České vysoké učení technické v Praze
Fakulta stavební

Czech Technical University in Prague
Faculty of Civil Engineering

doc. Ing. Jaroslav Kruis, Ph.D.

Řešení rozsáhlých úloh inženýrské praxe na paralelních počítačích

Solution of Large Engineering Problems on Parallel Computers

Summary

There are four basic groups of demanding computations. First group contains problems with many unknowns (more than million) which are solved only once. Such problems can be split into smaller subproblems which are distributed to processors of a parallel computer and they are solved independently there. At the end, the original continuity on subproblem interfaces is enforced. This is the principle of domain decomposition methods. Their parallelization requires data transfer among processors, synchronization of operations and load balancing.

Second group contains problems with moderate number of unknowns (tens of thousands) which are solved repeatedly and the order of solution is not important or it is important only partially. The Monte Carlo simulations, various optimization tasks or multi-scale analysis, where a problem defined on the lower level is solved for every finite element of the upper level can serve as examples. Parallelization of such computations can lead to the ideal speedup because the master processor sends and receives only small amount of data. The same number of operations executed on the slave processors leads to the ideal load balancing which is another advantage.

Third group contains problems with a moderate number of unknowns which are solved repeatedly and the order of solution is important. As an example can serve a numerical time integration. Parallelization of such problems is questionable and negligible speedup is usually attained.

Fourth group contains problems with many unknowns which should be solved repeatedly. These are not solvable at this time.

Two domain decomposition methods and two-level analysis of coupled heat and moisture transport in a masonry bridge structure are described and illustrate the problems of the first and second group discussed above.

Souhrn

Existují čtyři základní skupiny náročných výpočtů. První skupina obsahuje problémy s velkým množstvím neznámých (více než milion), které se řeší jen jednou. Takové problémy lze rozdělit na podproblémy, které jsou rozmístěny na procesory paralelního počítače, kde jsou nezávisle zpracovávány. Na závěr se vynutí původní spojitost na rozhraních podproblémů. To je princip metod rozložení oblasti na podoblasti. Jejich paralelizace vyžaduje přesuny dat mezi procesory, synchronizaci výpočtu a vyvážení výpočetní zátěže.

Druhá skupina obsahuje úlohy s přiměřeným počtem neznámých (desítky tisíc), které jsou ale řešeny opakovaně a na pořadí výpočtů nezáleží nebo záleží jen částečně. Příkladem jsou výpočty metodou Monte Carlo, různé optimalizační úlohy nebo víceúrovňové analýzy, ve kterých se pro všechny konečné prvky vyšší úrovně řeší jeden problém nižší úrovně. Parallelizace těchto úloh může vést k ideálnímu zrychlení, protože se z řídícího procesoru (master processor) rozesílá a přijímá jen malé množství dat. Další výhodou je, že se v mnoha případech na všech procesorech zpracovává stejné množství dat a operací, takže je výpočet dokonale vyvážen.

Třetí skupina obsahuje úlohy s přiměřeným počtem neznámých, které se řeší opakovaně a na pořadí záleží. Příkladem může být numerická integrace v čase. Paralelizace těchto úloh je problematická a obvykle se nedosáhne podstatného zrychlení.

Čtvrtá skupina pak obsahuje problémy s velkým množstvím neznámých, které by měly být řešeny opakovaně. Ty jsou v současné době prakticky neřešitelné.

Příklad dvou metod rozložení oblasti na podoblasti a dvouúrovňový výpočet sdruženého vedení tepla a vlhkosti na zděné mostní konstrukci ilustrují problémy první a druhé skupiny úloh popsaných výše.

Klíčová slova: paralelní výpočty, metody rozložení oblasti na podoblasti, FETI, metoda Schurových doplňků, víceúrovňová analýza, sdružený přenos tepla a vlhkosti.

Keywords: parallel computing, domain decomposition method, FETI, Schur complement method, multi-scale analysis, coupled heat and moisture transport.

Contents

1 Introduction

Although hardware is permanently being developed, requirements on computer power and memory arising from engineering practice exceed usually its possibility. Single processor computers are definitely unable to solve large problems. Therefore parallel computers are becoming very popular and they are used not only in academic institutions but also in engineering practice.

Parallel computers could be sorted out with respect to their memory. The computers with shared memory are called massively parallel computers. All processors have access to all data. On the contrary, a cluster of single-processor computers connected via suitable network is a parallel computer with distributed memory. Each processor has an access only to data stored in its memory. If a processor A needs data stored in memory of processor B, the data has to be sent. The massive parallel computers are significantly more expensive in comparison with clusters. The biggest parallel computers at this time are clusters of multi-processor nodes and the memory is distributed while in nodes is shared.

Recently, computers with many cores are becoming very popular. They can be handled similarly to the massive parallel computers with shared memory. Cloud computing is another possibility for high performance computing. Prediction of which computers will be used in future is very difficult. It can be demonstrated on quotation of Bell published in Science in 1992: "Distributed memory machines require considerable reprogramming of algorithms, so there is skepticism as to whether they will replace multi-processors with shared memory." Belytschko in 1997 wrote: "Heterogeneous work-station environments are increasingly being used for parallel processing by PVM (Parallel Virtual Machine)." At this time, distributed memory machines are used and communication is organized by MPI (Message Passing Interface).

The progress of parallel computer development could be seen in Top500 computers in the world. In June 1993, the top computer was CM-5/1024, Thinking Machines Corporation in Los Alamos National Laboratory, United States, with 1,024 cores and the power 131 GFlop/s. In June 2014, the top computers are:

1. The top on the World: National Super Computer Center in Guangzhou, China, Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P 3,120,000 cores, 54902.4 TFlop/s, 17,808 kW power

6. The top in Europe: Swiss National Supercomputing Centre, Switzerland, Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc. 115,984 cores, 7788 TFlop/s, 2,325 kW power

? The top in Czech Republic: IT4I Ostrava, Czech Republic, approximately 40,000 processors, 1 Pflop/s[1]

There are four basic groups of very demanding computational problems. First group contains problems with many unknowns (degrees of freedom) which are solved once. As an example could serve large structure with very detailed finite element mesh. Such problems are split into subproblems which are distributed to processors of a parallel computer where they are solved independently. At the end, the original continuity has to be enforced. The approach described is basis of domain decomposition methods. Parallelization of such problems requires communication among processors and several synchronization points. More details about this group of problems is described in Section 2.

Second group contains problems with moderate number of unknowns which are solved many times. An example of such problem is the usage of the Monte Carlo simulation where a problem is solved thousand or even million times. Another example is multi-scale problem where a single micro-scale problem is solved for every finite element of the macro-scale problem. Parallelization of such problems is much more easy because usually few parameters are sent from the master processor which controls the computation to the slave processors which solve the problems. Parallelization of a two-scale problem is briefly described in Section 3.

Third group contains problems with moderate number of unknowns which are successively solved many times. Typical example of such problem is time integration. Parallelization of such problems is questionable and usually does not lead to significant speedup.

Fourth group contains problems with many unknowns (millions of unknowns) which should be solved many times (thousand or more times). Unfortunately, these problems cannot be solved at this time.

