



High Performance Computing Applications

High Performance Computing in Micromechanics with an Application

Radim Blaheta¹, Ivan Georgiev², Krassimir Georgiev², Ondrej Jakl¹, Roman Kohut¹, Svetozar Margenov², Jiri Stary¹

¹*Institute of Geonics, Czech Academy of Sciences, Ostrava, Czech Republic*

²*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*

E-mails: jiri.stary@ugn.cas.cz georgiev@parallel.bas.bg

Abstract: *High Performance Computing (HPC) is required for many important applications in chemistry, computational fluid dynamics, etc., see, e.g., an overview in [1]. In this paper we shortly describe an application (a multiscale material design problem) that requires HPC for several reasons. The problem of interest is analysis of the fiber-reinforced concrete and we focus on modelling of stiffness through numerical homogenization and computing local material properties by inverse analysis. Both problems require a repeated solution of large-scale finite element problems up to 200 million degrees of freedom and therefore the importance of HPC computing is evident.*

Keywords: *Analysis of fiber-reinforced concrete, homogenization, identification of parameters, parallelizable solver, additive Schwarz method, two-level parallelization.*

1. Introduction

The paper concerns linear micromechanics exploiting Computed Tomography (CT) scans for determination of microstructure and numerical homogenization with a focus on a specific application – analysis of fiber-reinforced concrete. The analysis includes an identification problem and stochastic uncertainty, which bring new dimensions and enhance the need for fast solvers and ultrascale computations. Fiber-reinforced concrete with steel fibers has many applications in civil and geotechnical engineering. It is less expensive than hand-tied rebar, while still increasing the tensile strength many times. The shape, dimension, and length (standard 1 mm diameter, 45 mm length) of the fiber together with fiber volume amount and distribution are important parameters influencing the tensile strength of concrete.

The analysis includes an assessment of tensile stiffness for several samples of fiber-reinforced concrete which differ in amount and distribution of fibers. These samples are scanned by CT and analyzed with provided elastic parameters for steel fibers and concrete matrix. The detailed scan of a sample leads to solving of elastic problems with about 200 million degrees of freedom.

If the global response of the samples can be tested on a loading frame, then the output allows solving an inverse material identification procedure to determine the elastic properties of the concrete matrix. In this way, we can both determine the properties of concrete matrix, which can also be variable to some extent, as well as assess if some discrepancy can be explained by imperfect bonding of fibers.

It is also possible not only to investigate selected physical samples of the fiber-reinforced concrete but also to do stochastic analysis with repeated generation of stochastic microstructure, see, e.g., [6, 7].

2. Homogenization and identification of parameters

The numerical homogenization starts with solving the elasticity problem on the domain Ω with given microstructure. The solution is possibly repeated for different loadings by imposed boundary conditions. In an abstract way, we denote the loading conditions by L or in the case of multiple loading by $L^{(k)}$. The stress and strain tensors $\sigma^{(k)}$ and $\varepsilon^{(k)}$ are averaged over Ω and the homogenized elasticity tensor $\bar{C} \in R_{\text{sym}}^{6 \times 6 \times 6 \times 6}$, $C = [c_{ijkl}]$, $c_{ijkl} = c_{jikl} = c_{klij}$, is determined as a (generalized) solution of the system

$$\bar{C}\bar{\varepsilon}^{(k)} = \bar{\sigma}^{(k)}, \quad \bar{\sigma}^{(k)} = |\Omega|^{-1} \int_{\Omega} \sigma^{(k)} d\Omega, \quad \bar{\varepsilon}^{(k)} = |\Omega|^{-1} \int_{\Omega} \varepsilon^{(k)} d\Omega.$$

