National Conference with International Participation
ENGINEERING MECHANICS 2001
Svratka, Czech Republic, May 14 – 17, 2001

PARALLEL COMPUTING IN COMPUTATIONAL MECHANICS

Zdeněk Bittnar[*], Jaroslav Kruis, Jiří Němeček, Bořek Patzák, Daniel Rypl

Summary: *The Finite Element Method is a powerful numerical method for solving partial differential equations. It is widely used in many fields in civil, mechanical, biomechanical, aerospace, and electrical engineering. Using this method the engineer can simulate the thermo-chemo-hydro mechanical behaviour of solids, fluids, and structures. In many cases, these calculations are frequently very time demanding, especially when the underlying models are non-linear and three-dimensional. Typical examples are the simulation of crack growth which requires up to several weeks of computing time, and most of the problems in multiphysics. The typical trouble in these cases is ill-conditioning of governing equations. Since several of these simulations are required to evaluate a numerical model or structural design, it is necessary to speed up the computations. An attractive way to achieve this speed up is the use of parallel algorithms.*

1 Introduction

After opening the market with electronic computers civil engineers were among pioneers utilizing this technology preferably in the area of structural mechanics. The reason was very simple. The slope - deflection method was widely used by practicing engineers for the analysis of frame structures. The method was appropriate for simple algorithmization and for coding. There was developed a lot of codes for the analysis of frame structures. Several years after introducing computers on the market engineers discovered finite element method, at the moment general tool for the analysis of problems in civil, mechanical, electrical, chemical and environmental engineering. The development of the first systematic design tool was also connected with civil engineers. ICES (Integrated Civil Engineering System) was developed on MIT. The real milestone in increasing the popularity of computer method in engineering was the introduction of the first PC computer with advanced graphical opportunity. During that elapsed time the things changed significantly. The power of nowadays standard PC's is very comparable to the power of former biggest mini and mainframes. Now it is very difficult to imagine any design without computer support.

The aim of this paper is to show the opportunity how to use parallel technology in computational mechanics in civil engineering applications in the near future with the special attention to the cluster technology.

The solution of complex sophisticated problems to model various phenomena with sufficiently high accuracy and in reasonable time makes the parallel processing attractive for a large family of applications, including structural analysis. However it is important to realize that most of traditional algorithms are inherently not suitable for parallelization because of their development for sequential processing. The most natural way for parallelization is the decomposition of the problem being solved in time or space. The individual domains are then mapped on individual processors and are solved separately ensuring the proper response of the whole system by appropriate communication between the domains. An efficient parallel algorithm requires a balance

---

[*] Prof. Ing. Zdeněk Bittnar, DrSc., Czech Technical University in Prague, Faculty of Civil Engineering, Thákurova 7, 166 29, Prague; phone: ++420 2 2435 3869, e–mail: bittnar@fsv.cvut.cz

of the work (performed on individual domains) between the processors while maintaining the interprocessor communication (typical bottleneck of parallel computation) at a minimum.

Since the last decade the parallel computation has become quite feasible due to the following three aspects. Firstly, a lot of new algorithms, suitable for parallel processing, have been developed (including efficient algorithms for domain decomposition). Secondly, the parallel computation ceased to be limited to parallel supercomputers (equipped with high technology for even higher price) but can be performed on ordinary computers interconnected by network into computer cluster. Such a parallel cluster can even outperform the supercomputers (as IBM SP2, SGI Origin etc) while keeping the investment and maintenance costs substantially lower ! And thirdly, several message passing libraries (typically MPI, PVM), portable to various hardware and operating system platforms, have been developed, which allows to port the parallel applications almost to any platform (including multiplatform parallel computing cluster).

2 Material Modeling

Modeling of reinforced concrete (RC) structures is a problem having complex character. It involves both modeling of concrete and reinforcing steel and also the interaction of these materials. As it is widely known, concrete belongs among so called quasi-brittle materials. Its behaviour is strongly influenced by a stress state to which it is exposed. The behaviour of concrete varies from brittle to very ductile according to the lateral confinement which can be provided either by outside constraints or by transversal reinforcement, which is the common case. All types of transversal reinforcement such as different ties, stirrups or steel sections can be used to provide this lateral confinement. The change of concrete behaviour according to lateral confinement yields the need of triaxial modeling of concrete. Without respecting the triaxial behaviour of concrete one cannot describe the reality of RC structures. There is also one very important feature of concrete. General loading cases lead to the phenomenon called softening. Softening of concrete is characterized by a progressive loss of material integrity, which yields the descending load-deflection diagram.

Modeling of reinforcing steel must include elastoplastic behaviour of the material. Longitudinal reinforcement is usually placed near the surface of concrete and fixed laterally by transversal reinforcement. If the RC structure is subjected to the compressive load the longitudinal reinforcement can buckle. Thus, the model of longitudinal reinforcement must include also the possibility of buckling of steel.

What are the material models capable to describe triaxial nonlinear behaviour of concrete? Well, there is not much choices. Classical models are based mainly on the theory of plasticity. Theory of plasticity was firstly developed for modeling of metals, but it was also enlarged for modeling of concrete (see e.g. [1, 2, 3]). Theory of plasticity provides nonlinear description of concrete including loading, unloading and path dependence. The key point of these models is the definition of yield condition which is usually formulated in the stress space. This formulation can be very complicated and moreover multidimensional formulation can be hardly imagined and physically interpreted. Another type of models are based on continuum damage mechanics. These models are able to describe material stiffness degradation according to the certain damage parameter, which can be defined as a single scalar parameter or a tensor of higher orders. A popular damage model was proposed by Mazars in [4]. Some models combine the attitude of plasticity and damage mechanics (see [5]).

All these types of models suffer from some kind of insufficiency. One of the major errors can be caused by not respecting of anisotropy development within the material microstructure. Deficiency of these models is that they are usually derived in principal strain space. They do not respect the rotation of principal axes during loading process that can lead to big errors. Especially description of softening is very sensitive to it, because damage development changes the material from isotropic to highly anisotropic. The right way to solve this problem is to link damage with its orientation in the material and compute the material response directly for this concrete orientation. This is done so in concept of microplane model [6, 7, 8]). Final response is then given by combination of the responses from different orientations.

The classical approach to the constitutive modeling is based on a direct relationship between strain and stress tensors and their invariants. In contrary to it, constitutive relations of microplane model are formulated in terms of strain and stress components on planes of arbitrary spatial orientations, so called microplanes. This attitude excels in conceptual simplicity and allows straightforward modeling of anisotropy and other processes connected with planes with different orientations. The penalty to be paid is a great increase in computational effort. The relationship between micro and macro level is obtained by projecting strain tensor to the particular microplanes (so called kinematic constraint) or by projecting stress tensor (static constraint). Then constitutive relations between microstrains and corresponding microstresses are evaluated. The missing link (between microstresses and macrostress for kinematic constraint and between micro and macro strain for static constraint, respectively) is obtained by application of principle of virtual work. That kind of material model is capable to describe triaxial nonlinear behaviour of concrete including tensional and compressive softening, damage of the material, different types of loading, unloading or cyclic loading.

3 Parallel Computing

Typical engineering design offices are equipped with several PC computers with different processor power, memory available and disc capacity. The computers are usually connected with fast Ethernet. The computing performance is relatively high with respect to the demands of design process, involving CAD, structural analysis etc. This makes the offices well equipped for high performance computing provided relevant parallel software is available.

Parallel computing is nowadays considered a very efficient tool to overcome bottlenecks of traditional serial computing. These bottlenecks relate to both lack of resources (memory, disk space, etc.) and long computational times. Typical parallel application decreases the demands on memory and other resources by spreading the task over several mutually interconnected computers and speeds up the response of the application by distribution of the computation to individual processors. Note however that parallel computing is worth also for applications that require almost no resources but consume an excessive amount of time and for applications that cannot be performed on a single (even well equipped) computer regardless of the computational time. It is important to realize that from engineering point of view the scalability of the algorithm is not the only criterion to judge efficiency of parallel application. In many cases, the ability to analyze extremely large problems not solvable on individual machine is of primary interest.

