AN APPLICATION OF PARALLEL HIGH PERFORMANCE COMPUTING IN GEOMECHANICS

R. Blaheta, O. Jakl and J. Starý

Institute of Geonics AS CR,
Studentská 1768, 708 00 Ostrava–Poruba

Submitted to the Proceedings of 3μ Conference
October 1998

Abstract: The paper* describes new achievements in application of parallel high performance computing for the finite element modelling of large scale 3D problems of geomechanics. We show that solvers have the crucial role in these applications and focus our attention on efficient parallel implementation of iterative solvers based on the conjugate gradient method preconditioned by the displacement decomposition - incomplete factorization technique.

1. Introduction

This paper describes new achievements in application of parallel high performance computing in geomechanics and can be viewed as a continuation of the paper [9]. The considered application is the assessment of the geomechanical effects of mining in the uranium deposit Dolní Rožínka. The aims of the computations are the following (see [10], [11]):

- to assess the possibility of rockbursts during the mining on the new levels under 1000 meters below the surface,
- to compare stress fields which are induced by the use of two different mining methods – underhand or overhand stoping,
- to analyse stress changes during the forthcoming mining.

The above questions arose during preparation of the plan of exploitation of the deposit Dolní Rožínka for the years 1999–2002 and answers to these questions are also required by the Czech Mining Authority which has to approve this exploitation plan.

The mathematical modelling for answering the formulated questions leads to repeated finite element analyses with nearly four million degrees of freedom for each of them. This big problem size is further increased by the need of local analysis for comparing two mining technologies of underhand and overhand stoping. More detailed description of the solved problem will be given in Section 2 or can be found in [9].

For the numerical solution of the arising large scale linear systems, we exploited several iterative solvers based on the conjugate gradient method with the preconditioning by the displacement decomposition and incomplete factorization. We used also parallel implementations of these iterative methods on a cluster of workstations or distributed memory parallel computer like IBM SP. The development and performance of these solvers are described in Section 3.

Some other aspects of the performed mathematical modelling, namely the local analysis (submodelling) with the finite element method on composite grids and assessment of accuracy of the discretization by aposteriori error estimators are touched in contributions [13] and [14].

Note that more information about this and a related work done within the project of the European Commission COPERNICUS 940820 "High performance computing in geosciences" can be found on the web pages http://www-math.sci.kun.nl/math/Copernicus.

*The presented work was supported by Grant of the European Commission COPERNICUS 940820 and Grant OK - 157 of the Ministry of Education CR.

2. Mathematical model

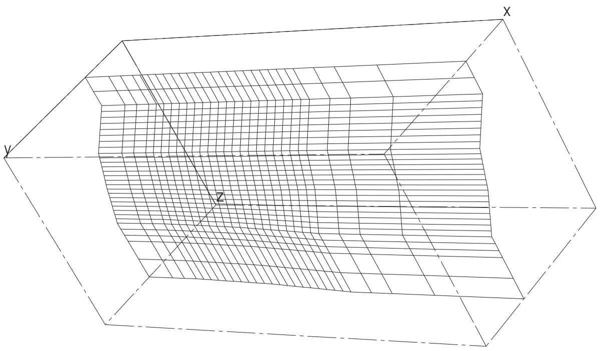
The details about the construction of mathematical model for the assessment of the effects of mining at the deposit Dolní Rožínka were described at [9]. Therefore, here we shall only repeat the basic facts for the sake of completeness.

For the construction of the mathematical model, we consider the domain of 1430x550x600 meters the top side of which is located in the depth of 700 meters under the surface, see Figure 1. Three uranium ore veins are located within this domain.

The boundary conditions are selected either as the pure Neumann–type boundary conditions, which simulate the action of the surrounding rocks in the pre–mining stress state, or the pure Dirichlet–type boundary conditions given by pre–mining displacements which must be computed in advance. For rocks and the goaf material in the mined out and caved volumes we assume linear elastic behaviour.