Assuming isotropy of the homogenized elasticity tensor, one loading is sufficient for getting elasticity constants. If $\varepsilon = \varepsilon_{\text{vol}} + \varepsilon_{\text{dev}}$ is the decomposition of $\varepsilon \in R_{\text{sym}}^{6 \times 6}$ into the volumetric and deviatoric parts and $\|\cdot\|$ is the Frobenius norm, then the bulk and shear moduli can be determined as

$$K = \frac{1}{3} \|\bar{\sigma}_{\text{vol}}\| / \|\bar{\varepsilon}_{\text{vol}}\|, \quad G = \frac{1}{2} \|\bar{\sigma}_{\text{dev}}\| / \|\bar{\varepsilon}_{\text{dev}}\|.$$

For the parameter identification, we assume that some local material properties are unknown, e.g. that the concrete matrix is described by unknown parameters $p = (K_c, G_c)$, where K_c and G_c are unknown bulk and shear parameters of concrete. More generally, Ω can be split into subdomains with different unknown elastic moduli of concrete. Then the parameters are found by minimization of a proper objective function J over a set of admissible parameters, see, e.g., [8].

The construction of the objective function can be as follows:

$$J(p) = \sum_k \left[w_{1k} \left\| \bar{\varepsilon}^{(k)}(p) - \bar{\varepsilon}_{\text{test}}^{(k)} \right\|^2 + w_{2k} \left\| \bar{\sigma}^{(k)}(p) - \bar{\sigma}_{\text{test}}^{(k)} \right\|^2 \right],$$

where $\bar{\sigma}^{(k)}(p)$ and $\bar{\varepsilon}^{(k)}(p)$ are averaged stresses and strains computed by solving the boundary value problem in Ω with given microstructure, local material properties involving parameters from p and the loading $L^{(k)}$. This boundary value problem represents a physical test on the specimen Ω . The test configuration is such that in the case of homogeneity of Ω , the problem has a solution with unique and constant stress $\bar{\sigma}_{\text{test}}^{(k)}$ and strain $\bar{\varepsilon}_{\text{test}}^{(k)}$, which can be determined from measurements. The weights w_{ik} can be determined by numerical experiments or simply set to be equal to $w_{ik} = 1$.

The optimization is performed by a suitable method; we already successfully tested the Nelder-Mead and Gauss-Newton methods.

More details on the exploited homogenization and identification methods can be found in [4, 5].

3. Additive Schwarz solver with two-level parallelization

A crucial component of the homogenization and identification procedures is the solver for boundary value problems of elasticity. We assume a finite element discretization leading to algebraic systems of type $Au = b$ or $A(p)u(p) = b$, where the latter indicates dependence on some local material parameters. The system can be solved by the Preconditioned Conjugate Gradient (PCG) method with one-level Additive Schwarz (AS) preconditioner B_{AS1} , and mostly by its extended two-level version B_{AS2} ,

$$B_{AS1} = \sum_{k=1}^N R_k^T \tilde{A}_k^{-1} R_k, \quad B_{AS2} = B_{AS1} + R_0^T \tilde{A}_0^{-1} R_0.$$

Here R_k is a restriction defined by subdomain Ω_k or algebraically by overlapping decomposition of the solution vector $u \in R^n$, \tilde{A}_k is an approximation to $A_k = R_k A R_k^T$. In our case \tilde{A}_k is a displacement decomposition – incomplete factorization of A_k . The one-level AS preconditioner is not scalable, the number of iterations increases with N , although this growth is partially compensated by the fact that \tilde{A}_k becomes a better approximation to A_k . It fits the algebraic form of the Schwarz methods if $R_0 \in R^{n_0 \times n}$ is a Boolean matrix, which defines aggregation of degrees of freedom, i.e., each row of R_0 defines one aggregate by unities in this row. On the other hand, each degree of freedom corresponds to just one aggregate,

i.e., there is precisely one unity in each column of R_0 . More details about this setting can be found, e.g., in [3].