3.1 Parallel Computers

According to Foster [19], a parallel computer is a collection of processors that are able to work cooperatively to solve a computational problem. Thus, a parallel computer can be a large special computer consisting of many processors or a cluster of cheap PCs connected by the Internet. At the moment, the parallel computer hardware is divided into three categories

- shared memory computers,
- distributed memory computers, and
- virtual shared memory computers.

3.1.1 Shared Memory Computers

A shared memory computer, sometimes referred to as Symmetrical Multi-Processor (SMP) architecture, consists of multiple processors sharing the same memory. Since the capacity of the communication bus is shared between all the processors, the memory access time increases with the number of processors. To do this process more effective, each processor has fast cache memory (very expensive, mostly more then processor itself) which, if properly used, significantly reduces the communication time. Nowadays, the most attractive SMP are two ways PIII processors from Intel. The increase of price with number of shared processor is highly nonlinear and from the point of computational mechanics non-effective (this can be demonstrated on two clusters at Cornell - cluster with 2ways PIII processors is faster than cluster built on 4ways PIII Xeon processors, which are significantly more expensive).

3.1.2 Distributed Memory Computers

A distributed memory computer consists of multiple processors, each having its own memory. Because the processors can access only their own memory, they have to use for exchange of data communication network. This network can be a special high-speed network (e.g. high performance switch in IBM SP2 machine) or general purpose network as fast Ethernet or even in near future the Internet (so called P2P architecture). The performance of the network communication is given by the latency and the bandwidth. The latency is the time necessary to start an interaction between two processors and the bandwidth is the number of bytes that can be transfered via the network within one second. Since a distributed memory computer has no shared resources like a bus, the number of processors is virtually unlimited. The today three largest computers have thousand processors

- ASCI White - IBM - 8192 processors,
- ASCI Red - Intel - 9632 processors,
- ASCI Blue Pacific - IBM - 5880 processors.

This is why distributed memory computers are called Massively Parallel Processors (MPP).

3.1.3 Virtual Shared Memory Computers

This type of parallel computer combines features of a shared memory computer with that of a distributed memory computer. It has a memory architecture that is physically distributed, but logically shared. The connectivity is often called as NUMA (Non-Uniform Memory Access). Typically, the number of processors ranges from two to a few hundred.

3.2 Parallel Programming Models

There are two basic types of parallel applications - single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD). While in the former case the same code is working on different (and distributed) data, the latter one is based on several codes working on different data. The parallelization in SIMD model can be based on standards as Open MP or HPF (High Performance Fortran). However, their application is limited to shared memory multiprocessors and their efficiency is generally not high (depending on the underlying problem and data distribution). On the other hand, the parallelization of MIMD models is based on message passing standards as MPI or PVM. The efficiency of these applications is dependent on the actually adopted parallel computing paradigm. The advantage of the MIMD model is that it works on shared memory computers as well as on massively parallel platforms.

4 Examples

The examples to be presented were (usually) performed on two different parallel hardware platforms - PC cluster (installed at the department of Structural Mechanics) and IBM SP2 and SP machines (installed at CTU computing centre).

Originally, the PC cluster consisted of four workstations DELL 610, each equipped with two processors. Two workstations contain dual PII Xeon processors at 450 MHz with 512 MB of shared system memory and the remaining two comprise dual PII Xeon processors at 400 MHz with 512 MB of shared memory. The workstations are connected by Fast Ethernet 100 Mb network using 3Com Superstack II switch, model 3300. Later on two additional workstations, equipped with dual PIII processors at 450 MHz with 760 MB and 1 GB of shared system memory, were added to the cluster. Note that this cluster represents a heterogeneous parallel computing platform with the combination of shared and distributed memory.

All workstations can be running either Windows NT 4.0 operating system, in which case the communication is based on MPI/Pro for Windows NT message passing library (MPI Software technology, Inc.) that supports both the distributed and shared memory communication, or under Linux 6.x operating system with public domain MPICH message passing library.

The IBM SP2 is a heterogeneous machine equipped with 22 P2SC processors running at 120 and 160 MHz, each one with at least 128 MB of memory and 2 GB of disk space. The processors, running AIX 4.1 operating system, are interconnected with HPS (high performance switch - 40 MB/s bidirectionally).

The IBM SP is also a heterogeneous machine equipped with 4 nodes with 4 Power3 processors, running at 332 MHz and having at least 1 GB of shared system memory and 32 GB of disk space, and with 8 nodes with 2 Power3 processors (optimized for floating point operations), running at 200 MHz and having at least 1 GB of shared system memory and 16 GB of disk space. The processors are running AIX 4.3 operating system and are interconnected with SPS (super performance switch).

Both machine are (SP2 and SP) are interconnected by standard Ethernet. The communication is based on MPI message passing library built on the top of the native MPL message passing library.

4.1 Parallel Mesh Generation

The mesh generation utilizes the tree-based approach and is designed for parallel processing on memory distributed computing platforms [15, 16]. The parallelization strategy is based on the domain decomposition concept. Two levels of the domain decomposition have been considered - the model level and the model entity parametric tree level. The discretization is accomplished by application of templates fitted into the cells of a generalized parametric tree data structure built over individual model entities. The compatibility of tree structures on the processor interface is ensured by an iterative process. The actual parallel computing scheme is based on the master and slaves parallel paradigm. Since a dynamic load balancing mechanism is employed, an even distribution of the work load among the processors is ensured. A very favourable ratio between the computation and communication has been achieved and a considerable speedup has been evidenced. The algorithm has been successfully implemented on several parallel computing platforms - IBM SP2, IBM SP, Transtech Paramid and Dell PC cluster.

Two different message passing libraries have been used for the implementation of communication: i) MPI (Message Passing Interface) and ii) Parmacs (Parallel Macros). MPI is the primary message passing library used for implementation on IBM SP2, SP and PC cluster. Since MPI is not available on Transtech Paramid machine, Parmacs has been chosen as an alternative message passing library.

To demonstrate the parallel performance of the algorithm a set of examples is presented - a chair (Fig. 1), a mechanical joint (Fig. 2) a junction of two pipes (Fig. 3). The chair has been discretized by two uniform two-dimensional meshes. The smaller one contains 338.512 nodes and the larger one 1.045.504 nodes. Similarly, two uniform three-dimensional meshes comprising 152.186 and 518.929 nodes, respectively, have been generated to discretize the model of the junction. The mechanical joint has been discretized by four three-dimensional meshes. Two uniform meshes contain 162.300 and 524.979 nodes, respectively, and two graded meshes comprise 165.771 and 574.053 nodes, respectively. Since the meshes are generally too large with

Figure 1: Mesh of a chair.

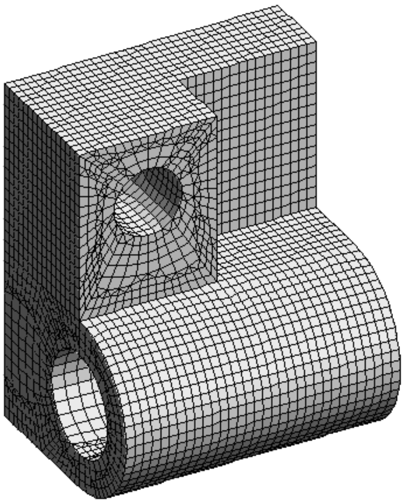

Figure 2: Mesh of a joint.

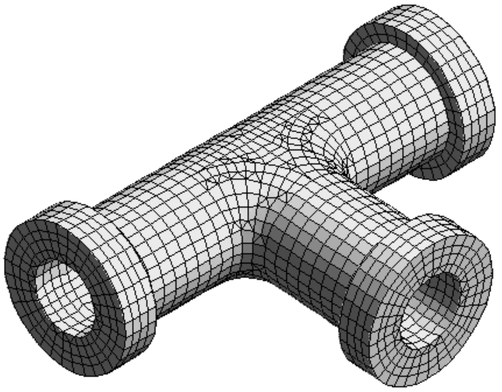

Figure 3: Mesh of a junction.

respect to the memory available on Transtech Paramid machine, only the results from IBM SP and PC cluster are presented. Since the master and slaves parallel computing scheme has been adopted, a separate processor must be allocated for the master process. Note however that the master processor has not been considered in the evaluation of the speedup and efficiency. This is affordable because it has been verified (on IBM SP2 only) that master process can be running together with one slave process on the same processor without impact on the performance. Despite this fact, the speedup on PC cluster was evaluated only up to slave processors due to the license restrictions limiting the total number of processes to 8 (under Windows NT). The execution times and speedups for individual meshes, hardware and software platforms are summarized in Figs 4 - 11. Note that the speedup on PC cluster is always evaluated using the single processor time obtained on the faster processor (450 MHz). This results in a slightly underestimated speedup if a slower processor was also involved in the computation.