2 Domain Decomposition Methods

Two methods of domain decomposition are described in this section. First is the Schur complement method and second is the FETI (Finite

[1]The computer is not installed now and therefore it is not known the order in the list of Top500 computers.

Element Tearing and Interconnecting) method.

2.1 The Schur Complement Method

The Schur complement method is also called the static condensation method, the method of sub-structuring or the primal domain decomposition method. The Schur complement method is clearly based on the Schur complements and this notation is used especially by mathematicians. Engineers often call it the condensation method or the method of sub-structuring. The method of sub-structuring was used many decades ago when the RAM of computers was extremely small. The decomposition of the structure into smaller substructures is native in engineering design and therefore was also employed in numerical simulation. Elimination (condensation) of internal unknowns (degrees of freedom) is also a very popular technique among engineers. Both features are common to the Schur complement method. The last notation, the primal domain decomposition method, is connected with the fact that only original unknowns are considered during computation and the concept has a special reason when dual unknowns are used such as in the FETI methods. The Schur complement method is described in many books and papers from the engineering point of view [20, 21] or from the mathematical point of view [2, 25, 29].

The following system of equations can be obtained from the finite element method applied to problem decomposed into m subdomains

$$\begin{pmatrix} \boldsymbol{K}_1^{[ii]} & & \mathbf{0} & & \boldsymbol{K}_1^{[ib]} \\ & \boldsymbol{K}_2^{[ii]} & & & \boldsymbol{K}_2^{[ib]} \\ \vdots & & \ddots & & \vdots \\ & & & \boldsymbol{K}_m^{[ii]} & \boldsymbol{K}_m^{[ib]} \\ \boldsymbol{K}_1^{[bi]} & \boldsymbol{K}_2^{[bi]} & \dots & \boldsymbol{K}_m^{[bi]} & \boldsymbol{K}^{[bb]} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_1^{[i]} \\ \boldsymbol{d}_2^{[i]} \\ \vdots \\ \boldsymbol{d}_m^{[i]} \\ \boldsymbol{d}^{[b]} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1^{[i]} \\ \boldsymbol{f}_2^{[i]} \\ \vdots \\ \boldsymbol{f}_m^{[i]} \\ \boldsymbol{f}^{[b]} \end{pmatrix} \tag{1}$$

where $\boldsymbol{d}_j^{[i]}$, $j \in \{1, 2, \dots, m\}$ are the vectors of unknowns defined inside the j-th subdomain, $\boldsymbol{d}^{[b]}$ is the vector of unknowns on the subdomain boundaries, $\boldsymbol{f}_j^{[i]}$, $j \in \{1, 2, \dots, m\}$ are the vectors of right hand sides defined inside the j-th subdomain, $\boldsymbol{f}^{[b]}$ is the vector of right hand side on the subdomain boundaries. The superscript i denotes an internal quantity while b stands for a boundary one. Clearly, ib and bi denotes coupling between internal and boundary quantities. The number of

equations collected in the j-th block is denoted by $n_j^{[i]}$. The number of equations collected in the last block is denoted by $n^{[b]}$. If the number of all rows in the system (1) is n, the following simple relationship is valid

$$n = n^{[b]} + \sum_{j=1}^{m} n_j^{[i]} \tag{2}$$

The diagonal blocks can be written as $\boldsymbol{K}_j^{[ii]} \in R^{n_j^{[i]} \times n_j^{[i]}}$, while the vectors $\boldsymbol{d}_j^{[i]}$ and $\boldsymbol{f}_j^{[i]}$ are from $R^{n_j^{[i]}}$ and the off-diagonal blocks $\boldsymbol{K}_j^{[ib]}$ are from $R^{n_j^{[i]} \times n^{[b]}}$. If all diagonal blocks $\boldsymbol{K}_1^{[ii]}$ to $\boldsymbol{K}_m^{[ii]}$ are nonsingular, particular blocks of the vector of unknowns $\boldsymbol{d}$ can be expressed in the form

$$\boldsymbol{d}_j^{[i]} = \left(\boldsymbol{K}_j^{[ii]}\right)^{-1} \left(\boldsymbol{f}_j^{[i]} - \boldsymbol{K}_j^{[ib]} \boldsymbol{d}^{[b]}\right) \ . \tag{3}$$

Then the substitution of equation (3) into the last equation of (1) gives a new system

$$\begin{pmatrix} \boldsymbol{K}_1^{[ii]} & & & \boldsymbol{0} & & \boldsymbol{K}_1^{[ib]} \\ & \ddots & & & & \vdots \\ & & \boldsymbol{K}_{j-1}^{[ii]} & & & \boldsymbol{K}_{j-1}^{[ib]} \\ \boldsymbol{0} & & & \boldsymbol{K}_{j+1}^{[ii]} & & \boldsymbol{K}_{j+1}^{[ib]} \\ & & & & \ddots & \vdots \\ \boldsymbol{K}_1^{[bi]} & & & \dots & & \tilde{\boldsymbol{K}}^{[bb]} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_1^{[i]} \\ \vdots \\ \boldsymbol{d}_{j-1}^{[i]} \\ \boldsymbol{d}_{j+1}^{[i]} \\ \vdots \\ \boldsymbol{d}^{[b]} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1^{[i]} \\ \vdots \\ \boldsymbol{f}_{j-1}^{[i]} \\ \boldsymbol{f}_{j+1}^{[i]} \\ \vdots \\ \tilde{\boldsymbol{f}}^{[b]} \end{pmatrix} \tag{4}$$

where the matrix $\tilde{\boldsymbol{K}}^{[bb]}$ is in the form

$$\tilde{\boldsymbol{K}}^{[bb]} = \boldsymbol{K}^{[bb]} - \boldsymbol{K}_j^{[bi]} \left(\boldsymbol{K}_j^{[ii]}\right)^{-1} \boldsymbol{K}_j^{[ib]} \tag{5}$$

and the vector $\tilde{\boldsymbol{f}}^{[b]}$ has the form

$$\tilde{\boldsymbol{f}}^{[b]} = \boldsymbol{f}^{[b]} - \boldsymbol{K}_j^{[bi]} \left(\boldsymbol{K}_j^{[ii]}\right)^{-1} \boldsymbol{f}_j^{[i]} \tag{6}$$

The j-th matrix equation is missing in (4). After repeating this process for each row of (1) except the last one, the reduced system of equations

becomes

$$\left(\boldsymbol{K}^{[bb]} - \sum_{j=1}^{m} \boldsymbol{K}_j^{[bi]} \left(\boldsymbol{K}_j^{[ii]} \right)^{-1} \boldsymbol{K}_j^{[ib]} \right) \boldsymbol{d}^{[b]} =$$

$$= \quad \boldsymbol{f}^{[b]} - \sum_{j=1}^{m} \boldsymbol{K}_j^{[bi]} \left(\boldsymbol{K}_j^{[ii]} \right)^{-1} \boldsymbol{f}_j^{[i]} \tag{7}$$

where $\boldsymbol{d}^{[b]} \in R^{n^{[b]}}$ is the vector of unknowns. The matrix of the system (7) is from the set $R^{n^{[b]} \times n^{[b]}}$. The original system of equations contains n unknowns while the reduced system (7) contains only $n^{[b]}$ unknowns. This reduction of unknowns is an important feature of the Schur complement method. When the vector $\boldsymbol{d}^{[b]}$ is computed, all vectors of internal unknowns $\boldsymbol{d}_j^{[i]}$ are established from equation (3).