In the case of computing on a massively parallel computer like Salomon [9], it is possible to exploit hundreds of processors, which makes the local problems A_k small even for large scale matrices A . It is difficult to keep balance of times for solving the local problems A_k and the coarse global one A_0 . For this reason, parallel inner CG iterations for the solution of problem A_0 were proposed and the algorithm employs two levels of parallelization.

4. Numerical experiments

Our numerical experiments present five real samples of fiber-reinforced concrete, each one of cubic shape and size 35 mm. Their microstructure is taken from industrial CT scanning performed at the CT lab of the Institute of Geonics of the CAS. The digital models arose from meshes of approx. $1400 \times 1400 \times 1400$ voxels, which were further trimmed to $1000 \times 1000 \times 1000$ voxels due to surface damage or irregular sides of the samples.

Consequent computational models use smaller REpresentative Volumes (REV) and standard linear tetrahedral finite elements. The size of each REV is $400 \times 400 \times 400$ voxels for homogenization experiments or $100 \times 100 \times 100$ voxels for tests related to material identification, respectively. Accordingly, the model leads to a (repeated) solution of the resulting linear system in size of about 193 or 3 million degrees of freedom. Main characteristics of each REV are summarized in Table 1.

Table 1. Characteristics of REV for each sample of reinforced concrete. Samples differ in the volumes of steel fibers as well as voids. The size of fibers: Length 6 mm, diameter 0.12 mm

Sample	Steel fibers, kg/m ³	Volume, steel, %	Volume, voids, %
0	0	0.00	1.55
2	50	0.92	1.22
3	100	1.82	0.75
4	150	2.57	0.71
5	200	2.11	1.83

The properties of materials involved in mathematical modelling are listed in Table 2. Voids (air bubbles in the microstructure) bring a kind of singularity caused by the finite elements weakly hanged in void space. They are replaced with a very weak elastic material. The convergence of the applied PCG method is then smoother and faster.

Table 2. List of involved materials and their properties.
Young's modulus E and Poisson's ratio ν

Material	E , GPa	ν
Concrete	19	0.2
Steel	200	0.3
Voids	0.01	0.1

The arising large-scale systems of linear equations are processed by parallel solvers based on the PCG method, with stabilization in the singular case [11]. Most of the computations were performed on the SGI cluster Salomon [9], only the results of the performance comparison study and scalability tests were obtained with the aid of other available High Performance Computing (HPC) computers:

- **Salomon** – IT4Innovations, Ostrava: SGI cluster, currently on 78 place in Top500 (June 2017), consists of 24192 cores and 129 TB of memory in total and with the theoretical peak performance over 2 PFlop/s. Most of its compute nodes are equipped with two 12-core processors Intel Xeon E5-2680 V3 and 128 GB of memory.

- **Anselm** – IT4Innovations, Ostrava: Bull cluster with 209 compute nodes, in total having 3344 cores with 15 TB of memory and giving over 94 TFlop/s of theoretical peak performance. The used compute nodes were Bullx B510 blade servers with 2×8-core processors Intel Sandy Bridge E5-2665 and 64 GB of memory.

- **Avitohol** – Institute of Information and Communication Technologies of the BAS, Sofia: HP cluster based on 150 SL230S GEN8 servers with 2×8-core processors Intel Xeon E5-2650 V2. In total, it consists of 20700 cores and 9600 GB of memory and gives 412 TFlop/s of theoretical peak performance.

- **Enna** – Institute of Geonics of the CAS, Ostrava: Symmetric multiprocessor Supermicro 5086B-TRF equipped with 8×8-core processors Intel Xeon E7-8837 and 512 GB of shared memory.

On all platforms, source codes of parallel solvers were compiled by Intel compilers (Intel Parallel Studio XE, see Table 6 for the versions used). The applied compilation flags Bincluded mainly – O3 specifying the level of code optimization, -xHost generating instructions for the highest instruction set available on the target processor (not supported and applied on Enna), -shared-intel linking Intel-provided libraries dynamically, and -mcmmodel medium telling the compiler to use a specific memory model to generate code and store data.

