langle g^i, r^i \rangle \\
v^{i+1} &= g^{i+1} + \beta_{i+1} v^i
\end{aligned}$$

end

In the above PCG algorithm, G is the preconditioner, which is used for computation of the pseudoresidual g. Note that g should be a vector that is closer in its direction to the error $A^{-1}r$ than the residual r. In computer implementation, g can share the computer memory with the auxiliary vector w.

The preconditioner can be constructed from the *space decomposition* of the form

$$V = V_1 + \ldots + V_p\,, \tag{4.3}$$

where V_k are subspaces of V, which are not necessarily linearly independent. Let V_k be isomorphic to R^{n_k}, then we can construct transfer matrices I_k, R_k, which represent the operations

- $I_k : R^{n_k} \to R^n$, which is the prolongation given by the inclusion $V_k \subset V$,
- $R_k : R^n \to R^{n_k}$, which is the restriction to the subspace, $R_k = I_k^T$.

These transfer matrices allow to introduce matrices $A_k = R_k A I_k$ corresponding to the subproblems on the subspaces. The matrices A_k will be again symmetric and positive definite.

Now, a class of space decomposition preconditioners can be introduced by using the following SD algorithm for computation of the pseudoresidual g from the residual r.

Algorithm SD

$g^0 = 0$
for $k = 1, \dots, p$ **do**
  $g^k = g^{k-1} + I_k A_k^{-1} R_k z^k$
end
$g = g^p$

A simple choice $z^k = r$ gives so called *additive preconditioner*

$$G(r) = G_A r \,, \; G_A = \sum_{k=1}^{p} I_k A_k^{-1} R_k \tag{4.4}$$

which is easily parallelizable and directly applicable in the PCG method, as G_A is symmetric positive definite.

The use of updated residuals $z^k = r - A g^{k-1}$ gives so called *multiplicative preconditioner,* which can be represented by a nonsymmetric linear mapping G_M. To get again symmetric positive definite preconditioner, we have to continue the corrections in the SD algorithm in the reverse order, i.e. the loop should be performed for $k = 1, \dots p, p-1, \dots, 1$. Such modification defines the *symmetrized multiplicative preconditioner.* More details about general space decomposition methods can be found e.g. in [21].

The operation $w_k = A_k^{-1} v_k$, which appears in all space decomposition preconditioners, can be replaced by an inexact solution of the system $A_k w_k = v_k$ by inner iterations. The inner iterations can be stopped to get an approximation $S_k(v_k)$ to $A_k^{-1} v_k$ with the accuracy

$$\| v_k - A_k S_k(v_k) \| \leq \varepsilon_0 \| v_k \| \,. \tag{4.5}$$

In the case of multiplicative preconditioner or inexact solution of the subproblems, the space decomposition preconditioner cannot be represented by linear symmetric positive definite mapping. In this case the standard PCG method may be not efficient or can even fail. In these cases, it can be favourable to use the generalized preconditioned conjugate gradient method with explicit orthogonalization of the new search direction to m previous search directions. This GPCG[m] algorithm is described and analyzed in [1], [18] and [9].

The space decomposition can be also used for parallelization of the solution of the FE systems (4.2). Typically, p processors can be employed for working with the data corresponding

Algorithm GPCG[m]

given u^0
compute $r^0 = b - Au^0$, $g^0 = G(r^0)$, $v^0 = g^0$
for $i = 0, 1, \ldots$ **until** $\| r^i \| \leq \varepsilon \| b \|$ **do**

$$w^i = Av^i$$
$$\alpha_i = \langle r^i, g^i \rangle / \langle w^i, v^i \rangle$$
$$u^{i+1} = u^i + \alpha_i v^i$$
$$r^{i+1} = r^i - \alpha_i w^i$$
$$g^{i+1} = G(r^{i+1})$$

for $k = 1, \ldots, \min\{i+1, m\}$ **do**

$$\beta_{i+1}^{(k)} = (\langle g^{i+1}, r^{i+2-k} \rangle - \langle g^{i+1}, r^{i+1-k} \rangle) / \langle g^{i+1-k}, r^{i+1-k} \rangle$$
$$g^{i+1} = g^{i+1} + \beta_{i+1}^{(k)} v^{i+1-k}$$

end

$$v^{i+1} = g^{i+1}$$

end

to p subspaces. Nonoverlapping data structures are used for matrix–vector multiplication, scalar product and vector updates with SAXPY operations, whereas both nonoverlapping or overlapping data structures can be used for implementation of the preconditioners. Additive space decomposition preconditioners allow straightforward parallelization whereas multiplicative preconditioners are parallelizable only in the case that some groups of subproblems are independent and the computation is suitably ordered.