The system of equations (7) is also called the coarse grid problem because only unknowns defined on the boundaries are used. Each subdomain could be assumed to be one super-element and the coarse grid is assumed to be created by those super-elements.

It is clear from equation (3) that only two appropriate blocks of the matrix $\boldsymbol{K}$ are involved in computation of the blocks $\boldsymbol{d}_j^{[i]}$. In other words, particular blocks $\boldsymbol{d}_j^{[i]}$ are expressed independently of other parts of the vector of unknowns except the last block $\boldsymbol{d}^{[b]}$ which depends on all blocks of the system. The aim of the method is to minimize $n^{[b]}$ which results in an efficient solution of equation (7).

The reduced system of equations can be solved by a direct or an iterative method. The choice of a solution method depends on many factors which are not strict. If the Schur complements are small and the number of subdomains and therefore the number of processors is also small, the direct method can be applied very efficiently. In this case, only small amounts of data (condensed matrices and vectors) are sent between slaves and the master and the reduced system is placed in the RAM on the master processor. No communication is necessary during solution of the reduced problem and when boundary unknowns are evaluated, only a small amount of data is sent from the master to the slaves.

In the case of iterative methods, there are two basic possibilities. The first one is based on the similar strategy to the previous one, that is the condensed matrices and vectors are sent from slaves to the master processor, the matrix of the reduced system is assembled in the RAM and the sequential iterative algorithm is used to solve the system. Once

again, no communication is required during solution and at the end the evaluated boundary unknowns are sent from the master to the slaves. Reduction of communication is an advantage but there is also a disadvantage which lies in the fact that only the master processor is working during solution while all the slaves are inactive.

The second possibility is more appealing and is based on distributed computation. In this case the condensed matrices are not sent from slaves to the master and the iterative method capitalises on the fact that only the matrix-vector multiplication is necessary for solution. The master processor manages the computation. It sends appropriate parts of a vector which should be multiplied by a matrix of the reduced system to slaves where multiplication is computed with the condensed matrices. After that, slaves send the results obtained to the master processor where particular contributions are added into the global vector. During this algorithm small amounts of data are sent among the processors but on the contrary to the previously described strategy, all processors are active. The second approach of the iterative solution of the reduced system of equations is suitable for large problems because the matrix of the reduced system is not assembled on the master processor. Therefore the size of the RAM on the master is not a restrictive factor.

Figure 1 shows a three-dimensional finite element mesh generated on a reactor vessel. The mesh is decomposed into 20 submeshes and an analysis could be solved on 20 processors.

2.2 FETI Method

The FETI method was introduced in [10] in 1991. The original method was extended to the fourth-order static and dynamic plate bending problems in reference [7] and to shell problems in reference [6]. Special attention is devoted to the continuity conditions at the substructure corners and it leads to a two level FETI method.

Convergence and scalability are analysed in reference [8] which also contains discussion on the optimal and computationally efficient preconditioners. In this reference, a closed connection with the balancing domain decomposition method introduced in [19] is mentioned. Papers [24] and [23] are devoted to preconditioners of the FETI method for heterogenous mechanics problems and constrained problems.

Application of the FETI method to large scale linear and geometrically non-linear structural analysis problems is described in [9]. Selection of one or two level FETI methods is studied. Attention is also devoted to particular components of the method and to handling of

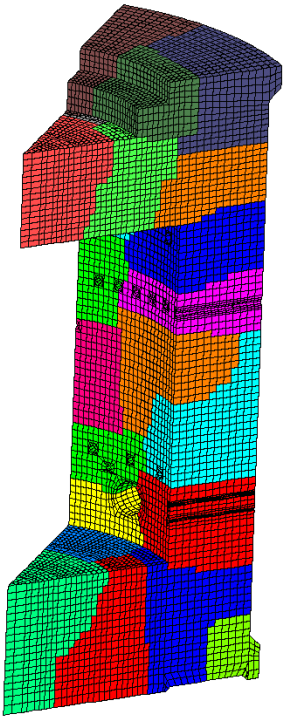

Figure 1: Mesh of a reactor vessel decomposed into 20 subdomains.

singularities. Reference [1] describes scalability of the FETI method on 1,000 processors and application to highly heterogeneous problems. A problem with more than 5.5 million of degrees of freedom is presented.

Papers [3], [5] are devoted to the contact problems which are described by variational inequalities. Application of the FETI method to analysis of composite materials are studied in [4], [15], [16], [12], [14].

The derivation of the basic equations will be shown on mechanical problem and can be completed with the help of a energy functional of an elastic solid body which has the following form

$$\begin{aligned}\Pi(\boldsymbol{u}) &= \frac{1}{2}\int_{\Omega}(\boldsymbol{\varepsilon}(\boldsymbol{x}))^T\,\boldsymbol{D}(\boldsymbol{x})\boldsymbol{\varepsilon}(\boldsymbol{x})\mathrm{d}\Omega - \int_{\Omega}(\boldsymbol{u}(\boldsymbol{x}))^T\boldsymbol{b}(\boldsymbol{x})\mathrm{d}\Omega \\ &- \int_{\Gamma^t}(\boldsymbol{u}(\boldsymbol{x}))^T\boldsymbol{t}(\boldsymbol{x})\mathrm{d}\Gamma^t \end{aligned} \tag{8}$$

where the integrals are taken over the whole domain. $\boldsymbol{\varepsilon}$ is the strain vector, $\boldsymbol{D}$ is the stiffness matrix of material, $\boldsymbol{u}$ is the vector of displacements, $\boldsymbol{b}$ is the vector of body forces and $\boldsymbol{t}$ is the vector of surface traction. Energy is a scalar quantity and therefore it can be computed as the sum of several contributions. This is a suitable property in connection with domain decomposition methods because the original domain (elastic body) is decomposed into several smaller subdomains, in case of the FETI method in non-overlapping subdomains. The energy functional of the body decomposed into m subdomains has the form

$$\begin{aligned}\Pi(\boldsymbol{u}) &= \sum_{j=1}^{m}\left(\frac{1}{2}\int_{\Omega_j}(\boldsymbol{\varepsilon}(\boldsymbol{x}))^T\,\boldsymbol{D}(\boldsymbol{x})\boldsymbol{\varepsilon}(\boldsymbol{x})\mathrm{d}\Omega_j - \int_{\Omega_j}(\boldsymbol{u}(\boldsymbol{x}))^T\boldsymbol{b}(\boldsymbol{x})\mathrm{d}\Omega_j\right) \\ &- \sum_{j=1}^{m}\int_{\Gamma_j^t}(\boldsymbol{u}(\boldsymbol{x}))^T\boldsymbol{t}(\boldsymbol{x})\mathrm{d}\Gamma_j^t + \sum_{j=1}^{m}\int_{\Gamma_j^i}[\boldsymbol{u}(\boldsymbol{x})]\boldsymbol{\mu}(\boldsymbol{x})\mathrm{d}\Gamma_j^i \end{aligned} \tag{9}$$

The j-th subdomain is denoted by Ω_j, the boundary of the j-th subdomain is denoted by Γ_j. Each boundary Γ_j is the union of two parts, Γ_j^i and Γ_j^e. The external part of the boundary, Γ_j^e, is defined as $\Gamma_j^e = \Gamma \cap \Gamma_j$ and the internal part, Γ_j^i, is the remaining part, therefore $\Gamma_j^i = \Gamma_j - \Gamma_j^e$. The internal parts of the subdomains boundaries are formed after decomposition of the original domain into subdomains. The union of $\bar{\Omega}_j$ gives the original domain $\bar{\Omega}$ and the union of the external parts of the boundaries Γ_j^e gives the boundary of the original domain Γ. Γ_j^t denotes for the parts of the boundary of the j-th subdomain with a prescribed traction.