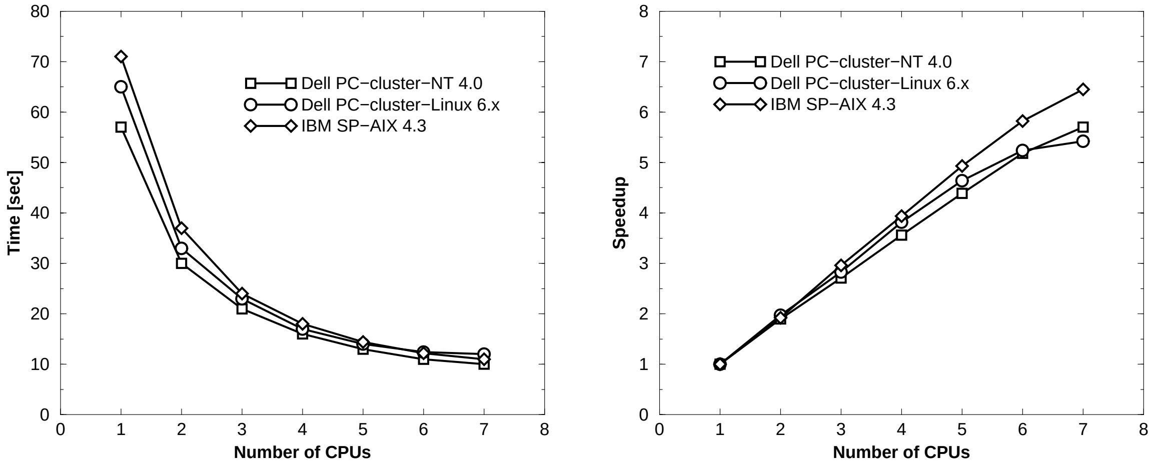

Figure 4: Time and speedup for chair (338.512 nodes).

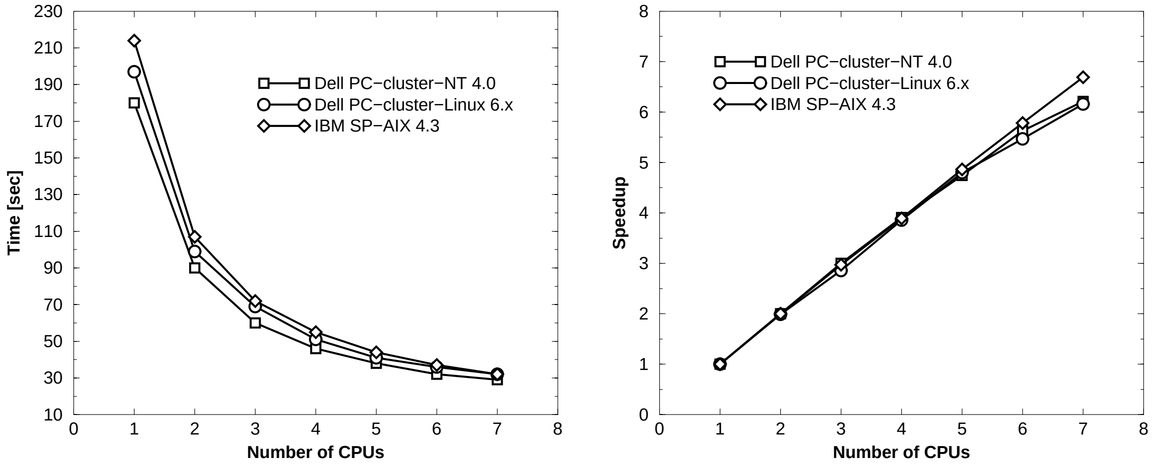

Figure 5: Time and speedup for chair (1.045.504 nodes).

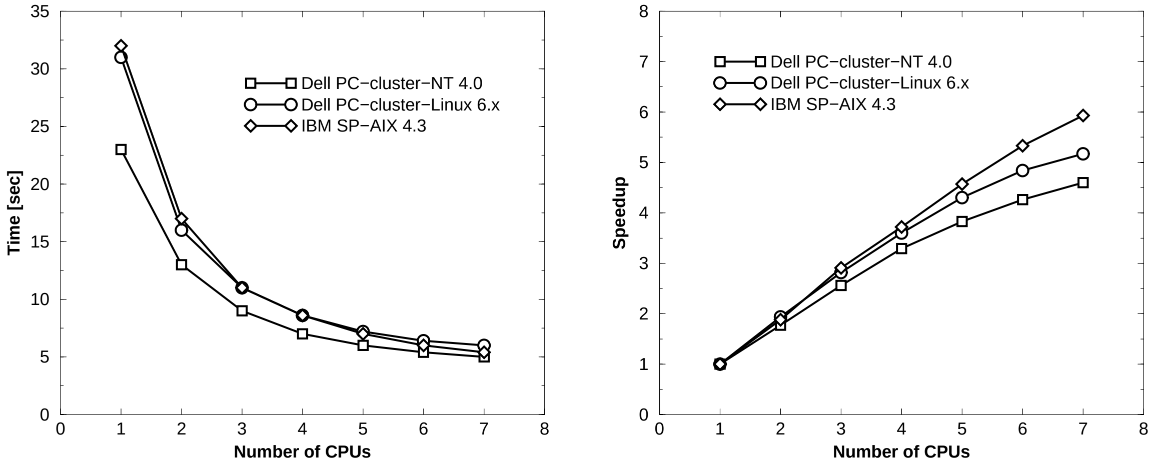

Figure 6: Time and speedup for junction (152.186 nodes).

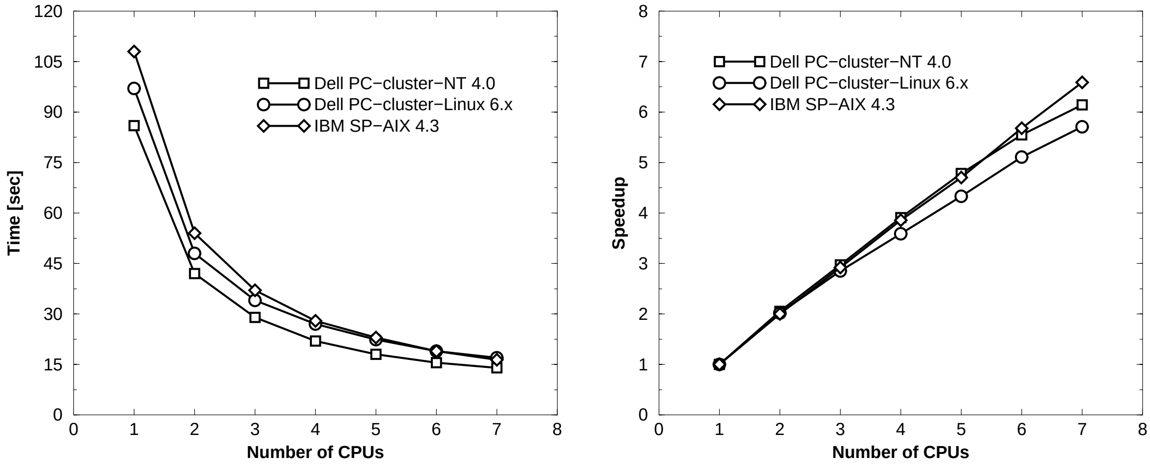

Figure 7: Time and speedup for junction (518.929 nodes).