Table 3 presents results of numerical homogenization applying pure Dirichlet and pure Neumann Boundary Conditions (BC). The choice of BC sets a configuration of homogenization procedure, which simulates an appropriate laboratory test under uniaxial loading. The Dirichlet BC prescribe some non-zero displacement on the top side in the direction of uniaxial loading, the other sides have zero normal displacements. The Neumann BC enter opposite non-zero forces on the top and bottom sides in the direction of uniaxial loading, the other sides have zero normal forces. The use of pure Dirichlet and pure Neumann BC allow us to get upper and lower bounds for the upscaled elasticity tensor, see, e.g., [4].

Table 3. Results obtained by numerical homogenization applying Dirichlet and Neumann BC. Values of material parameters for different directions (X, Y, Z) of uniaxial loading and averaged (below)

Sample	Dirichlet BC						Neumann BC					
	E , GPa			ν			E , GPa			ν		
0	18.365	18.370	18.407	0.199	0.199	0.199	18.307	18.305	18.216	0.199	0.199	0.197
	18.381			0.199			18.276			0.198		
2	19.050	18.960	19.063	0.200	0.201	0.200	18.692	18.822	18.798	0.197	0.199	0.199
	19.024			0.200			18.771			0.198		
3	20.015	19.621	19.768	0.200	0.202	0.201	19.912	19.599	19.716	0.203	0.199	0.201
	19.801			0.201			19.742			0.201		
4	20.865	19.977	19.960	0.198	0.203	0.203	20.613	19.948	19.435	0.204	0.199	0.195
	20.267			0.201			19.999			0.199		
5	19.345	19.508	19.715	0.202	0.202	0.201	18.297	17.193	19.213	0.190	0.178	0.199
	19.523			0.202			18.234			0.189		

Due to irregular placement of steel fibers as well as voids in the microstructure, the results document anisotropy of tested material, when the values of material properties strongly vary for different directions of loading, e.g., the Young's modulus E (Sample 4, Neumann BC) in Table 3 varies about more than 1 GPa. However, as expected and consistent with theory, their averaged values follow the increase of volume of steel fibers in concrete.

The corresponding values for the pure Dirichlet and pure Neumann BC give quite close bounds for real material properties. However we observe that these bounds grow away with the increasing volume of voids in the microstructure, moreover when the voids are closer to the border of the studied domain and the pure Neumann BC are applied, see the values for the Sample 5. Comparing with the others, Sample 5 contains also another abnormality. Although this sample should contain the most of steel fibers according to Table 1, the real volume of steel in REV is not the biggest one. Moreover, REV of this sample overcomes the others in the volume of the void space in its microstructure.

The previous tests were related to the direct problem, which is the computation of stiffness of the fiber-reinforced concrete based on known material distribution and local material properties. The next numerical experiments describe one of the possible inverse problems, an identification of the material properties (Young's modulus E and Poisson ratio ν) of the concrete matrix from known material distribution, elastic properties of fibers and response of the sample (REV) to uniaxial or triaxial loading tests. This inverse problem exploits the objective function (the cost functional) $J(p)$, $p = (E; \nu)$, $w_{1k} = w_{2k} = 1$, introduced in Section 2. For more details, see [4].