A new term in the energy functional occurs after the decomposition of the original body into several smaller subdomains. It is the last term where the Lagrange multipliers $\boldsymbol{\mu}(\boldsymbol{x})$ were defined. These multipliers enforce the continuity among subdomains. The discontinuity of the vector of displacements, $\boldsymbol{u}(\boldsymbol{x})$, is denoted by $[\boldsymbol{u}(\boldsymbol{x})]$. The Lagrange multipliers are defined only on the boundaries Γ_j^i which were created by the decomposition.

The discretized form of the energy functional (9), where the approximation by the finite element method was used, has the form

$$\begin{aligned}\Pi &= \sum_{j=1}^{m} \frac{1}{2} \int_{\Omega_j} \boldsymbol{d}_j^T \boldsymbol{B}_j^T \boldsymbol{D} \boldsymbol{B}_j \boldsymbol{d}_j \mathrm{d}\Omega_j - \sum_{j=1}^{m} \int_{\Omega_j} \boldsymbol{d}_j^T \left(\boldsymbol{N}_j^{[d]} \right)^T \boldsymbol{N}_j^{[b]} \bar{\boldsymbol{b}}_j \mathrm{d}\Omega_j \\ &- \sum_{j=1}^{m} \int_{\Gamma_j^t} \boldsymbol{d}_j^T \left(\boldsymbol{N}_j^{[d]} \right)^T \boldsymbol{N}_j^{[t]} \bar{\boldsymbol{t}}_j \mathrm{d}\Gamma_j^t + \sum_{j=1}^{m} \boldsymbol{\lambda}^T \boldsymbol{\mathcal{E}}_j \boldsymbol{d}_j \end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{E}}_j$ denotes the assembling matrices, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, $\boldsymbol{d}_j$ is the vector of nodal displacements on the j-th subdomain, $\boldsymbol{N}_j^{[d]}$ and $\boldsymbol{N}_j^{[b]}$ are the matrices of shape functions, $\boldsymbol{B}_j$ are the strain-displacement matrices, $\bar{\boldsymbol{b}}_j$ are the vectors of nodal volume forces and $\bar{\boldsymbol{t}}_j$ are the vectors of nodal surface traction.

The last term of the functional (10) plays an important role because it enforces the compatibility of displacements by utilizing Lagrange multipliers.

There are $m+1$ unknown vectors $\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_m, \boldsymbol{\lambda}$ in the functional (10). The extreme value of the original energy functional (8) is reached at the stationary point of (10) which can be obtained by solving the following system of equations

$$\begin{aligned}\text{For } j &= 1, 2, ..., m \\ \frac{\partial \Pi}{\partial \boldsymbol{d}_j} &= \int_{\Omega_j} \boldsymbol{B}_j^T \boldsymbol{D} \boldsymbol{B}_j \boldsymbol{d}_j \mathrm{d}\Omega_j - \int_{\Omega_j} \left(\boldsymbol{N}_j^{[d]} \right)^T \boldsymbol{N}_j^{[b]} \bar{\boldsymbol{b}}_j \mathrm{d}\Omega_j \\ &- \int_{\Gamma_j^t} \left(\boldsymbol{N}_j^{[d]} \right)^T \boldsymbol{N}_j^{[t]} \bar{\boldsymbol{t}}_j \mathrm{d}\Gamma_j^t + \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda} = \boldsymbol{0} \end{aligned} \tag{11}$$

$$\frac{\partial \Pi}{\partial \boldsymbol{\lambda}} = \sum_{j=1}^{m} \boldsymbol{\mathcal{E}}_j \boldsymbol{d}_j = \boldsymbol{0} \tag{12}$$

With the usual notation

$$\boldsymbol{K}_j = \int_{\Omega_j} \boldsymbol{B}_j^T \boldsymbol{D} \boldsymbol{B}_j \mathrm{d}\Omega_j \tag{13}$$

for the stiffness matrix and

$$\boldsymbol{f}_j = \int_{\Omega_j} \left(\boldsymbol{N}_j^{[d]}\right)^T \boldsymbol{N}_j^{[b]} \bar{\boldsymbol{b}}_j \mathrm{d}\Omega_j + \sum_{j=1}^{m} \int_{\Gamma_j^t} \left(\boldsymbol{N}_j^{[d]}\right)^T \boldsymbol{N}_j^{[t]} \bar{\boldsymbol{t}}_j \mathrm{d}\Gamma_j^t \qquad (14)$$

for the load vector, the system of equations has then the form

$$\text{For} \quad j = 1, ..., m \qquad \boldsymbol{K}_j \boldsymbol{d}_j = \boldsymbol{f}_j - \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda} \qquad (15)$$

$$\sum_{j=1}^{m} \boldsymbol{\mathcal{E}}_j \boldsymbol{d}_j = \boldsymbol{0} \qquad (16)$$

From the physical point of view, equations (15) are equilibrium conditions on particular subdomains and equation (16) is a compatibility condition.

The solution of equations (15) and (16) is not straightforward because of the singularity of the matrices $\boldsymbol{K}_j$. As described above, the original domain is split into independent subdomains. The continuity on the boundaries is enforced by the Lagrange multipliers. The independence of the subdomains leads to singularity, if there are not enough supports on the subdomain. Instead of the matrices $\boldsymbol{K}_j^{-1}$ which generally do not exist, the pseudo-inverse matrix, $\boldsymbol{K}^+$, which is defined as

$$\boldsymbol{K}\boldsymbol{K}^+\boldsymbol{K} = \boldsymbol{K} \qquad (17)$$

has to be used.