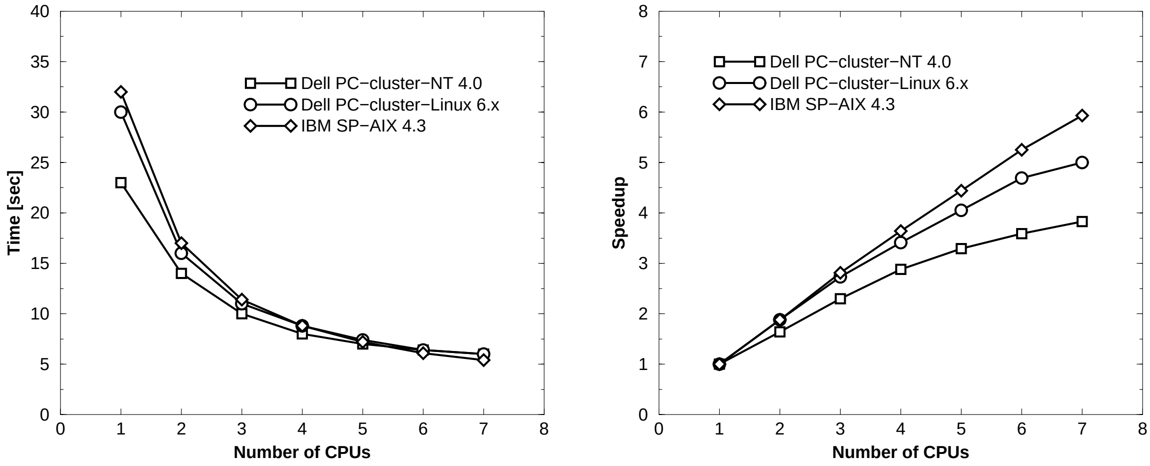

Figure 8: Time and speedup for joint (uniform - 162.300 nodes).

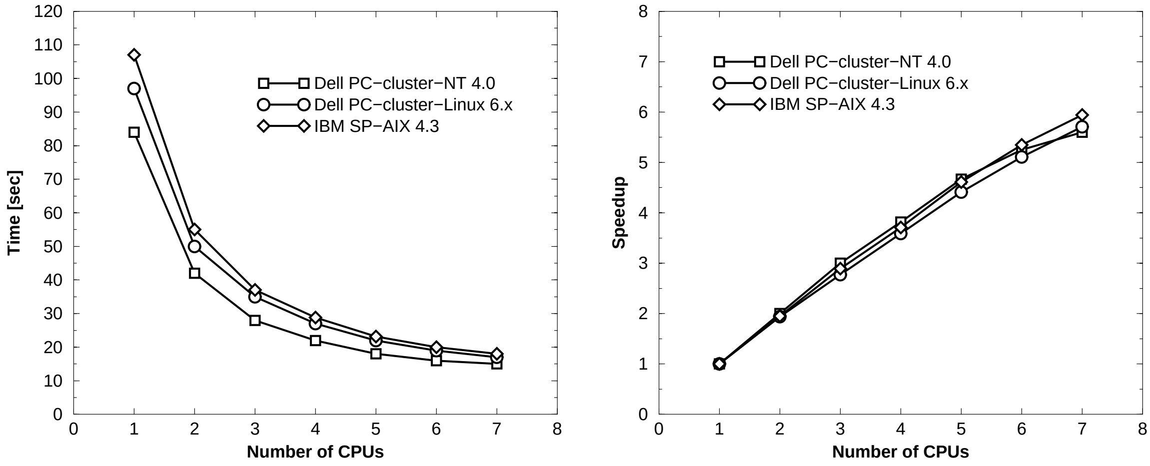

Figure 9: Time and speedup for joint (uniform - 524.979 nodes).

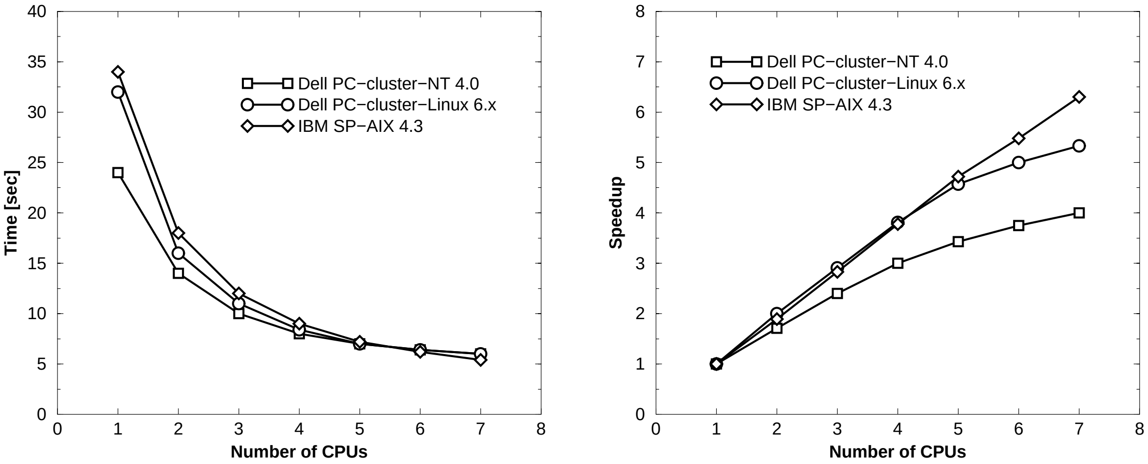

Figure 10: Time and speedup for joint (graded - 165.771 nodes).

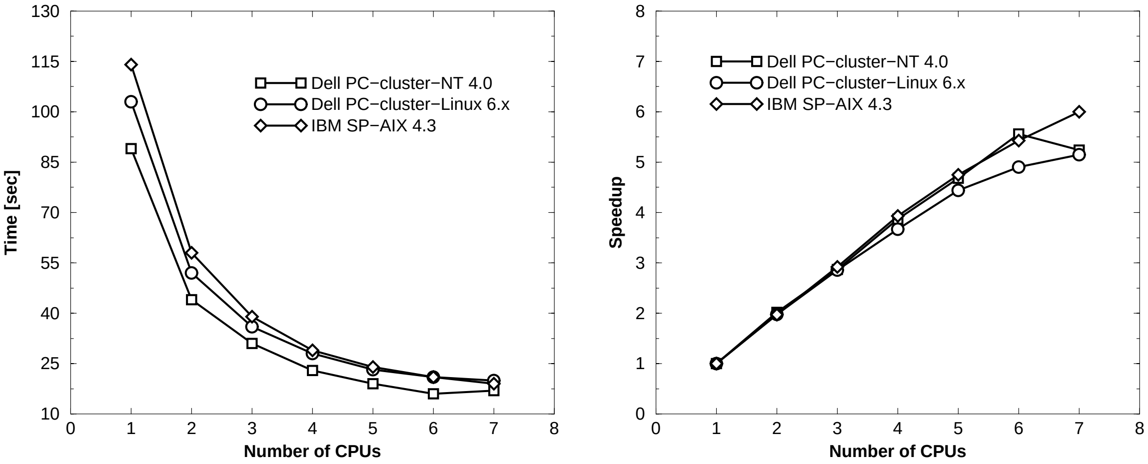

Figure 11: Time and speedup for joint (graded - 574.053 nodes).

4.2 Nonlinear Parallel Analysis by Explicit Algorithm

A 3D notched specimen has been analyzed in three-point-bending using the direct explicit integration. The specimen geometry is shown in Fig. 12. Initially, the employed constitutive model is a nonlocal variant of rotating crack model. Once cracking process reaches a certain critical state (identified by principal stress to tensile strength ratio and by current shear stiffness to shear modulus ratio), the procedure switches to a damage type formulation. The final stage is then described by the damage model, that uses the anisotropic stiffness multiplied by a scalar factor, that decays to zero value as the cracking continues. The constitutive properties are summarized in Table 1. In order to simulate static test, the specimen loading has been controlled by the prescribed displacement of a transverse edge in the middle of the top specimen surface, which has been determined from the requirement of minimal inertia forces [12]. The mesh contains 1964 nodes and 9324 linear tetrahedral elements. The total number of time steps analyzed is 7500. The modified node-cut strategy (allowing for nonlocal material model) has been used.