The use of inner–outer iterations can be favourable for parallelization as it allows to balance the load of the processors as well as tune the ratio between computations and communications.

5 Displacement decomposition (DiD)

Let us start with a brief description of the *displacement decomposition* method, which was initally introduced for the construction of incomplete factorizations, see [5], [7]. The use of the displacement decomposition for parallel computations was described in [10]. Further results about DiD can be found in [2] and [9].

We shall consider the solution of 3D elasticity problems by the finite element method with linear tetrahedral or more generally Lagrangian finite elements. In this case, the degrees of freedom u_i in (4.1) represent nodal displacements in the coordinate directions. Then it is easy to define the displacement decomposition $V = V_1 + V_2 + V_3$, where V_k corresponds to the vectors, which have nonzero degrees of freedom only in the k-th coordinate direction.

We tested the efficiency of the *additive DiD preconditioner* with subproblems solved by

- replacement of the subproblem matrices by their MIC(0)* incomplete factorizations, see [7] for the details. This variant is denoted as *DiD-IF*.

- inner PCG method with relative accuracy ε_0 and the MIC(0)* incomplete factorization as preconditioner. This variant is denoted as $DiD - II(\varepsilon_0)$.

The DiD was implemented for computations on three processor systems using Fortran 77 and PVM or MPI.

The implemented parallel DiD solvers based on the described DiD preconditioning and using the standard PCG or generalized GPCG[1] methods were tested on the solution of the benchmark problem BM1 on various computer systems:

- SUN symmetric multiprocessor SUN HPC 6500 with Ultra SPARC II/400 processors and 18 GB of the inner memory, codes with MPI,
- LC1 Linux cluster of PC's with the Pentium III/500 processors equipped with 384 MB RAM, Ethernet 100 interconnection, codes with PVM,
- THEA Linux cluster of PC's with the AMD Athlon/1400 processors equipped with 768 MB RAM, Ethernet 100 interconnection, codes with PVM.

SD method	# outer iterations	computing times [s]		
		SUN	LC1	THEA
CG-DiD-IF	83	668	2 653	420
CG-DiD-IF ($\varepsilon_0 = 10^{-1}$)	11	653	918	326
GPCG[1]-DiD-II ($\varepsilon_0 = 10^{-1}$)	8	557	–	274

Table 1: Parallel solution of **BM1** with accuracy $\varepsilon = 10^{-4}$.

The test results can be seen in Table 1. Here, we can see very suitable computing times, moreover, reached on a cheap computer equipment such as a cluster of three standard PC's with Fast Ethernet network. In the case of slower interconnection and/or lack of the inner memory, the inner–outer iteration technique DiD–II outperforms DiD–IF, because it reduces the number of outer iterations, which are responsible for the high communication and data transfer requirements.

6 Domain decomposition (DD)

In this section, we shall concentrate on another type of the space decomposition methods based on a domain decomposition. More details can be found e.g. in [19] and the references therein.

For simplicity, we shall consider the finite element solution of a boundary value problem in a domain Ω, which is divided into finite elements $E \in \mathcal{T}_h$. The decomposition starts from the decomposition of Ω into non-overlapping subdomains $\tilde{\Omega}_k$, which are subsequently extended into overlapping subdomains Ω_k, $\Omega = \bigcup_{k=1}^{p} \Omega_k$. We assume that each $\tilde{\Omega}_k$, Ω_k can be represented as a union of some elements from the global division $\mathcal{T}_h$. Then the division of Ω induces a decomposition of the finite element space V into the subspaces V_k,

$$V_k = \{v \in V : v = 0 \text{ in } \Omega \setminus \Omega_k\}.$$

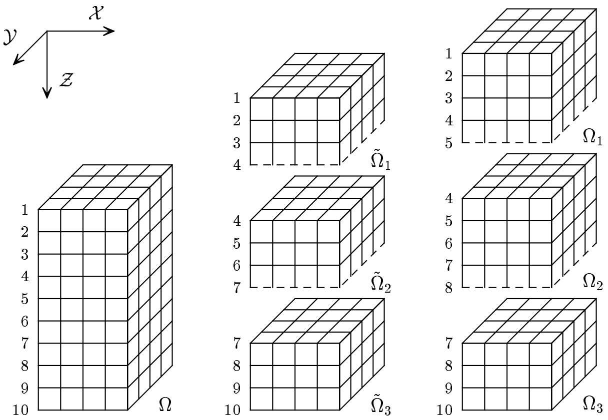

Figure 5: The exploited type of domain partition.