A matrix $\boldsymbol{K}_j \in R^{n_j \times n_j}$ will be considered singular. This means that there are linearly dependent rows in the matrix. It is therefore possible to rewrite the matrix in the form

$$\boldsymbol{K}_j = \begin{pmatrix} \boldsymbol{K}_j^{[nn]} & \boldsymbol{K}_j^{[ns]} \\ \boldsymbol{K}_j^{[sn]} & \boldsymbol{K}_j^{[ss]} \end{pmatrix} \qquad (18)$$

where the block $\boldsymbol{K}_j^{[nn]}$ is nonsingular. The pseudo-inverse matrix to the matrix (18) has the form

$$\boldsymbol{K}_j^+ = \begin{pmatrix} (\boldsymbol{K}_j^{[nn]})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \qquad (19)$$

The possible singularity of the matrices $\boldsymbol{K}_j$, must be taken into account when solving the system (11). The necessary condition for solvability is

$$\left(\boldsymbol{f}_j - \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda}\right) \perp \ker \boldsymbol{K}_j \tag{20}$$

where $\ker \boldsymbol{K}_j$ stands for the kernel of the matrix. The kernel is a set which is defined as

$$\ker \boldsymbol{K}_j = \{\boldsymbol{x} \in R^{n_j} : \boldsymbol{K}_j \boldsymbol{x} = \boldsymbol{0}\} \tag{21}$$

More details can be found in reference [11]. The displacement vector can be written with the defined pseudo-inverse matrix and condition (20) as

$$\boldsymbol{d}_j = \boldsymbol{K}_j^+ \left(\boldsymbol{f}_j - \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda}\right) + \boldsymbol{R}_j \boldsymbol{\alpha}_j \tag{22}$$

where the matrix $\boldsymbol{R}_j$ contains in columns the rigid body motions of the subdomain. From a mathematical point of view, the matrix $\boldsymbol{R}_j$ contains the basis vectors of the kernel of the matrix $\boldsymbol{K}_j$. The vector $\boldsymbol{\alpha}_j$ contains the coefficients of linear combinations. After combining relations (22) and (16) the following equation is obtained

$$\sum_{j=1}^{m} \boldsymbol{\mathcal{E}}_j \boldsymbol{K}_j^+ \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda} = \sum_{j=1}^{m} \left(\boldsymbol{\mathcal{E}}_j \boldsymbol{K}_j^+ \boldsymbol{f}_j + \boldsymbol{\mathcal{E}}_j \boldsymbol{R}_j \boldsymbol{\alpha}_j\right) \tag{23}$$

The solvability conditions (20) may be expressed in the form

$$\text{For} \quad j = 1, ..., m \;:\; \boldsymbol{R}_j^T \left(\boldsymbol{f}_j - \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda}\right) = \boldsymbol{0} \tag{24}$$

For the subsequent analysis the following quantities will be defined

$$\boldsymbol{F} = \sum_{j=1}^{m} \boldsymbol{\mathcal{E}}_j \boldsymbol{K}_j^+ \boldsymbol{\mathcal{E}}_j^T \tag{25}$$

$$\boldsymbol{G} = (-\boldsymbol{\mathcal{E}}_1 \boldsymbol{R}_1, -\boldsymbol{\mathcal{E}}_2 \boldsymbol{R}_2, ..., -\boldsymbol{\mathcal{E}}_m \boldsymbol{R}_m) \tag{26}$$

$$\boldsymbol{g} = \sum_{j=1}^{m} \boldsymbol{\mathcal{E}}_j \boldsymbol{K}_j^+ \boldsymbol{f}_j \tag{27}$$

$$\boldsymbol{e}^T = \left((-\boldsymbol{R}_1^T \boldsymbol{f}_1)^T, (-\boldsymbol{R}_2^T \boldsymbol{f}_2)^T, ..., (-\boldsymbol{R}_m^T \boldsymbol{f}_m)^T \right) \tag{28}$$

and

$$\boldsymbol{\alpha}^T = \left(\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T, ..., \boldsymbol{\alpha}_m^T \right) \tag{29}$$

The vector $\boldsymbol{g}$ is usually in the literature denoted as $\boldsymbol{d}$ which denotes the vector of displacements in this book. The resulting system of equations can be written with quantities from the equations (25), (26), (27), (28) and (29) as

$$\begin{pmatrix} \boldsymbol{F} & \boldsymbol{G} \\ \boldsymbol{G}^T & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{e} \end{pmatrix} \tag{30}$$

The solution of equation (30) will be obtained by the modified conjugate gradient method. The original conjugate gradient method must be modified because of the presence of the additional condition. When the resulting vector of Lagrange multipliers $\boldsymbol{\lambda}$ is obtained, it is possible to compute the vector of coefficients of the linear combination $\boldsymbol{\alpha}$ from the equation

$$\boldsymbol{F}\boldsymbol{\lambda} + \boldsymbol{G}\boldsymbol{\alpha} = \boldsymbol{g} \tag{31}$$

The matrix $\boldsymbol{G}$ is not a square matrix that means that the inverse matrix does not exist. After multiplying relation (31) from the left by $\boldsymbol{G}^T$, the invertible matrix $\boldsymbol{G}^T\boldsymbol{G}$ will be on the left hand side. The vector of $\boldsymbol{\alpha}$ coefficients is expressed as

$$\boldsymbol{\alpha} = \left(\boldsymbol{G}^T\boldsymbol{G} \right)^{-1} \boldsymbol{G}^T \left(\boldsymbol{g} - \boldsymbol{F}\boldsymbol{\lambda} \right) \tag{32}$$

All the necessary data are now available for computing the resulting displacements. The displacement vector is the sum of two parts, the first contribution is the vector based on the solution of the coarse problem

$$\boldsymbol{d}_j^{[\infty]} = \boldsymbol{K}_j^{+} \left(\boldsymbol{f}_j - \boldsymbol{\mathcal{E}}_j^T \boldsymbol{\lambda} \right) \tag{33}$$

and the second vector is created by the linear combination of the rigid body motions

$$\boldsymbol{d}_j^{[\mathrm{ker}]} = \boldsymbol{R}_j \boldsymbol{\alpha}_j \tag{34}$$

For sufficiently supported subdomains the vector $\boldsymbol{d}_j^{[\mathrm{ker}]}$ is equal to the zero vector. The resulting displacement can be written in the form

$$\boldsymbol{d}_j = \boldsymbol{d}_j^{[\infty]} + \boldsymbol{d}_j^{[\mathrm{ker}]} \tag{35}$$

3 Multi-scale problem

The coupled heat and moisture transport in extremely heterogeneous material like masonry is hardly solvable even with modern computers. The reason is that there are relatively small mortar layers in comparison with bricks which enforce a very dense finite element mesh. Therefore, only details of structures can be solved.

In order to solve the whole structures, multi-scale techniques are usually used. The classical first order homogenization in a spatial domain in the framework of the two-step multi-scale computational scheme was proposed in [27]. This technique is based on a meso or micro-scale problems and a macro-scale problem. Whether the micro or meso-scale level is used depends on the characteristic size of the representative volumes which describe the real complicated geometry. In masonry structures, the representative volume contains several bricks and adjacent mortar layers and therefore the meso-scale level is used. The macro-scale problem describes the problem solved by relatively coarse mesh because homogenized material parameters are obtained from meso-scale level. The multi-scale technique sends the actual values from the macro-scale level problem to the meso-scale problem where the appropriate material parameters are obtained and sent back to the macro-scale problem.

It is evident that the multi-scale method is very demanding because a meso-scale problem has to be solved for each macro-scale finite element. If the number of macro-scale finite elements is in thousands only, the solution of meso-scale problems is very demanding even if each meso-scale problem contains only hundreds of degrees of freedom. The multi-scale problems are ideal for parallelization because the meso-scale problems are completely independent and only reasonable amount of data has to be sent between the macro-scale problem and the meso-scale problems.