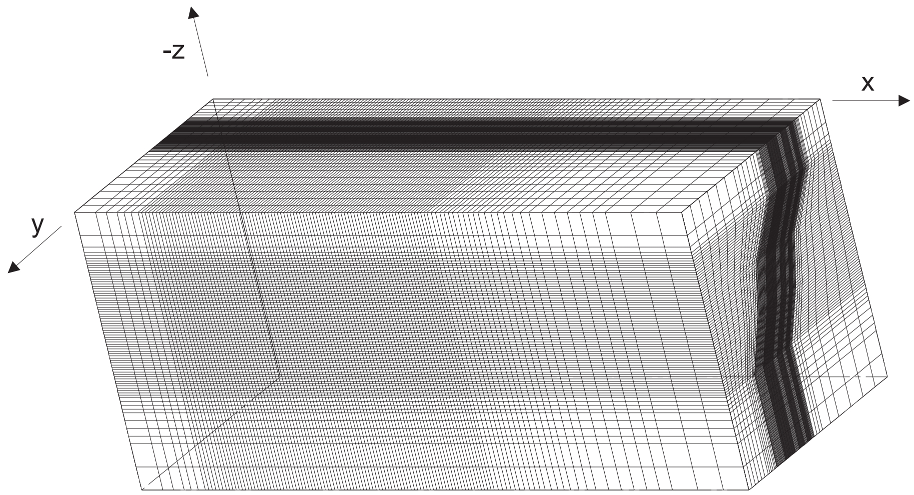
The modelling considers four stages of successive mining in the selected domain. These four stages are modelled either as a sequence of four Neumann's boundary value problems (Neumann's modelling sequence - NMS) or as a sequence of one auxiliary Neumann's problem (computation of the pre–mining displacements) and four Dirichlet's boundary value problems (Dirichlet's modelling sequence - DMS).

Figure 1



The discretization is given by division of the problem domain in bricks, see Figure 2, which are then divided into six tetrahedra each. This leads to the finite element systems with 3 873 thousands of unknowns. Note that these systems are singular in the case of the Neumann's modelling sequence and regular with exception of the first auxiliary step in the case of the Dirichlet's modelling sequence. Moreover, the singular systems can be slightly inconsistent due to some imbalance given by different unit weight of material in the mined out parts and due to rounding-off errors. This means that for the singular systems we seek for the generalized solutions. The nonuniqueness of solution of the singular systems is afterwards eliminated because for the modelling we need only the stresses in the considered domain.

Figure 2



3. Parallel iterative solution of the FE systems

The mathematical modelling described in the previous Section leads to the necessity of solving large linear systems

$$Au = b,$$

of nearly four million unknowns. To this end, we tested the use of several iterative methods and evaluate their performance on two computers:

1. IBM RS 6000 3CT at the Institute of Geonics, Ostrava
   Technical data: processor Power 2/66.7 MHz, 128 MB of main memory,

2. IBM SP at the Technical University of Ostrava
   Technical data: 8 processors Power 2/66.7 MHz, (Thin2), ATM interconnection 20 MB/sec, HPS–USS interconnection 36 MB/ sec (not available under PVM).

Note that the iterative methods were implemented under XL FORTRAN and PVM. The matrix was stored on disk and read from it in each iteration. For an MPI implementation, see [12].

Solver PCG–1

We started with the PCG–1 solver [6] based on the preconditioned conjugate gradient method with preconditioning by displacement decomposition and incomplete factorization, see [3], [5] and [2]. The construction of the preconditioner can be simply described when we use the separated displacement component ordering which leads to the partition

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12}^T & A_{22} & A_{23} \\ A_{13}^T & A_{23}^T & A_{33} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A_u & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}$$

where u_i contains the unknowns representing nodal displacements in the direction of the axis x_i.

The preconditioner C is constructed by MIC(0)* modified incomplete factorization of the diagonal blocks of $A_{ii}, i.e.$

$$A_{ii} \sim C_{ii} = [X_{ii} - L_{ii}]\, X_{ii}^{-1}\, [X_{ii} - L_{ii}]^T$$

where $-L_{ii}$ is the lower triangular part of A_{ii} and X_{ii} is a diagonal matrix determined by the condition rowsums(C_{ii}) = rowsums$(A_{ii}$ + diagonal perturbation), details can be found in [5].

Table 1 shows the computing times (elapsed times) on IBM RS/6000 3CT computer. The solution is performed in steps corresponding to the selected modelling stages, the solution of one stage serves as an initial guess to the next stage. Accuracy of iterations given by the condition $\| b - Au^i \| / \| b \| < \varepsilon = 10^{-3}$. We can se that

- the use of the Dirichlet's modelling sequence gives speed up of 1.6,
- the times which are spent by solvers during the Neumann's and Dirichlet's modelling sequences gives respectively 93% and 83% of the overall computing time of the finite element analysis (FEA).