The mesh partitioning implementation is based on METIS partitioning library. A general front-end application to METIS serving simultaneously as a data converter between the (sequen-

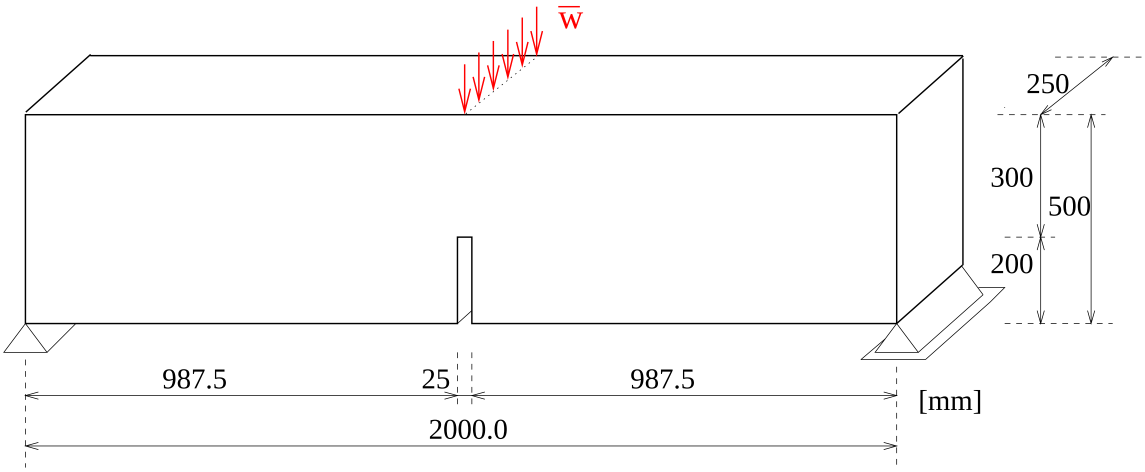

Figure 12: Geometry, support and loading of specimen for 3-point bending test.

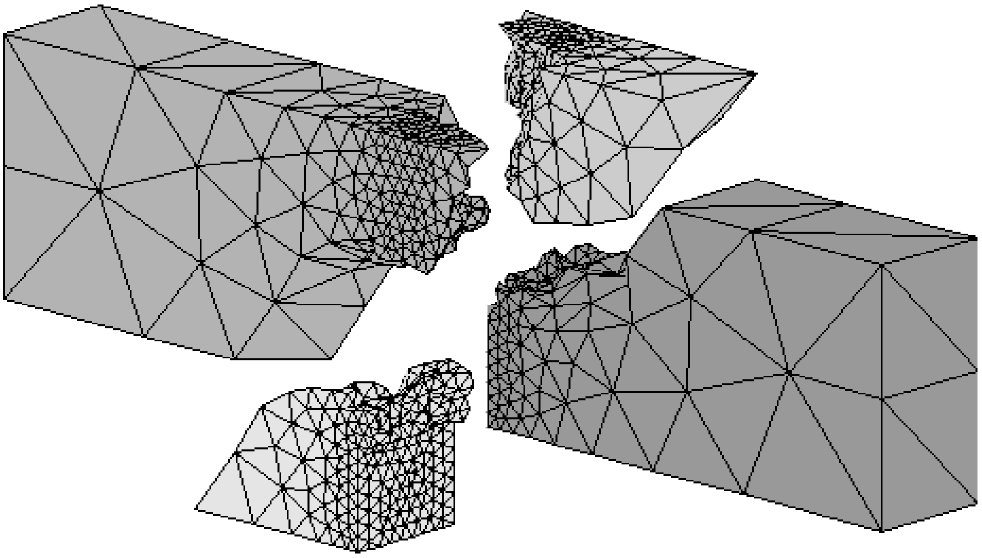

Figure 13: Decomposition of the specimen into 4 partitions.

tial) mesh generator [16] and the object oriented computational code [14] has been written. This application firstly transforms the general mesh into an appropriate graph structure, according to the selected cut strategy. A METIS graph partitioning routine is then used to obtain the mesh partitioning which is further modified to account for zones involved in averaging algorithms. The partitions have been generated prior the analysis and have been kept constant throughout the whole analysis (static load balancing). An example of domain decomposition for 4-processor analysis is depicted in Fig. 13.

The results achieved on Dell PC cluster and SP2 machine are presented in Figs 14 and 15, respectively. Note that the heterogeneity of the computing platforms has been taken into account neither in the mesh partitioning (all partitions are equally load balanced) nor in the speedup or efficiency evaluation. Since the single processor computation has been always performed on the most powerful processor, the speedup is slightly underestimated whenever a slower processor

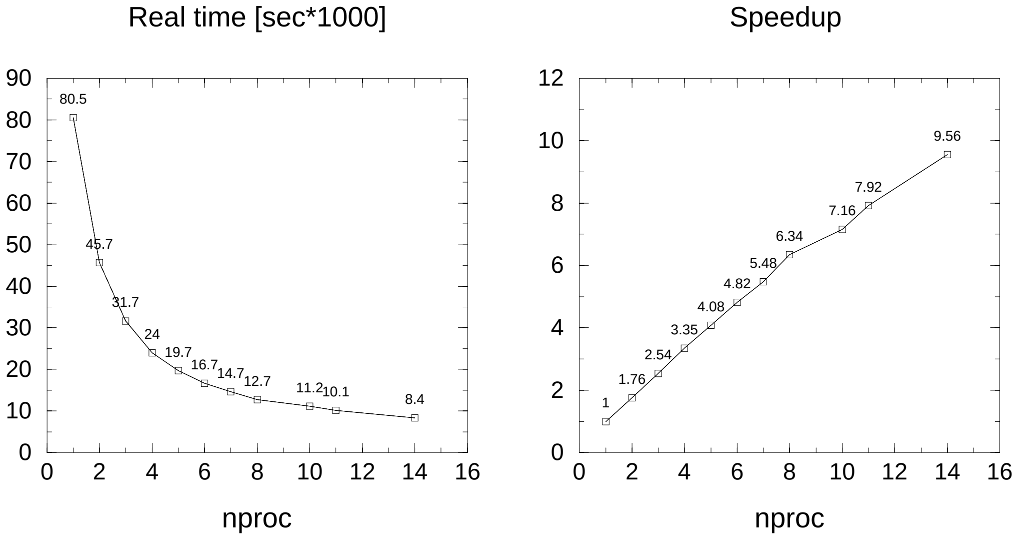

Figure 14: Timing and speedups for 3-point bending test on SP2 machine.

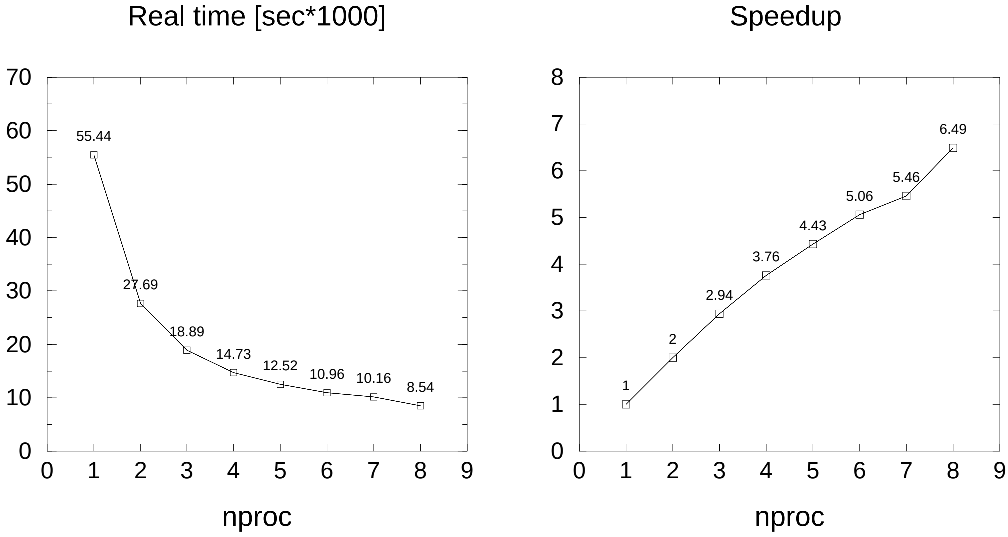

Figure 15: Timing and speedups for 3-point bending test on PC cluster.

has participated in the calculation. The degradation of the speedup profile is also caused by the adopted static load balancing. Since the computational complexity at some regions is increasing

Density	2.5	kN/m^3
Young's Modulus	20.0	GPa
Poisson's ratio	0.2	
Tension strength	2.5	MPa
Ultimate tension deformation	0.0005	
Radius of nonlocal aver. zone	0.025	m
Shear stress transition coef.	0.6	
Normal stress transition coef.	0.0	

Density	2.5	kN/m^3
Young's modulus	60.0	GPa
Poisson's ration	0.18	
k1 (M4-param)	0.000095	
k2 (M4-param)	160.0	
k3 (M4-param)	10.0	
k4 (M4-param)	150.0	

Table 1: Material properties for 3-point bending tests (non-local rotating crack model with transition to scalar damage - left, microplane M4 model - right).