The coupled heat and moisture transport is described by the Künzel material model which was introduced in 1995 in reference [18] and modified in [30]. The Künzel model is based on the temperature and relative humidity which have to satisfy the mass and heat balance equations in the form of partial differential equations. The mass balance equation has the form

$$\frac{\partial \rho_v}{\partial \varphi}\frac{\partial \varphi}{\partial t} = \mathrm{div}\left((D_\phi + \delta_p p_s)\mathrm{grad}\varphi + \delta_p \varphi \frac{\mathrm{d}p_s}{\mathrm{d}T}\mathrm{grad}T\right) \tag{36}$$

and the heat balance equation has the form

$$\frac{\partial H}{\partial T}\frac{\partial T}{\partial t} = \mathrm{div}\left((\lambda + L_v\delta_p\varphi\frac{\mathrm{d}p_s}{\mathrm{d}T})\mathrm{grad}T + L_v\delta_p p_s\mathrm{grad}\varphi\right) \tag{37}$$

where T is the temperature (K), φ is the relative humidity (-), L_v is the latent heat of evaporation (J/kg), D_φ is the liquid water transport coefficient (kg/m/s), λ is the thermal conductivity (W/m/K), p_s is the partial pressure of saturated water vapour in the air (Pa), δ_p water vapour permeability (kg/m/s/Pa), H is the enthalpy density (J/m^3).

System of partial differential equations (36-37) is solved by the finite element method. Let N denote the number of nodes in the macro-scale problem and n denote the number of nodes in meso-scale problem. The balance equations (36-37) on the macro-scale level after space discretization have the form

$$\boldsymbol{C}\dot{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{d} = \boldsymbol{f} \tag{38}$$

where $\boldsymbol{C}$ is the capacity matrix $(2N, 2N)$, $\boldsymbol{K}$ is the conductivity matrix $(2N, 2N)$, $\boldsymbol{d}$ is the vector of nodal unknowns $(2N)$ and $\boldsymbol{f}$ is the vector of the right hand side assembled from boundary conditions. First order homogenization applied to the balance equations (36-37) leads to the system of algebraic equations on the meso-scale level in the matrix-vector form

$$\begin{pmatrix} \boldsymbol{K}_{TT} & \boldsymbol{K}_{T\varphi} \\ \boldsymbol{K}_{\varphi T} & \boldsymbol{K}_{\varphi\varphi} \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{u}}_T^* \\ \bar{\boldsymbol{u}}_\varphi^* \end{pmatrix} = - \begin{pmatrix} \boldsymbol{L}_{TT}^T & \boldsymbol{L}_{T\varphi}^T \\ \boldsymbol{L}_{\varphi T}^T & \boldsymbol{L}_{\varphi\varphi}^T \end{pmatrix} \begin{pmatrix} \nabla T(\boldsymbol{x}) \\ \nabla\varphi(\boldsymbol{x}) \end{pmatrix} , \tag{39}$$

where the right hand side is assembled from the gradients of the macro-scale variables. The matrices $\boldsymbol{K}_{TT}$, $\boldsymbol{K}_{T\varphi}$, $\boldsymbol{K}_{\varphi T}$ and $\boldsymbol{K}_{\varphi\varphi}$ have n rows and n columns while the matrices $\boldsymbol{L}_{TT}^T$, $\boldsymbol{L}_{T\varphi}^T$, $\boldsymbol{L}_{\varphi T}^T$ and $\boldsymbol{L}_{\varphi\varphi}^T$ have n rows and 3 columns. $\bar{\boldsymbol{u}}_T^*$ and $\bar{\boldsymbol{u}}_\varphi^*$ are the vectors of temperature and moisture fluctuations on the meso-scale problem.

The multi-scale analysis assumes each macroscopic integration point be connected with a certain mesoscopic problem represented by an appropriate periodic unit cell. The solution of a meso-scale problem then provides instantaneous effective data needed on the macro-scale. Such an analysis is particularly suitable for a parallel computing because the amount of transferred data is small. In this regard, the master-slave strategy can be efficiently exploited [28]. To that end, the macro-problem is assigned to the master processor while the solution at the meso-scale is carried out on slave processors. At each time step the current temperature and moisture together with the increments of

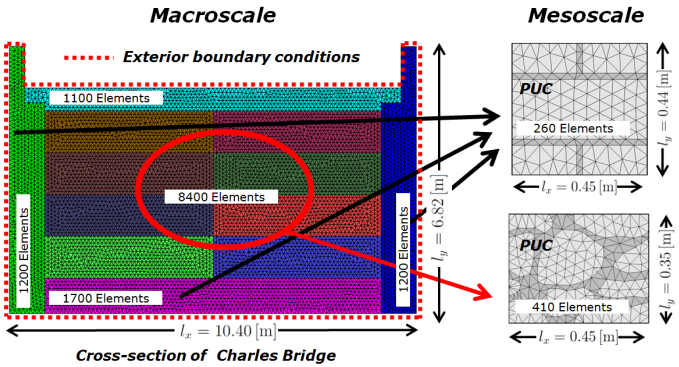

Figure 2: Two-dimensional model: macro and meso scale problems.

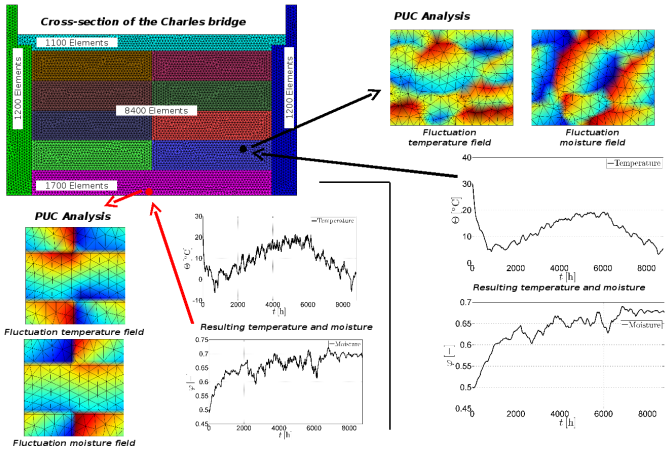

Figure 3: Two-dimensional model: distribution of the temperature and moisture.

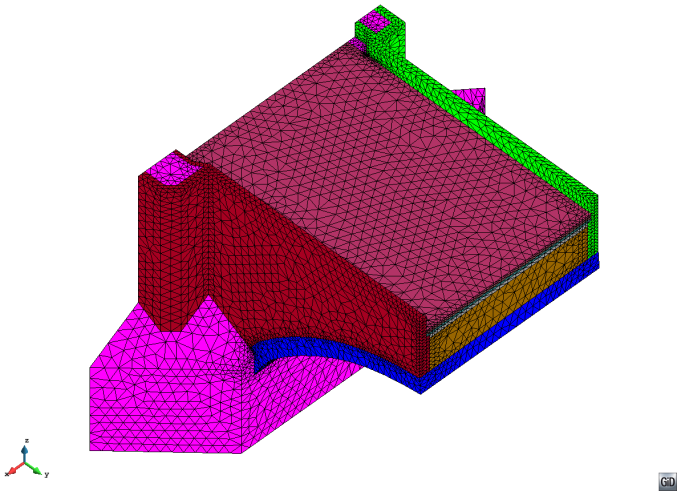

Figure 4: Three-dimensional model: macro-level problem.

their gradients at a given macroscopic integration point are passed to the slave processor (imposed onto the associated periodic cell), which, upon completing the small scale analysis, sends the homogenized data (effective conductivities, averaged storage terms and fluxes) back to the master processor.