In all spaces $V,\ V_1,\dots,V_p$, we can use the same finite element basis functions. The isomorphism of $V,\ V_1,\dots,V_p$ respectively with $R^n,\ R^{n_1},\dots,R^{n_p}$ allows simple construction of the prolongation represented by a Boolean matrix,

$$I_k = [c_{ij}],\ 1 \le i \le n,\ 1 \le j \le n_k$$

where $c_{ij} = 1$ if the degrees of freedom i and j correspond to the same finite element basis function, otherwise $c_{ij} = 0$.

In our solver, we confine ourselves to a simple partition of the domain Ω into slides, see Figure 5. In this case, one subdomain has at most two neighbouring subdomains and the communications can be restricted to the neighbouring parallel processes and a small part of the degrees of freedom located along the inner boundaries. This makes communications very fast.

The efficiency of the domain decomposition preconditioner improves with the increasing overlap, but deteriorates with the increasing number of subdomains, see e.g. [19] for an explanation. This drawback can be removed and the overall efficiency can be improved by using extended decomposition with an additional subspace V_0 corresponding to a discretization of the global problem with the aid of a coarser finite element division $\mathcal{T}_H$. If $\mathcal{T}_h$ is a refinement of $\mathcal{T}_H$, then the prolongation $I_0 : V_0 \to V$ is simply defined by the inclusion $V_0 \subset V$. If $\mathcal{T}_H$ and $\mathcal{T}_h$ are not nested, then we have to use more complicated interpolation I_0, see [14], which may be relatively costly to create and perform. For this reason, we consider also another choice of V_0, where this space is constructed from V by aggregation. This construction was introduced for multigrid methods in [6], its use for the overlapping Schwarz method is analyzed e.g. in [16].

The aggregation can be described in the following way. Let $\{\phi_i : i = 1,\dots,n\}$ is the FE basis of V_h and let $\{1,\dots,n\} = \bigcup_{k=1}^{N} J_k$, where J_k are disjoint sets. Then the aggregation space can be defined as $V_0 = \operatorname{span}\{\Psi_k : k = 1,\dots,N\}$, where $\Psi_k = \sum_{i \in J_k} \phi_i$. In this case, V_0 is isomorphic to R^N and the prolongation $I_0 : R^N \to R^n$ can be again represented by a Boolean matrix, which allows a cheap construction of the matrix $A_0 = (I_0)^T A I_0$. In our solvers, we used quasi-regular aggregation given by grouping the neighbouring $m_x \times m_y \times$

# subd.	one–level method	two–level method		
		$3 \times 3 \times 3$	$6 \times 6 \times 6$	$9 \times 9 \times 6$
2	92	45	56	61
3	102	47	60	66
4	110	51	64	71
5	117	53	67	75
6	121	55	70	78
7	125	57	72	80
8	128	–	–	–

Table 2: Convergence of CG with the additive DD methods with minimal overlap (one layer of finite elements). Table shows numbers of iterations for getting relative accuracy $\varepsilon = 10^{-4}$. The subdomain problems are replaced by incomplete factorization, the aggregated problem is solved by inner PCG with accuracy $\varepsilon_0 = 10^{-1}$.

# subd.	one-level method	two–level method		
		$3 \times 3 \times 3$	$6 \times 6 \times 6$	$9 \times 9 \times 6$
2	386	267	242	262
3	289	241	175	192
4	242	240	145	160
5	210	236	124	138
6	190	244	115	126
7	170	265	111	117
8	161	–	–	–

Table 3: Computer times in seconds on THEA cluster for the test described above. Computer time includes the time of computations and communications. The times of disk operations, which are performed in the beginning and end of the computations, are not included.

m_z nodes. Note that the computational process corresponding to the coarse grid has to communicate with all other processes. This communication has to be realized in the short aggregated vectors, otherwise, it can fully destroy the benefits from the reduction of the number of iterations.