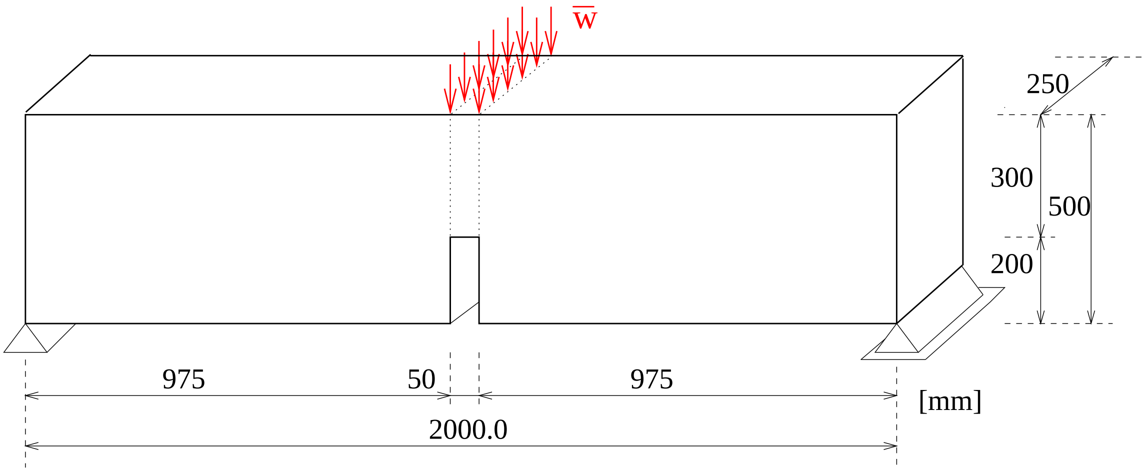

Figure 16: Geometry, support and loading of specimen for 3-point bending test.

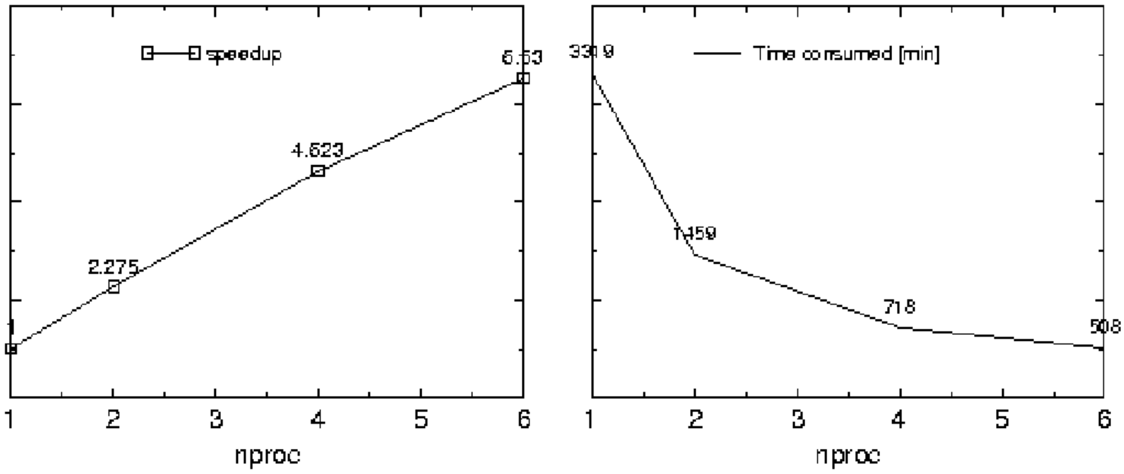

Figure 17: Timing and speedups for 3-point bending test on PC cluster.

considerably during the analysis (strain-softening), the load balance is disturbed, resulting in the less loaded processors to be idle. This effect is becoming more significant as the number of processors increases. Despite these facts, the achieved speedup and efficiency are significant, leading to considerable reduction of the computational time.

The same problem was solved using the microplane material model M4. The model geometry (see Fig. 16) was slightly modified in order to enable the use of uniform structured mesh. This is necessary to ensure the material properties (summarized in Table 1) to be the same for each element, otherwise separate fitting procedure would be required for each element to specify appropriate material properties. Note that the dependence of material properties can be eliminated by introduction of nonlocal version of microplane model. Again, the static loading was simulated by prescribed displacement of transverse edges on top of specimen surface.

The structured mesh contains 2772 nodes and 2030 linear brick elements (each with 8 integration points). The analysis has been performed using 7500 time increments. The achieved computation times and speedups on PC cluster are presented in Fig. 17. Note that superlinear speedup has been achieved, which can be explained by enlarged amount of available cache and by preserving computation to communication ratio at high values.

4.3 Parallel Implicit Analysis of Composite Plates

In contrast to explicit methods, implicit algorithms require solution of system of algebraic equations. Since there occur many degrees of freedom, thanks to complexity and size of the models, usual hardware is not able to solve those huge systems. This makes parallel computers together with parallel algorithms very promising.

When a refined Mindlin-Reissner theory is employed for description of each layer of a laminated composite material, the matrix of resulting system of equations is sparse and exhibits regular sparsity pattern. However, it is easy to observe that its solution by a variant of the Gauss elimination is not efficient due to the fill-in [13]. For example, only the 6-layer composite plate discretized by mesh of 16x16 elements may be treated on a computer with 128 MB memory. Moreover, since the matrix of the system is indefinite, it is difficult to find an efficient preconditioner for application of the standard iterative methods. The refined Reissner-Mindlin theory in connection with the finite element method leads to special form of the resulting matrix which is created from blocks. However, one diagonal block contains only zero components and this causes difficulty for solution methods. Similar matrix occurs in Finite Element Tearing and Interconnecting method (FETI).

The FETI method was introduced by Farhat and Roux in [9]. The idea of the method is based on elimination of displacements from equilibrium conditions and substituting them to the compatibility conditions which are expressed with the help of localization matrices. Non-supported subdomains require the definition of pseudoinverse matrices. The rigid body motions of the non-constrained subdomains are used for global data interchange. The linear combination of rigid body motions must be added to the displacements with respect to solvability conditions on particular subdomains.