If the meso-scale problems are large enough, the ideal solution is to assign one meso-problem to one slave processor. Clearly, even for very small macro-problems with a few thousands of finite elements, the hardware requirements would be in such a case excessive. On the other hand, if the meso-problems are relatively small, i.e. they contain small number of finite elements, the corresponding analysis might be even shorter than the data transfer between the processors. Then, the computational time associated with the data transfer between the master processor and many slave processors may grow excessively. It is worth mentioning that this time consists of two contributions. The first one represents the latency time (the processors make connection) which is independent of the amount of transferred data whilst the second contribution clearly depends on the amount of data being transferred. For small meso-problems it is therefore reasonable to assign several of them to a single slave processor. The master processor then sends a larger package of data from many macroscopic integration points at

Processor No.	1	2	3	4	5	6
No. of meso-problems	1218	1748	1046	1052	1214	1210
Processor No.	7	8	9	10	11	12
No. of meso-problems	1052	1054	1046	1052	1054	1048

Table 1: Decomposition of the macro-problem into sub-domains.

the same time to each slave processor so that the latency time does not play a crucial role. This approach was adopted hereinafter.

Coupled heat and moisture transport in Charles bridge in Prague, Czech Republic, was solved. The meso-scale problem is depicted in Figure 2. The number of elements in the two meso-problems amounts to 265 (160 nodes) for SEPUC 1 and to 414 (239 nodes) for SEPUC 2, respectively. Similarly to the macro-problem, the meso-problems have to account for the material heterogeneity. Clearly, the ideal speedup and load balancing are obtained when the decomposition of the macro-problem reflects the meso-problem meshes. However, this is considerably more difficult when compared to the classical mesh decomposition.

The finite element mesh used at the macro-level is evident from Figure 2 featuring $7,081$ nodes and $13,794$ triangular elements with a single integration point thus amounting to the solution of $13,794$ meso-problems at each macroscopic time step. This figure also shows decomposition of the macro-problem into 12 slave processors. The numbers of elements in individual sub-domains being equal to the number of meso-problems handled by the assigned slave processor are listed in Table 1. It should be noted that the assumed decomposition of the macro-problem is not ideal. In comparison with domain decomposition methods, the macro-problem has to be split with respect to the heterogeneity of the material resulting in the variation of number of elements in individual sub-domains between 1046 and 1748.

The actual analysis was performed on a cluster built at our department. Each node of the cluster is a single processor personal computer Dell Optiplex GX620 equipped with 3.54 GB of RAM. The processors are Intel Pentium with the frequency 3.4 GHz. The cluster is based on Debian linux 5.0 and 32-bit architecture. Each time step took 2.08 minute.

Figure 3 contains distribution of the temperature and relative humidity on the meso-scale problems. Generalization of the model to three dimensions is visible in Figure 4.

4 Conclusions

Two different examples of application of parallel computers in engineering computations were described. The parallel computers are necessary for solution of extremely large problems which are not manageable on a single processor computer. As an example can serve a problem reported in [13], where 314,505,600 unknowns are solved by the FETI method on 4,800 cores. On the other hand, problems with moderate number of unknowns which are solved many times can be found in multi-scale problems [26], optimization problems [22], stochastic problems solved by the Monte Carlo or similar method, fuzzy set problems solved by α-cut methods [17]. They are also very demanding and application of a parallel computer can lead to the ideal speedup.

Acknowledgement

The author thank to colleagues Tomáš Koudelka, Tomáš Krejčí, Michal Šejnoha, Jan Sýkora and Jiří Maděra for help with topic described in this text.

References

[1] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems—scalability results on 1000 processors and discussion of highly heterogeneous problems. *International Journal for Numerical Methods in Engineering*, 47:513–535, 2000.

[2] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. *Acta Numerica*, pages 61–143, 1994.

[3] Z. Dostál, D. Horák, R. Kučera, V. Vondrák, J. Haslinger, J. Dobiáš, and S. Pták. FETI based algorithms for contact problems: scalability, large displacements and 3D Coulomb friction. *Computer Methods in Applied Mechanics and Engineering*, 194:395–409, 2005.

[4] Z. Dostál, D. Horák, and O. Vlach. FETI-based algorithms for modelling of fibrous composite materials with debonding. *Mathematics and Computers in Simulation*, 76:57–64, 2007. doi:10.1016/j.matcom.2007.01.026.

[5] Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, and A. Markopoulos. Scalable TFETI algorithm for the solution of multibody contact problems of elasticity. *International Journal for Numerical Methods in Engineering*, 82:1384–1405, 2010. doi: 10.1002/nme.2807.

[6] C. Farhat, P. S. Chen, J. Mandel, and F. X. Roux. The two-level FETI method-Part II: Extension to shell problems, parallel implementation and performance results. *Computer Methods in Applied Mechanics and Engineering*, 155:153–179, 1998.

[7] C. Farhat and J. Mandel. The two-level FETI method for static and dynamic plate problems-Part I: An optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–151, 1998.

[8] C. Farhat, J. Mandel, and F. X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 115:365–385, 1994.

[9] C. Farhat, K. Pierson, and M. Lesoinne. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer methods in applied mechanics and engineering*, 184:333–374, 2000.

[10] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.

[11] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, USA, 3rd edition, 1996.

[12] P. Gruber, J. Zeman, J. Kruis, and M. Šejnoha. Homogenization of composite with interfacial debonding using duality-based solver and micromechanics. *Computer Assisted Mechanics and Engineering Science*, 16:59–76, 2009.

[13] T. Kozubek, V. Vondrák, M. Menšík, D. Horák, Z. Dostál, V. Hapla, P. Kabelíková, and M. Čermák. Total FETI domain decomposition method and its massively parallel implementation. *Advances in Engineering Software*, 60-61:14–22, 2013.

[14] J. Kruis. FETI method in selected engineering problems. In P. Ivanyi, editor, *Engineering research*, pages 137–145, Pécs, Hungary, 2010. University of Pécs, Pollack Mihály Faculty of Engineering.

[15] J. Kruis and K. Matouš. Domain decomposition of composite laminated plates. In B.H.V. Topping, editor, *Developments in Engineering Computational Technology*, pages 183–189, Edinburgh, UK, 2000. Civil-Comp Press. doi:10.4203/ccp.68.8.1.

[16] J. Kruis, K. Matouš, and Z. Dostál. Solving laminated plates by domain decomposition. *Advances in Engineering Software*, 33:445–452, 2002.