Table 4. Results of material identification applying Dirichlet and Neumann-Dirichlet BC. The number of transformation steps of the applied Nelder-Mead method and the identified averaged material properties of the concrete matrix for each REV

Sample	Dirichlet BC			Neumann BC		
	Steps	E , GPa	ν	Steps	E , GPa	ν
0	135	19.020	0.199	138	18.996	0.200
2	141	19.000	0.200	135	19.004	0.200
3	141	19.005	0.200	135	19.006	0.200
4	141	19.029	0.200	162	19.034	0.200
5	141	19.007	0.200	129	19.007	0.200

Optimization is performed by the non-gradient Nelder-Mead (NM) method with starting values (E, ν) provided by three pairs (17.000, 0.26), (21.000, 0.17), (18.000, 0.23). In each step of the NM method, three direct problems (three computation of local stresses and strains), corresponding to simulation of uniaxial loading tests for each direction X , Y and Z , are solved. The Dirichlet BC describe the same loading as in case of homogenization tests. The Neumann-Dirichlet BC enter a combination of the pure Dirichlet and pure Neumann BC introduced earlier. It means the prescribed non-zero displacement on the top side in the direction of loading, zero displacement on the bottom side in the direction of loading and zero normal forces on the other sides. The NM iterations are stopped if the decrease of the cost functional and differences in the identified parameters are sufficiently small.

The numbers of transformation steps performed by the NM optimization procedure and the averaged values of the identified material properties are summarized in Table 4. The Dirichlet BC on the whole sample boundary are used for comparison purposes. They are applicable if the loading response is computed artificially. The results obtained show a good accordance with values for the concrete matrix presented in Table 2. Considering the number of NM steps and the need to repeat the FEM calculation several times in each step, the results document also substantially increased requirements on the computational power of the used computer.

5. Tuning of parallel solvers

Nowadays powerful parallel computers for HPC have hundreds or thousands of cores, or more. It brings new paradigm of solver programming and urges us to adapt our parallel solver originally developed for Beowulf type clusters with up to 20 processors.

The original solver is based on the PCG method, uses one-directional domain decomposition for parallelization of the iterative process as well as construction of efficient one-level and two-level AS preconditioners (AS1, AS2), see their definition through B_{AS1} and B_{AS2} in Section 3. Parallel processes communicate through message passing (MPI standard).

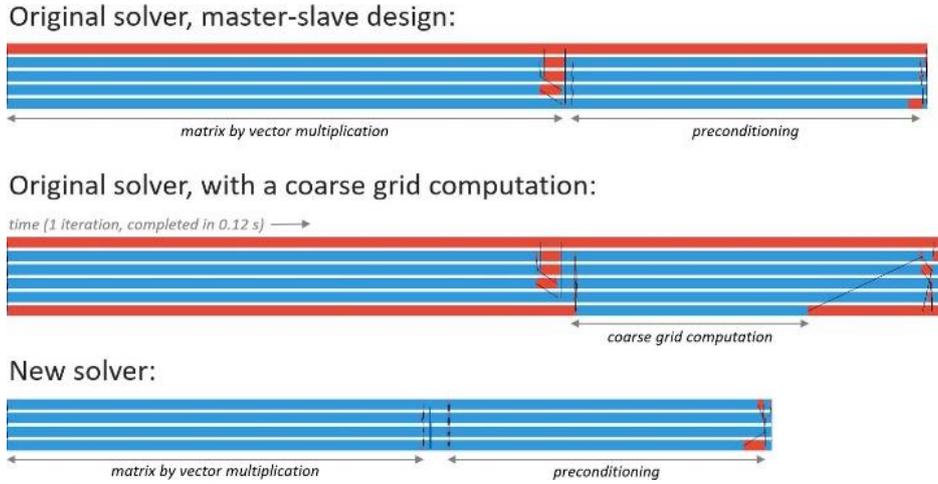


Fig. 1. Traces of one PCG iteration processing four subdomains. From top, records for the original solver with AS1 and AS2, and new solver with AS1 only. States of parallel processes: work (blue), wait or idle (red)

Fig. 1 shows traces of runs of parallel solvers produced by the Intel Trace Analyzer. The implementation of the original solver follows the master-slave design, when the first process (from above) is the master, almost idle, just controlling the iterative process and computing two global scalar products. Each of the four slave processes (below the master) work on its portion of data, especially during the dominating operations: Matrix by Vector multiplication (MXV) and PREConditioning (PREC).