The performance of DD methods was tested on the BM1 benchmark with computations performed on the THEA cluster described in the previous section. The solvers are again implemented in Fortran 77 and PVM.

The test results, which can be seen in Tables 2 and 3, show efficiency of both one–level and two–level DD methods. In the case of two–level methods, much more care should be devoted to global communication and balancing of the load of processors. For more details, see [20].

7 Composite meshes (CM)

Now, we shall describe the composite mesh finite element method as third example of the space decomposition methods. For simplicity, we shall restrict to the case of one global and one local mesh, which also corresponds to the benchmark BM2. More details about the composite mesh technique can be found e.g. in [11] and the references therein.

Let us again consider a boundary value problem in a domain Ω, which is divided into finite elements $E \in \mathcal{T}_H$; capital H implies that this division is considered as coarse. The division $\mathcal{T}_H$ defines a standard FE space

$$V_H(\Omega) \;=\; \{v \in C(\bar{\Omega}) \cap \mathcal{V} : \, v \mid E \in P_1 \quad \forall E \in \mathcal{T}_H\} \tag{7.6}$$

Note that P_1 means the set of linear polynomials but a generalization to higher order finite elements is straightforward. For elasticity problems, the functions appearing in (7.6) are vector functions.

Let Ω_0 be a part of Ω, where a denser grid is required, $\mathcal{T}_h$ be a division of Ω_0, which defines another FE space

$$V_h(\Omega_0) \;=\; \{v \in C(\bar{\Omega}) \cap \mathcal{V} : \, v \equiv 0 \quad \text{in}\ \Omega \setminus \Omega_0\,,\, v \mid E \in P_1 \quad \forall E \in \mathcal{T}_h\}. \tag{7.7}$$

Then the composite mesh FE space will be

$$V \;=\; V_H(\Omega) + V_h(\Omega_0)\,,$$

i.e. the finite element space is now introduced by means of its decomposition.

In the implementation of the CM method, we shall assume compatibility of the divisions $\mathcal{T}_H$ and $\mathcal{T}_h$, i.e. Ω_0 can be represented as a union of elements from a subset of $\mathcal{T}_H$ and each coarse element from this subset can be decomposed into finer elements. In the space V, we shall consider a nodal basis, but the stiffness matrix corresponding to the CM space will not be formed. Instead, the matrix–vector products with this matrix will be computed in element-by-element manner, cf. [11]. Our implementation also allows that the original coarse grid problem does not match exactly all the details of the solved problem, which are introduced

	DiD–IF			DiD–II, $\varepsilon_0 = 10^{-1}$		
SD preconditioner	# OI	# II–C	# II–F	# OI	# II–C	# II–F
multiplicative	6	160	45	5	23	21
symm. multiplic.	4	203	34	4	40	18
additive	10	221	86	9	36	36

Table 4: Solution of the BM2 benchmark, accuracy $\varepsilon = 10^{-6}$. The symbols #OI, #II-C, #II-F respectively denote the number of outer iterations and the numbers of inner iterations for coarse and fine mesh problems.

	part of the CPU time
coarse grid problem	75 %
fine grid problem	8 %
CM matrix - vector product	13 %
other operations	4 %

Table 5: Composition of the computer CPU times for BM2 benchmark and one CG iteration with multiplicative CM preconditioner.

in the local and composite mesh problems, see also [11] for more details. A theory for the composite mesh decomposition method can be based on the strengthened C.B.S. constant, see [17], [4].

The performance of CM methods was tested on the BM2 benchmark. At present, the testing was performed on IBM RS/6000 43P with only 512MB of inner memory and sequential versions of the programs. For this reason, we shall present here only computer independent information as numbers of iterations, see Table 4. The computer times were substantially longer than the times presented in the previous sections, but the analysis of these times shows that most of the time is spent for solving the subproblems, see Table 5. This indicates that a combination with solvers described in the previous sections should lead to the computing times on THEA cluster below 1000 seconds. The corresponding numerical experiments are in progress. As the parallelization is concerned, it seems that it will be sufficient to use the multiplicative method with parallelization of the subproblem solvers and CM matrix-vector multiplication (computation of the residuals).