The FETI method was implemented in parallel environment and successfully applied for

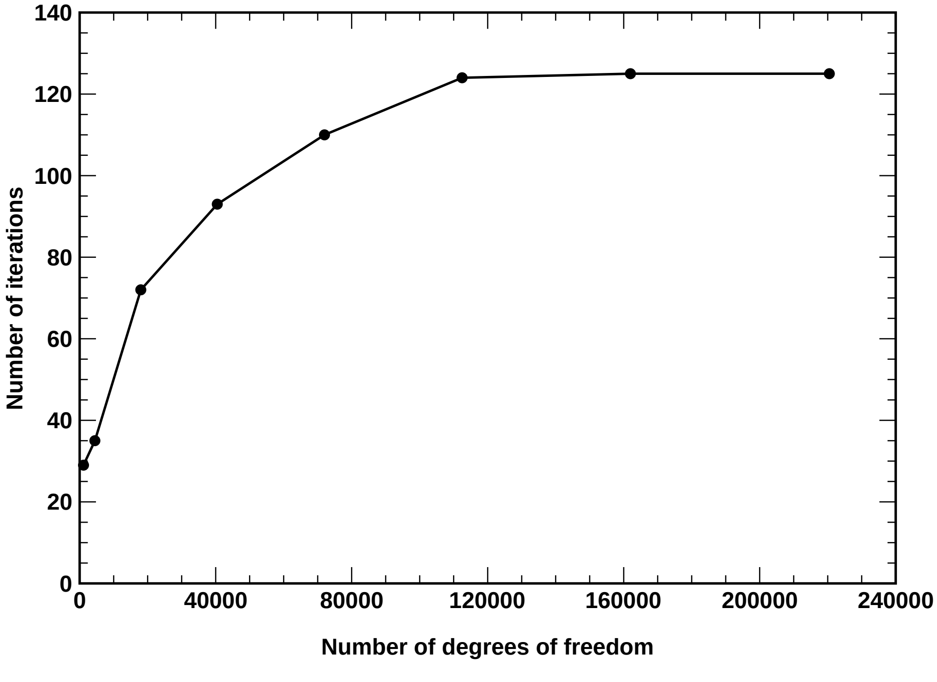

Figure 18: Number of iterations vs. number of DOFs for laminated rectangular plate.

solution of composite laminated plates and shells. Excellent behaviour was observed in analysis of composite laminated plates. The most interesting result consists in the fact that almost the same number of iterations was evidenced for rectangular plate with 120.000 degrees of freedom and more as it is depicted in Fig. 18.

The analysis of another composite laminated plate with layup $[0, 60, 90]_s$ is presented as the second example. The unstructured mesh of the domain (single layer) is illustrated in Fig. 19. The domain is clamped on the external perimeter and subjected to point load at the “corner” nodes. A special layer-based domain decomposition is adopted resulting in a separate domain for each layer. Each of the 6 plies of the composite is made of aligned T-50 graphite fibers bounded to the 6061 Aluminum matrix with volume fraction 0.5. The individual material properties are listed in Table 2. The overall properties are obtained by the Mori-Tanaka averaging method [10, 11]. The calculation was performed on PC cluster using 6 processors. The dependence of the number of iterations on number of DOFs is sketched in Fig. 20. Note, that with increasing size of the problem the number of iterations grows only polylogarithmically.

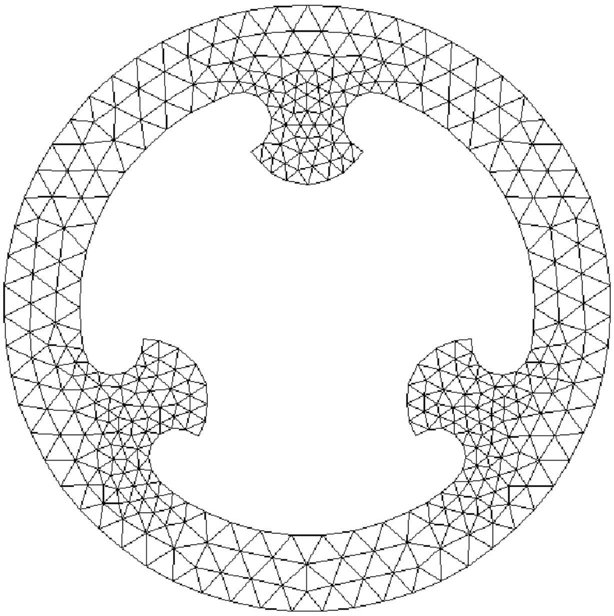

Figure 19: Mesh of the composite plate domain.

Material	E_L [kPa]	E_T [kPa]	G_L [kPa]	G_T [kPa]	ν
T-50 graphite	3.864	7.60	15.2	2.60	0.41
6061 Aluminum	7.250	7.25	2.73	2.73	0.33

Table 2: Material properties of composite plate.

5 Conclusions

Parallel computing can be viewed as a very efficient tool to solve large and complex problems either in order to speed up the response or to overcome lack of resources. Since the last decade the parallel computation has become quite feasible even for small design offices that can create a parallel environment by interconnecting individual workstations into a cluster by standard Ethernet (10 or 100 Mb/s). Such a cluster could even outperform parallel supercomputers

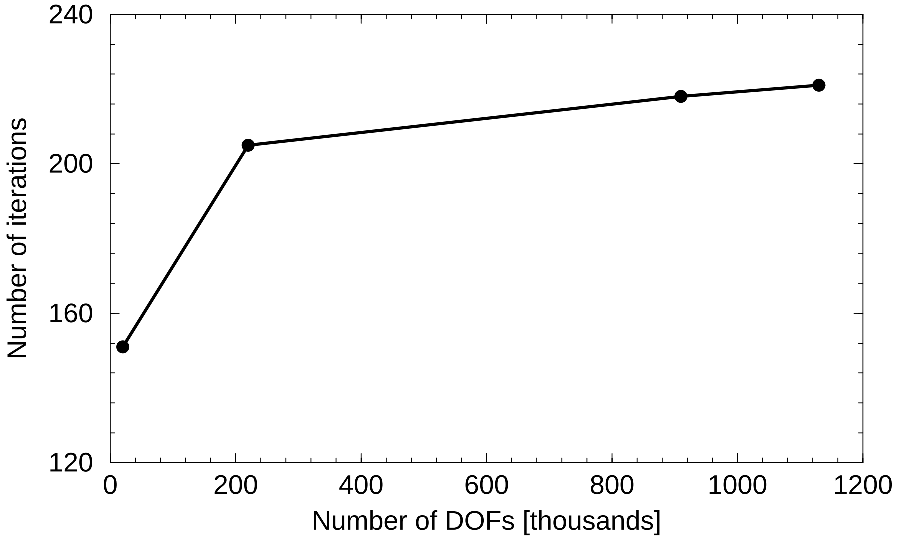

Figure 20: Number of iterations vs. number of DOFs for composite plate.

at substantially lover investment and maintenance cost. The state-of-the-art message passing libraries enables to port the parallel application to various combinations of hardware and software platforms.

5.1 Future Work

A lot of very sophisticated material models was developed during last few decades. Such complicated events as cracking, damage and fracture of concrete can be simulated on powerful workstations. Despite the fact that actual power of PCs based on newest PIII processors is very high, demanding simulations take dozens of hours of computer time.

Let us present the following example for demonstration. It includes experimental investigation and computational simulation using microplane model. The experimental program was focused on RC column loaded by compressive load with a small eccentricity (see Fig. 21). Especially the post-peak behaviour, size effect and confinement effect of stirrups were studied and the following conclusions have been drawn:

- No significant yield plateau was observed in force-deflection diagrams. Due to the compressive softening of concrete, specimens lost their load-bearing capacity very fast. See Fig. 21.
- Longitudinal compressive load causes high transversal expansion of concrete and leads to yielding of stirrups.
- The longitudinal density of stirrups influences the ductility and load capacity of RC columns.

Numerical studies were focused on 3D modeling of RC columns using finite element method. The nonlinear zone of the column was modeled by microplane model for concrete. Geometrically nonlinear element were used for longitudinal steel reinforcement in order to capture buckling of steel. The finite element model consisted of 4896 linear elastic space elements, 3456 nonlinear space elements including microplane model and 1619 nonlinear 3D beam elements (reinforcing steel). Explicit time integration method was used. The computation was performed on a single processor PC - PII-Xeon 400 MHz, 512 MB, where 6000 time increments lasted for 61.65 hours. The computational effort needed for model with microplane material is evident, even if explicit