The second trace adds a coarse grid computation to AS2. This process is very important because the coarse grid computation strongly improves the efficiency of the preconditioner and speeds up the convergence of the PCG iterations. A separate process, in this case idle for more than a half of the iteration execution time, performs the computation. For larger number of subdomains, this ratio changes and the coarse grid computation becomes most time consuming, see further discussion.

The third trace documents a run of the new version of the parallel solver, surpassing the original one in the execution time and a better utilization of processes. New solver works internally with data in double precision and dynamic allocation of memory, uses a modified domain decomposition (with an overlapping of subdomains) leading to a better load balancing of processes, has optimized (mainly global) communication of processes and calculations in loops (during MXV and PREC operations). It abandons the master-slave design, the negligible amount of work performed by the master process was taken over by the other processes.

The new solver was tested on the solution of large-scale linear system of 193 million degrees of freedom. The test problem arose from the numerical homogenization experiment performed on the material Sample 4, when the Dirichlet BC prescribing the loading in direction Z were applied. The solver used the one-level additive Schwarz preconditioner AS1 and iterations were stopped with the obtained relative residual accuracy 10^{-4} . The results of performance comparison study and scalability tests for various available HPC computers are summarized in Table 5.

Table 5. A performance comparison study and scalability tests of newly implemented parallel solver executed on various HPC computers. The Number of Processes (No P), the total Number of Iterations (No It), the computation time T , the averaged computation time per iteration T_1 and the relative speed-up S_1 (computed from T_1)

No P	No It	Salomon			Anselm			Avitohol			Enna		
		T , s	T_1 , s	S_1	T , s	T_1 , s	S_1	T , s	T_1 , s	S_1	T , s	T_1 , s	S_1
1	219	5631	25.71	–	10349	47.26	–	10975	50.11	–	15112	69.00	–
2	356	4627	13.00	1.98	8213	23.07	2.05	8642	24.34	2.06	12259	34.44	2.00
4	391	2702	6.91	3.72	4582	11.72	4.03	4813	12.31	4.07	6687	17.10	4.03
8	439	1763	4.02	6.40	2962	6.75	7.00	2788	6.35	7.89	3809	8.68	7.95
16	487	858	1.76	14.59	1439	2.95	15.99	1676	3.44	14.56	2613	5.35	12.89
32	529	463	0.88	29.38	833	1.58	29.95	1059	2.01	24.99	2315	4.38	15.74
64	535	292	0.55	47.11	506	0.95	49.87	517	0.97	51.76	2474	4.63	14.87

The resulting computation times T correspond to general performance characteristics of processors in the used compute nodes, see Table 6 for the values from PassMark benchmarks [12]. On all computer platforms, the computation times scale down with the increasing number of parallel processes. However, on Enna, this behavior of the solver deteriorates on the highest numbers of used cores (32, 64), when the solver approaches limits given by communication subsystem of this shared memory NUMA multiprocessor.

As expected, the number of PCG iterations grows with the increasing number of used cores. This effect is caused by the applied one-level additive Schwarz preconditioner AS1 and can be eliminated by involving a global coarse problem to make the two-level preconditioner AS2, which should stabilize the number of iterations and improve the efficiency of the solver.

The relative speed-up S_1 is computed from the averaged computation times per iteration T_1 . For smaller numbers of used cores (2-16), it grows almost ideally, when the whole calculation including communications runs within one compute node. After that, the relative speed-up gently decelerates due to increasing volume of communication and involving communications among compute nodes. Even in such case, the parallel implementation of the solver is very efficient when the computation time is reduced from more than 1.5 hour (one core) to less than 5 minutes (64 cores) on Salomon.