	S0	S1	S2	S3	S4	$\sum$	total FEA
Neumann	–:–	25:54	05:40	04:28	09:14	45:16	48:23
Dirichlet	25:45	01:03	00:36	00:36	01:02	28:17	32:07

Table 1. Speed up **1.6** for the Dirichlet's Modelling Sequence.

Solver PCG–1P

To reduce the computer times of the solver, we tried to parallelize the computation by splitting the problems according to the displacement decomposition. Such decomposition is very natural and means that we use three processors with their local memories and distribute the matrix as well as the vectors of unknowns, rhs and the iteration vectors among these processors, see the following scheme (also [7] and [8]).

Scheme: Parallel displacement decomposition algorithm

		Processor1 - Memory1	Processor2 - Memory2	Processor3 - Memory3
A, u, v, r, w		A_{11}, A_{21}, A_{31} u_1, v_1, r_1, w_1	A_{12}, A_{22}, A_{32} u_2, v_2, r_2, w_2	A_{13}, A_{23}, A_{33} u_3, v_3, r_3, w_3
$w = Av$	P	$w_{11} = A_{11}v_1$ $w_{21} = A_{21}v_1$ $w_{31} = A_{31}v_1$	$w_{12} = A_{12}v_2$ $w_{22} = A_{22}v_2$ $w_{32} = A_{32}v_2$	$w_{13} = A_{13}v_3$ $w_{23} = A_{23}v_3$ $w_{33} = A_{33}v_3$
	C1	send w_{21}, w_{31} receive w_{12}, w_{13} $w_1 = w_{11} + w_{12} + w_{13}$	send w_{12}, w_{32} receive w_{21}, w_{23} $w_2 = w_{21} + w_{22} + w_{23}$	send w_{13}, w_{23} receive w_{31}, w_{32} $w_3 = w_{31} + w_{32} + w_{33}$
$s = \langle w, v \rangle$	P	$s_1 = \langle w_1, w_1 \rangle$	$s_2 = \langle w_2, w_2 \rangle$	$s_3 = \langle w_3, w_3 \rangle$
  $\alpha = s/s_0$	C2	receive s_2, s_3 $s = s_1 + s_2 + s_3$ $\alpha = s/s_0$ send α	send s_2 receive α	send s_3 receive α
$u = u + \alpha v$ $r = r - \alpha w$ $w = B(r)$ $s_1 = \langle r, w \rangle$	P	$u_1 = u_1 + \alpha v_1$ $r_1 = r_1 - \alpha w_1$ $w_1 = B(r_1)$ $s_{11} = \langle r_1, w_1 \rangle$	$u_2 = u_2 + \alpha v_2$ $r_2 = r_2 - \alpha w_2$ $w_2 = B(r_2)$ $s_{22} = \langle r_2, w_2 \rangle$	$u_3 = u_3 + \alpha v_3$ $r_3 = r_3 - \alpha w_3$ $w_3 = B(r_3)$ $s_{33} = \langle r_3, w_3 \rangle$
  $\beta = s_1/s_0$ $s_0 = s_1$	C3	receive s_{22}, s_{33} $s_1 = s_{11} + s_{22} + s_{33}$ $\beta = s_1/s_0$ $s_0 = s_1$ send β	send s_{22} receive β	send s_{33} receive β
$v = w + \beta v$	P	$v_1 = w_1 + \beta v_1$	$v_2 = w_2 + \beta v_2$	$v_3 = w_3 + \beta v_3$

Implementing this idea on three processors of the IBM SP system with the ATM interconnection lead us to the computing times according Table 2. From it we can see the speed up of 2.12 and 2.15 for the Neumann's and Dirichlet's modelling sequence respectively.

	S0	S1	S2	S3	S4	$\sum$
Neumann	–:–	11:55	02:42	02:36	04:46	21:20
Dirichlet	12:00	00:29	00:16	00:15	00:30	13:30

Table 2. Speed up **2.12 / 2.15** by using 3 processors.

Solver PCG–1P–V

The speed up of solver PCG–1P is less then the theoretical speed up of three because of the communication and synchronization which is necessary to do within each iteration. To reduce the amount of communication, we can introduce the solver with variable preconditioning, see the following algorithm.