[17] J. Kruis and P. Štemberk. Parallelization of fuzzy problems. In J. Náprstek and C. Fischer, editors, *Engineering Mechanics 2006*, pages 188–189. Institute of Theoretical and Applied Mechanics, AS CR, Prague, 2006. ISBN 80-86246-27-2.

[18] H.M. Künzel. Simultaneous heat and moisture transport in building components. Technical report, Fraunhofer Institute of Building Physics, Fraunhofer IRB Verlag Stutgart, 1995. ISBN 3-8167-4103-7.

[19] J. Mandel. Balancing domain decomposition. *Commun Appl Numer Methods*, 9:233–241, 1993.

[20] M. Papadrakakis, editor. *Solving Large-Scale Problems in Mechanics, The Development and Application of Computational Solution Methods*. John Wiley & Sons Ltd, Chichester, UK, 1993.

[21] M. Papadrakakis, editor. *Parallel Solution Methods in Computational Mechanics*. John Wiley & Sons Ltd, Chichester, UK, 1997.

[22] A. Pospíšilová and M. Lepš. Global optima for size optimization benchmarks by branch and bound principles. *Acta Polytechnica*, 52(6):74–81, 2012.

[23] D. Rixen. Extended preconditioners for the FETI method applied to constrained problems. *International Journal for Numerical Methods in Engineering*, 54:1–26, 2002.

[24] D. Rixen and C. Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering*, 44:489–516, 1999.

[25] B. F. Smith. Domain decomposition algorithms for the partial differential equations of linear elasticity. Technical Report 517, Courant Institute of Mathematical Sciences, 1990.

[26] J. Sýkora, T. Krejčí, J. Kruis, and M. Šejnoha. Computational homogenization of non-stationary transport processes in masonry structures. *Journal of Computational and Applied Mathematics*, 236(18):4745–4755, December 2012. DOI: 10.1016/j.cam.2012.02.031.

[27] J. Sýkora, M. Šejnoha, and J. Šejnoha. Homogenization of coupled heat and moisture transport in masonry structures including interfaces. *Applied Mathematics and Computation*, 219(13):7275–7285, March 2013.

[28] B.H.V. Topping and A.I. Khan. *Parallel Finite Element Computations*. Saxe-Coburg Publications, Edinburgh, UK, 1996.

[29] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, Germany, 2005.

[30] R. Černý, J. Maděra, J. Kočí, and E. Vejmelková. Heat and moisture transport in porous materials involving cyclic wetting and drying. In C.A. Brebbia and G.M. Carlomagno, editors, *Fourteenth International Conference on Computational Methods and Experimental Measurements*, Southampton, UK, 2009. WIT Press.

Jaroslav Kruis – CV

Name:	Jaroslav Kruis
Date of birth:	20 January 1971
Place of birth:	Prague
Address:	Department of Mechanics, Faculty of Civil Engineering Czech Technical University in Prague Thákurova 7, 166 29, Prague 6, Czech Republic e-mail: jk@cml.fsv.cvut.cz tel: +420 224 354 580, fax: +420 224 310 775

Education, Academic and Professional Qualifications

1989-1994	Faculty of Civil Engineering, CTU in Prague Constructions and Transport Engineering (Ing.) Diploma thesis: p-version of the finite element method
1994-1999	Faculty of Civil Engineering, CTU in Prague, doctoral degree programme Constructions and Transport Engineering (Ph.D.) PhD thesis: Solution of large systems of equations on parallel computers
2006	Habilitation (doc.) habilitation thesis: Domain Decomposition Methods for Distributed Computing

Positions

1997-2006	assistant professor, Department of Mechanics Faculty of Civil Engineering, CTU in Prague
2006 - present	associate professor, Department of Mechanics Faculty of Civil Engineering, CTU in Prague

Teaching Activities

Lectures: Dynamics of structures (132DSTD and 132DSK1), Loading and reliability (132LORE and 132ZASP), Reliability of Civil Engineering Structures (132SSK), Durability of materials and structures

(132ZM10).
Lecturer in Ph.D. courses High Performance Computations for Engineering in Pécs, Hungary: 17. - 22. 5. 2009, 7. - 11. 6. 2010 and 29. 5. - 3. 6. 2011.

Scientific Interests

computational mechanics, numerical methods, solution of large systems of linear and nonlinear equations, parallel computing, coupled problems

Professional and Scientific Activities

- WoS: 27 items in WoS, 73 citations, 61 citations without self-citations, h-index 5,
- Scopus: 48 items in Scopus, 97 citations, 58 citations without self-citations, h-index 5,
- 8 papers in journals with impact factor, author and co-author of 165 conference contributions and papers,
- member of editorial board of series of conferences (International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering; International Conference on Civil, Structural and Environmental Engineering Computing; International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering; Central European Symposium on Building Physics),
- member of editorial board of journals Advances in Engineering Software (Elsevier Ltd); Pollack Periodica–An International Journal for Engineering and Information Sciences (Pécs University); Acta Technica Jaurinensis,
- reviewer of Advances in Engineering Software, Pollack Periodica, Computers and Structures, Future Generation Computer Systems, Acta Technica Jaurinensis,
- reviewer for PRACE (Partnership for Advanced Computing in Europe),
- principal investigator of 5 grant projects of the Grant Agency of Czech Republic and one project of the Technological Agency of Czech Republic.

Selected publications:

B. Patzák, D. Rypl, J. Kruis. MuPIF - A distributed multi-physics integration tool. Advances in Engineering Software, Volume: 60-61, Pages: 89-97, 2013, DOI: 10.1016/j.advengsoft.2012.09.005.

J. Sýkora, T. Krejčí, J. Kruis, M. Šejnoha. Computational homogenization of non-stationary transport processes in masonry structures. Journal of Computational and Applied Mathematics, Volume: 236, Issue: 18, 2012, Pages: 4745-4755, DOI: 10.1016/j.cam.2012.02.031.

J. Kruis, T. Koudelka, and T. Krejčí. Multi-physics analyses of selected civil engineering concrete structures. Communications in Computational Physics, Volume: 12, Issue: 3, Pages: 885-918, 2012, September 2012. DOI: 10.4208/cicp.031110.080711s.

P. Šolín, L. Dubcová, and J. Kruis. Adaptive hp-FEM with dynamical meshes for transient heat and moisture transfer problems. Journal of Computational and Applied Mathematics, Volume: 233, Issue: 12, Pages: 3103-3112, 2010, DOI:10.1016/j.cam.2009.07.025.

J. Kruis, T. Koudelka, and T. Krejčí. Efficient computer implementation of coupled hydro-thermo-mechanical analysis. Mathematics and Computers in Simulation, Volume: 80, Issue: 8, Pages: 1578-1588, 2010, DOI: 10.1016/j.matcom.2008.11.010.

O. Medek, J. Kruis, P. Tvrdík, and Z. Bittnar. Static load balancing applied to Schur complement method. Computers and Structures, Volume: 85, Issue: 9, Pages: 489-498, 2007,
DOI: 10.1016/j.compstruc.2006.08.025.

J. Kruis. Domain Decomposition Methods for Distributed Computing. Saxe-Coburg Publications, Kippen, Stirling, Scotland, 2006, ISBN-13: 978-1-874672-23-4.