Table 6. Brief CPU performance characteristics according to PassMark tests. The Number of Cores (No C), single thread rating STR, the averaged CPU mark AM and the used version of Fortran compilers (Intel Parallel Studio XE)

Computer	Processor	No C	STR	AM	Compiler
Salomon	E5-2680 V3	12	1872	18782	2017.1.132
Anselm	E5-2665	8	1510	11812	2015.3.187
Avitohol	E5-2650 V2	8	1655	13119	2017.2.050
Enna	E7-8837	8	602	11752	2016.1.056

Fig. 2 indicates the next step in the parallel solver optimization. With the increase of processes, the execution time of the most demanding MXV and PREC operations performed by worker processes scales down correspondingly, whilst the execution time of a coarse grid computation stays constant. In the example shown in Fig. 2, the described effect limits the possible speed-up of the solver only to three, instead of expected eight, which corresponds to the increase of the number of processes.

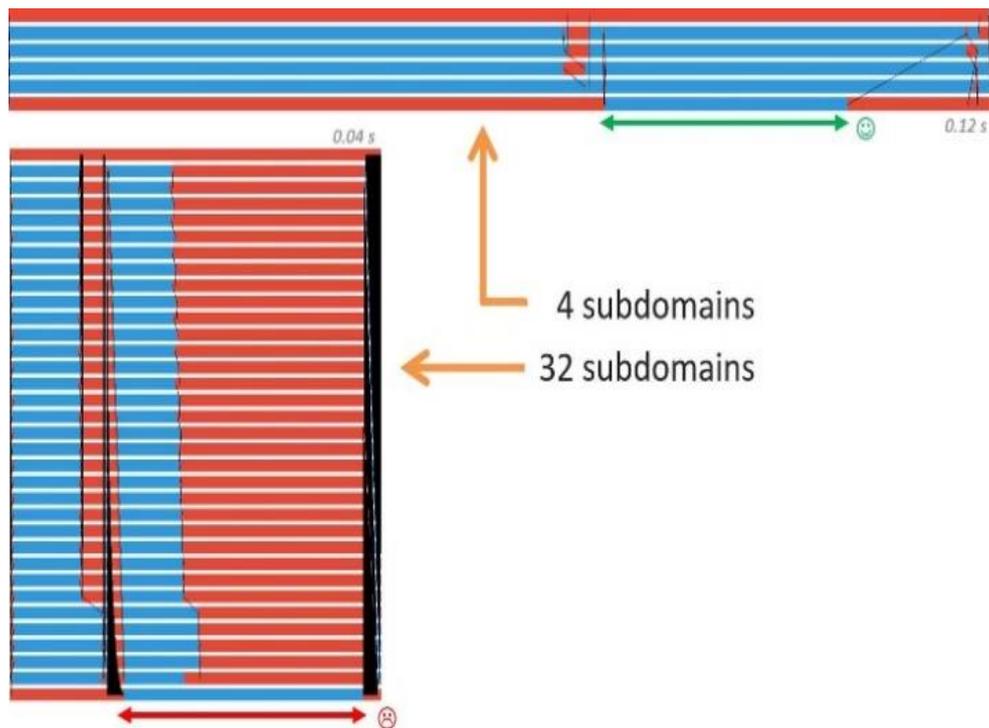


Fig. 2. A coarse grid computation bottleneck in the original solver. Traces of one PCG iteration processing 4 and 32 subdomains

Such negative effect can be eliminated by a coarse grid parallelization in a hybrid way, when not all processes perform the same calculations. On hundreds of computing elements (processors or cores), such hybrid parallelization includes the most of processes solving the subproblems corresponding to subdomains and only a few (units or tens) of processes performing coarse grid computations in parallel. It should not substantially decrease convergence properties of the applied AS2 preconditioner, but dramatically increase the efficiency of the resulting PCG iterations. However, the described hybrid parallelization can bring difficulties how to treat optimal load balancing of processes.