Initialization: given u_0,
compute $r^0 = b - Au^0$, $w^0 = B(r^0)$, $v^0 = w^0$
Iterations:
for $k = 0, 1, \ldots$ **until** $\|r^k\| \le \varepsilon \|b\|$ **do**

$$\begin{aligned}
\alpha_k &= < r^k, w^k > / < Av^k, v^k > \\
u^{k+1} &= u^k + \alpha_k v^k \\
r^{k+1} &= r^k - \alpha_k A v^k \\
w^{k+1} &= B(r^{k+1}) \\
\beta_k &= < r^{k+1}, w^{k+1} > / < r^k, w^k > \\
v^{k+1} &= w^{k+1} + \beta_k v^k
\end{aligned}$$

endfor

The application of preconditioner $w = B(r)$ is now defined as an approximate solution of the system $\tilde{A}w = r$ by the inner conjugate gradient method preconditioned by the same incomplete factorization of the diagonal blocks A_{ii}. The solution of $\tilde{A}w = r$ again naturally splits into three independent tasks, the initial guess for this solution is given by the zero vector and the accuracy of inner iterations can be chosen relatively low without influencing much the convergence of the outer iterative process, e.g. we shall use $\varepsilon_0 = 10^{-1}$.

This method can be again solved on three processor computer equipment with a reduced communication which takes place now only in outer iterations.

Note 1. Flexible CG (see [18]). To preserve A-orthogonality of the search directions v^{k+1} and v^k, it is necessary to replace the above expression for β_k by

$\beta_k =< r^{k+1}, w^{k+1} > - < r^k, w^{k+1} > / < r^k, w^k >$.

This may lead to a little improvement of the convergence of outer iterations.

Note 2. To restore orthogonality of all directions v^k, it is necessary to use the concept of generalized CG, see e.g. [1].

Solver PCG–1P–Vs

In the performed finite element analysis, we had to solve both singular and regular linear systems. The performance of PCG–1P–V solver with variable preconditioning was very nice for regular problems, see also [9], but this method diverged in the case of singular (slightly inconsistent) systems. This drawback was removed by using the projection P to the theoretical rank of the matrix A, $R(A) = N(A)^{\perp}$ due to symmetry of A. This projection is constructed from the knowledge of basis of the null space N(A). The basis of $N(A)$ is created from the vectors which represent six independent rigid body translations and rotations.

$$Pv = v - \sum_{i1}^{6} \alpha_i w_i$$

where the coefficients α_i are determined by the conditions $(Pv, w_j) = 0$ for $j = 1, \ldots, 6$.

The preconditioned conjugate gradient method with variable preconditioning and stabilization can be then written as follows:

for $k = 0, 1, \ldots$ **until** $\|r^k\| \leq \varepsilon \|b\|$ **do**

$$\begin{array}{rcl} \alpha_k & = & < r^k, w^k > / < Av^k, v^k > \\ u^{k+1} & = & u^k + \alpha_k v^k \\ r^{k+1} & = & r^k - \alpha_k A v^k \\ & & \\ r^{k+1} & = & P(r^{k+1}) \\ & & \\ w^{k+1} & = & B(r^{k+1}) \\ \beta_k & = & < r^{k+1}, w^{k+1} > / < r^k, w^k > \\ v^{k+1} & = & w^{k+1} + \beta_k v^k \end{array}$$

endfor

The performance of PCG–1P–V a PCG–1P–Vs solvers for the Dirichlet's modelling sequence can be seen from the times in Table 3. Note that these results were obtained for anisotropic pre–mining stress and accuracy $\varepsilon = 10^{-3}$, $\varepsilon_0 = 10^{-1}$. The computations were performed on IBM SP with the ATM interconnection of the computing nodes.

	S0	S1	S2	S3	S4	$\sum$
PCG-A-V-Ps	02:12	00:19	00:12	00:12	00:14	03:09

Table 3. Speed up about **10 !**

The time for the whole modelling sequence represents speed up of about 10 in comparison with the initial results for PCG–1 solver and the Dirichlets's modelling sequence. In comparison with the very first results for the Neumann's modelling sequence we obtain even higher speed up of about 14. The superlinear speed up seems to be caused by smaller memory demands (less paging), more efficient data transfers and more efficient use of caches when lot of the effort was concentrated to the solution of smaller subproblems corresponding to the diagonal blocks A_{11}, A_{22}, A_{33}. The importance of the caches for efficiency of the codes are now more intensively investigated, see e.g. [19].