8 Additional notes and conclusions

The use of large–scale FE computations for solving geomechanical problems leads to huge computational requirements, which are concentrated mainly into solving large–scale systems of linear equations.

In this paper, we showed that even very large systems can be efficiently solved by means of the conjugate gradient or generalized conjugate gradient method with preconditioning given by space decomposition. This approach is also suitable for parallel computations and very promising results have been obtained from computation on small and cheap clusters of PC's.

The developed methods and solvers are by our opinion suitable for solving many other geotechnical problems related to mining, underground constructions, waste repositories, etc. Moreover, the efficiency of linear system solvers gives a potential for solving nonlinear problems, performing optimization or evaluation the uncertainty included in the input data.

References

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

[2] Axelsson, O.: On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods. Mathematics and Computers in Simulation 50 (1999), 11-30

[3] Axelsson, O., Barker, V.A., Finite Element Solution of Boundary Value Problems. Theory and Computation. Academic Press, Orlando 1984. Reprint in Classics in Applied Mathematics, SIAM, Philadelphia 2001

[4] O. Axelsson, R. Blaheta: Two simple derivations of universal bounds for the C.B.S. inequality constant, Submitted to Appl. Math

[5] Axelsson, O., Gustafsson, I.: Iterative methods for the solution of the Navier's equations of elasticity. Computer Meth. Appl. Mech. Engng. 15 (1978), 241-258

[6] Blaheta, R.: A multilevel method with overcorrection by aggregation for solving discrete elliptic problems. J. Comp. Appl. Math. 24 (1988), 227-239

[7] Blaheta, R.: Displacement Decomposition - Incomplete Factorization Preconditioning Techniques for Linear Elasticity Problems. Num. Lin. Alg. Appl. 1 (1994), 107-128

[8] Blaheta, R.: Identification methods for application in geomechanics and materials science. Report DAM 2000/2, IGAS Ostrava, 2000, 29pp.

[9] Blaheta, R., GPCG - generalized preconditioned CG method and its use with nonlinear and nonsymmetric displacement decomposition preconditioners. Report DAM 2001/2, Institute of Geonics AS CR, Ostrava, 2001 . Submitted to Numer. Linear Alg. Appl.

[10] Blaheta, R., Jakl, O., Starý, J., A parallel CG solver for FE analysis of 3D problems in geomechanics, in GEOMECHANICS'96, A.A. Balkema, Rotterdam 1997.

[11] Blaheta, R., Byczanski, P., Kohut, R., Composite grid finite element method: Implementation and iterative solution with inexact subproblems, To appear in Appl. Math. 2002

[12] Blaheta, R., Byczanski, P., Kohut, R., Kolcun A., Rakowski, Z. and Hájek, A., Large Scale Mathematical Modelling of the effects of mining with GEM3/HPC software. Czech-Polish Geomechanical Symposium, Documenta Geonica 1999

[13] Braess, D., Finite Elements. Second edition. Cambridge University Press, Cambridge 2001

[14] Cai, Xiao-Chuan, The use of pointwise interpolation in domain decomposition methods with non-nested meshes, SIAM J. Sci. Comput. 16(1995), 250-256

[15] Johnson, C., Numerical Solution of Partial Differential Equations by the Finite Element Method, Cambridge University Press, Cambridge 1987

[16] Jenkins, E.W., Kelley, C.T., Miller, C.T., Kees, C.E.: An aggregation-based domain decomposition preconditionerfor groundwater flow. SIAM J. Sci. Comput. 23 (2001), 430-441

[17] Mandel, J., McCormick, S., Iterative Solution of Elliptic Equations with Refinement: The Two-Level Case, in Domain Decomposition Methods, edited by Tony F. Chan, Roland Glowinski, Jacques Periaux, Olof B. Widlund, SIAM, Philadelphia, 1989, pp. 81–92.

[18] Notay, Y., Flexible conjugate gradients, SIAM J. Sci. Comp., 22(2000), 1444-1460

[19] Smith, B.F., Bjørstad, P.E., Gropp, W.D.: Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, 1996

[20] Starý, J., Parallel algorithms for the solution of elasticity problems, thesis, TU Ostrava and Institite of Geonics AS CR Ostrava, in progress

[21] Xu, J.: The method of subspace corrections. J. Comp. Appl. Math. 128 (2001), 335-362