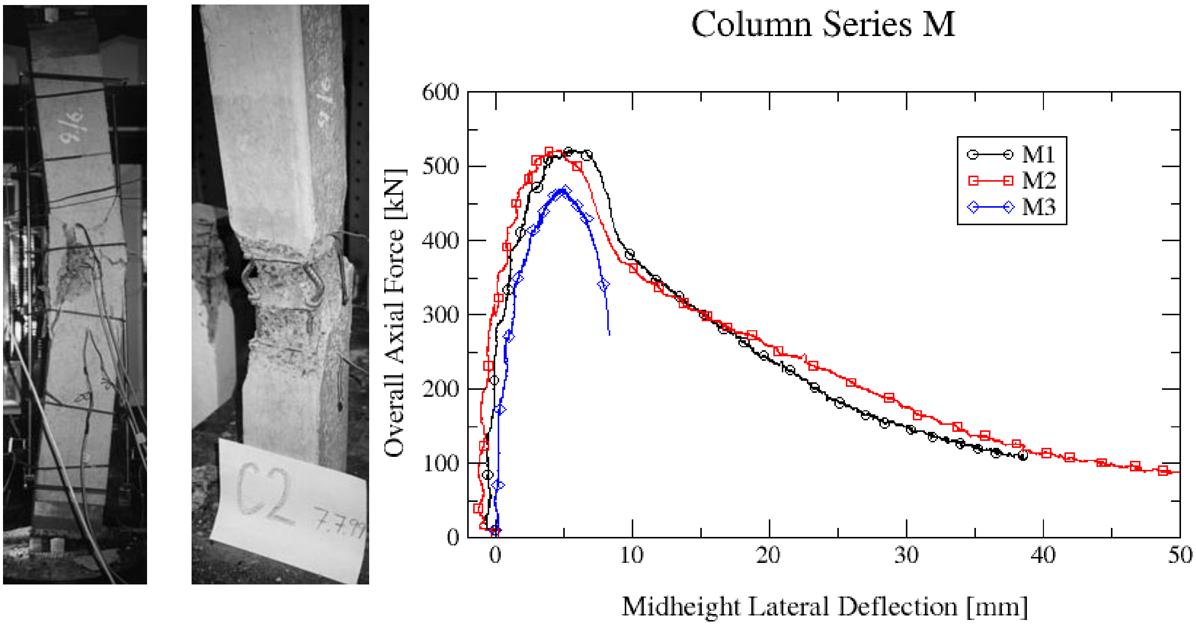

Figure 21: Experimental arrangement (left), typical breakdown of the column (middle), load-deflection diagram (right).

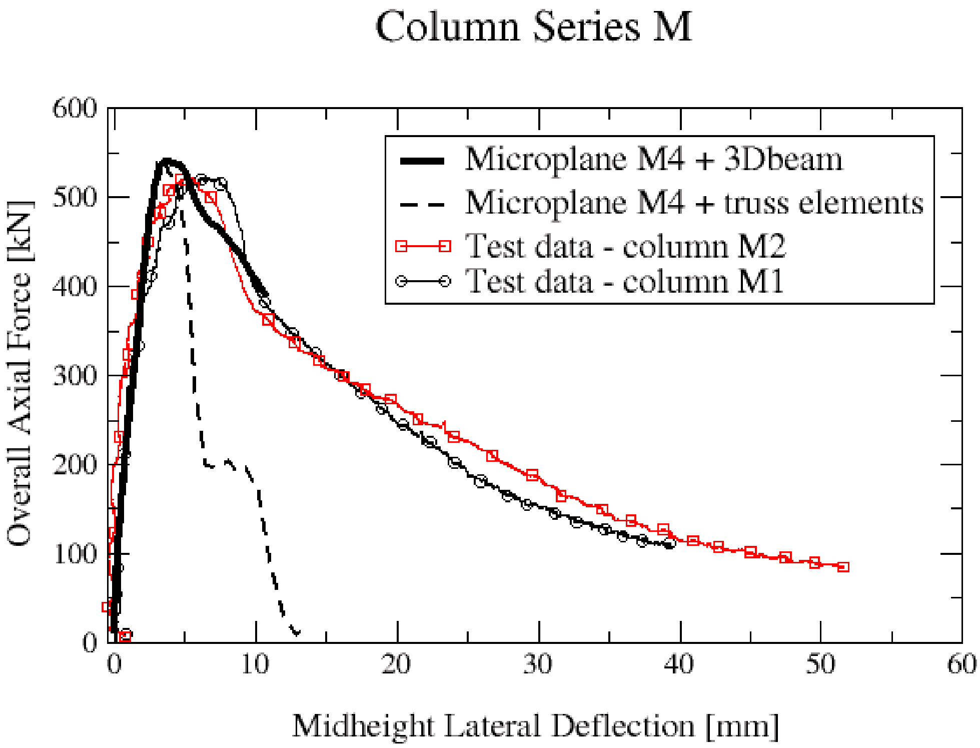

Figure 22: Comparison of experimental and simulation data.

method is used. The comparison of experimental data with simulation, presented in Fig. 22, reveals reasonable agreement in ultimate bearing capacity (peak value) as well as in the post-peak behaviour (the descending branch of the load-deflection diagram), when nonlinear 3D beams elements are used to model the reinforcement.

Acknowledgments

This work was partially supported by Ministry of Education of Czech Republic project No. MSM 210000003.

Some of the results published in this paper have been already presented also on the Fifth International Conference on Computational Structure Technology held in Leuven in 2000.

References

[1] A.C.T. Chen and W.F. Chen. Constitutive Relations for Concrete. *Journal of Engineering Mechanics*, (1975).

[2] E.S. Chen and O. Buyukozturk. Constitutive Model for Concrete in Cyclic Compression. *Journal of Engineering Mechanics*, (1985).

[3] F.B. Lin, Z.P. Bažant, J.C. Chern and A.H. Marchertas. Concrete Model with Normality and Sequential Identification. Computers & Structures, (1987).

[4] J. Mazars. *A Model of a Unilateral Elastic Damageable Material and its Application to Concrete*. Elsevier Science Publishers B.V., Amsterdam, (1986).

[5] Z.P. Bažant and S.S. Kim. Plastic-fracturing Theory for Concrete. *Journal of Structural Mechanics*, (1984).

[6] Z.P. Bažant, Y. Xiang and P.C. Prat. Microplane Model for Concrete I.: Stress-Strain Boundaries and Finite Strain. *Journal of Engineering Mechanics*, (1996).

[7] Z.P. Bažant, Y. Xiang, M.D. Adley, P.C. Prat and S.A. Akers. Microplane Model for Concrete II.: Data Delocalization and Verification. *Journal of Engineering Mechanics*, (1996).

[8] Z.P. Bažant and P.C. Prat. Microplane Model for Brittle Plastic Materials I: Theory, II: Verification. *Journal of Engineering Mechanics*, (1988).

[9] Ch. Farhat and F.X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. Journal for Num. Methods in Engng*, **32**, pp. 1205-1227, (1991).

[10] Y. Benveniste. A New Approach to the Application of Mori-Tanaka's Theory in Composite Materials. *Mechanics of Materials*, **6**, pp. 147-157, (1987).

[11] G.J. Dvorak and M. Šejnoha. Initial Failure Maps for Ceramic and Metal Matrix Composite Laminates. *Modelling Simul. Mater. Sci. Engng*, **4**, pp. 553-580, (1996).

[12] P. Řeřicha. Optimum Load Time History for Non-linear Analysis Using Dynamic Relaxation. *International Journal for Num. Methods in Engng*, **23**, pp. 2313-2324, (1986).

[13] M. Šejnoha. Laminated 4-node Plate Element Based on Refined Theory. *Acta Polytechnica*, **38 (1)**, pp. 63-74, (1998).

[14] B. Patzák. Object Oriented Finite Element Modeling. *Acta Polytechnica*, **39 (2)**, pp. 99-112, (1999).

[15] D. Rypl and Z. Bittnar. Mesh Generation Techniques for Sequential and Parallel Processing. *Aspects in Modern Computational Structural Analysis*, Balkema, Rotterdam, pp. 257-276, (1997).

[16] D. Rypl. Sequential and Parallel Generation of Unstructured 3D Meshes. *CTU Reports*, **2 (3)**, PhD. Thesis, (1998).

[17] B. Patzák, D. Rypl and Z. Bittnar. Nonlocal Parallel Explicit Dynamic Computations on SP2. *CTU Reports*, **3 (4)**, pp. 7-19, (1999).

[18] B. Patzák, D. Rypl and Z. Bittnar. Explicit Parallel Dynamics with Nonlocal Constitutive Models. *Developments in Analysis and Design Using Finite Element Methods*, Civil-Comp Press, pp. 77-83, (1999).

[19] I. Foster. *Designing and Building Parallel Programs.* Addison-Wesley, (1995).