Further improvements

In the case of variable preconditioning, we are solving simultaneously three systems with the matrices A_{11}, A_{22}, A_{33} which correspond to three different differential operators and (in our case) the same boundary conditions. This implies different computational load on the three exploited processors. This drawback can be improved by use of averaged blocks $A_{11}+A_{22}+A_{33}$. The averaged blocks should give a similar preconditioning effect and also improve stability of the incomplete factorization, see [5] and [2].

So far we have used the simplest incomplete factorization of the blocks A_{ii} . Use of more sophisticated block incomplete factorization described in [15] and [16] was tested within the COPERNICUS 940820 project. For the regular problems and serial computing with the whole matrix in the main memory on SUN Ultra-Enterprise (170 MHz, 1 GB RAM) the speed up of 2.7 and 4.4 was observed in comparison of the PCG–1 solver with solvers which use respectively fixed and variable preconditioners constructed by the block incomplete factorization, see [4].

4. Conclusions

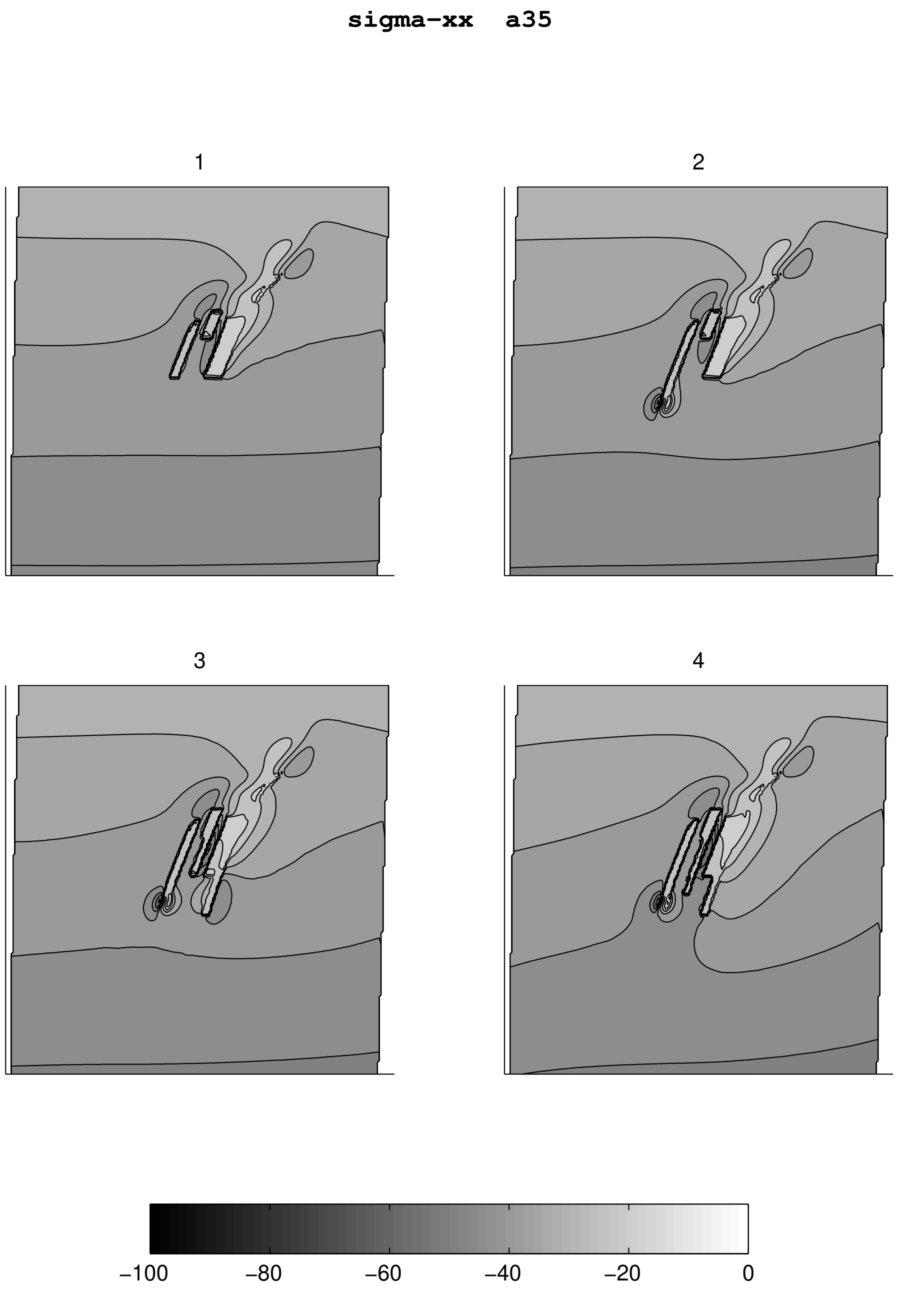
It was shown that the application of the finite element analysis for the modelling of the effect of mining at the uranium mine Dolní Rožínka leads to the solution of very large linear systems of nearly four million of equations. Moreover, some of the systems are singular.