6. Conclusions

The paper demonstrates the need for high performance computing by focusing on one engineering application – investigation of fiber-reinforced concrete. The primary analysis is solving a microscale problem for homogenization within the range of linear material behavior. This basic problem can be modified (extended) in several directions and any of them substantially increases the computational demands. One extension, roughly described in this paper, is the solution of the inverse problem of identification of local material parameters or some level of deboning of the matrix and fibers. This problem is solved by an optimization procedure which requires repeated solution of the basic problem. The computational demands can increase about hundred times. Another extension is in solving not only selected and scanned samples of the concrete, but stochastic generation of a set of such samples and evaluation of the mean properties by Monte Carlo or multi-level Monte Carlo methods, see, e.g., [10]. The last extension is to consider strengths and nonlinear post peak behavior, which include damage mechanics techniques, see, e.g., [6] and the references there.

The tests reported in Table 5 used highly parallelizable CG iterations with one-level additive Schwarz preconditioner. The testing with two-level Schwarz method, which is a bit more demanding for interpretation, will be done in the near future.

Acknowledgements: The work is supported by COST Action IC1305 Project Network for Sustainable Ultrascale Computing and a bilateral Project BAS-17-08 of collaboration between the Institute of Geonics – CAS, and Institute of Information and Communication Technologies – BAS. Further support is through the projects LD15105 Ultrascale Computing in Geo-Sciences and LQ1602 IT4Innovations Excellence in Science supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Mihajlovic, M., et al. Applications for Ultrascale Computing. – International Journal Supercomputing Frontiers and Innovations, Vol. 2, 2015, pp. 19-48.
2. Blaheta, R., A. Kolcun, O. Jakl, K. Soucek, J. Stary, I. Georgiev. HPC in Computational Micromechanics of Composite Materials. – NESUS Workshop, Cracow, Poland, 2015.
3. Blaheta, R., O. Jakl, R. Kohut, J. Stary. GEM – A Platform for Advanced Mathematical Geosim-Ulations. – In: R. Wyrzykowski et al., Eds. PPAM 2009. Part I. LNCS 6067, 2010, pp. 266-275.
4. Blaheta, R., R. Kohut, A. Kolcun, K. Soucek, L. Stas, L. Vavro. Digital Image Based Numerical Micromechanics of Geocomposites with Application to Chemical Grouting. – International Journal of Rock Mechanics and Mining Sciences, Vol. 77, 2015, pp. 77-88.
5. Blaheta, R., R. Kohut, J. Stary. Computational and Reliability Aspects of Micro-Geomechanics. – In: Oka, Murakami, Uzuoka and Kimoto, Eds. Computer Methods and Recent Advances in Geomechanics, Taylor & Francis Group, London, 2015.
6. Hickman, M. A., P. K. Basu. Stochastic Multiscale Characterization of Short-Fiber Reinforced Composites. – Technische Mechanik, Vol. 36, 2016, No 1-2, pp. 13-31.
7. Guan, X., X. Liu, X. Jia, Y. Yuan, J. Cui, H. A. Mang. A Stochastic Multiscale Model for Predicting Mechanical Properties of Fiber Reinforced Concrete. – International Journal of Solids and Structures, Vol. 56-57, 2015, pp. 280-289.

8. Haslinger, J., R. Blaheta, R. Hrtus. Identification Problems with Given Material Interfaces. – Journal of Computational and Applied Mathematics, Early View, June 2016.
9. Salomon Cluster Documentation.
<https://docs.it4i.cz/salomon>
10. Blaheta, R., S. Domesova, M. Beres. A Study of Stochastic FEM Method for Porous Media Flow Problem. – In: Proc. of PhD Workshop, Institute of Geonics, CAS, December 2015.
11. Blaheta, R., O. Jakl, J. Stary, E. Turan. Parallel Solvers for Numerical Upscaling. – In: PARA'2012, LNCS 7782, Springer-Verlag, 2013, pp. 375-386.
12. PassMark Software CPU Benchmarks.
<https://www.cpubenchmark.net>