This motivated an effort for finding appropriate solvers in correspondence with the available parallel computer equipment. To this respect, we consider several variants of the displacement decomposition – incomplete factorization technique which at the end enables to reach the speed up of about 14 in comparison with the basic serial solver and the basic formulation of the modelling sequence. We also shortly mention some direction of further improvements as the block incomplete factorization or AMLI methods, c.f. [4].

Note that the performed modelling stimulated also research in other areas as iterative composite grid methods, visualization of the results of computation and aposteriori analysis of the discretization error. An example of the exploited visualization in a chosen cross–section can be seen from the following Figure 3.

Finally emphasize that the performed modelling was able to give answers to the basic engineering questions which was mentioned in the introduction. The engineering conclusion from the modelling can be found in [11].

Figure 3



References

[1] O. Axelsson: Iterative solution methods, Cambridge Univ. Press, Cambridge 1994

[2] O. Axelsson: On iterative solvers in structural mechanics. In: Proceedings of PRISM'97, O. Axelsson, M. Neytcheva and B. Polman, eds. KUN Nijmegen 1997.

[3] O. Axelsson and I. Gustafsson: Iterative methods for the solution of the Navier equations of elasticity, Comp. Meth. Appl. Mech. Engrg. 15(1978), 241-258

[4] R. Beauwens and Y. Notay. Efficient block preconditioners for elasticity problems. Technical Report GANMN 98-01, Université Libre de Bruxelles, Brussels, Belgium, 1998.

[5] R. Blaheta: Displacement decomposition – incomplete factorization preconditioning techniques for linear elasticity problems, Numer. Lin. Alg. Appl. 1(1994), 107-126

[6] R. Blaheta, R. Kohut: PCG–1 iterative solver, Report DAM, Institute of Geonics AS CR, Ostrava, 1995

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

[8] R. Blaheta, O. Jakl and J. Starý, PVM-implementation of the PCG method with Displacement Decomposition. The Fourth European PVM-MPI Users' Group Meeting in Krakow. In, M. Bubak, J. Dongarra, J. Wasniewski (eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface, Lecture Notes in Computer Science 1332, Springer Verlag 1997.

[9] R. Blaheta, O. Jakl, R. Kohut and A. Kolcun: An application of large scale mathematical modelling in geomechanics, In: Proc. 6th Seminar Contemporary mathematical methods in engineering, J. Doležalová ed., Technical University of Ostrava, 1997

[10] A. Hájek, Z. Rakowski: Mathematical models for development of mining technology, COPERNICUS 940820 Project Report, Ostrava, 1995

[11] A. Hájek and Z. Rakowski, Role of large scale mathematical models in development of new mining technology. HIPERGEOS Report, GEAM D. Rožínka, March 1998.

[12] O. Jakl and J. Starý: Parallel computations in the MPI environment, these proceedings

[13] R. Kohut: Solving problems on composite grids, these proceedings

[14] A. Kolcun: Aposteriori error estimates for 3D elasticity problems, these proceedings

[15] Magolu Monga-made, Y. Notay: Dynamically relaxed block incomplete factorizations, TR–985, Department of Mathematics, University of Utrecht, 1996.

[16] Magolu Monga-made, B. Polman: Theoretical comparison of pointwise, linewise and planewise preconditioners for solving three–dimensional problems, TR–9609, Department of Mathematics, University of Nijmegen.

[17] Y. Notay: Solving positive (semi) definite linear systems by preconditioned iterative methods, in Preconditioned Conjugate Gradient Methods, O. Axelsson, L. Yu. Kototilina (eds.), LNM 1457, Springer–Verlag, Berlin 1990.

[18] Y. Notay: Benchmarking for HIPERGEOS II, TR – GANMN 9802, University of Brussels, 1997

[19] Ch. Weiss, H. Hellwagner, U. Rude and L. Stals, Data locality optimizations to improve efficiency of multigrid methods, Report University of Augsburg, 1998