Dynamic-explicit finite element simulation of complex problems in civil engineering by parallel computing

U. Gabbert, J. Grochla, H. Köppe

Institut für Mechanik, Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

Abstract
The paper deals with the simulation of the non-linear and time dependent behaviour of complex structures in engineering. Such simulations have to provide high accuracy in the prediction of deformations and stability, by taking into account the long term influences of the non-linear behaviour of the material as well as the large deformation and contact conditions. The limiting factors of the computer simulation are the computer run time and the memory requirement during solving large scale problems. To overcome these problems we use a dynamic-explicit time integration procedure for the solution of the semi-discrete equations of motion, which is very suited for parallel processing. In the paper at first we give a brief review of the theoretical background of the mechanical modelling and the dynamic-explicit technique for the solution of the semi-discrete equations of motion. Then the concept of parallel processing will be discussed. A test example concludes the paper.

1 Introduction

The growing demand on sophisticated simulations of complex structural behaviour can only be met by increasing computational resources. Finite element models with more than 1 million degrees of freedom are frequent nowadays in the aerospace and automotive industry as well as in civil engineering. We developed for instance the serial finite element package ANSALT II for the analysis of the coupled thermal-mechanical behaviour of deep underground structures in rock salt, which is capable to handle models with several 100.000 3D finite elements (see Gabbert et al. 1995). One limiting factor of large scale applications - the memory requirement - has been overcome by using an explicit time integration strategy in transient as well as in static problems (dynamic relaxation). In an explicit analysis neither element nor system matrices are built, and consequently, the memory requirements are insignificant. Unfortunately, explicit methods are only conditionally stable and so the time step size has to be smaller than a critical value, which is directly dependent on the largest frequency of the finite element discretization (smallest element). As a consequence, in large scale problems we get extremely short time steps ($< 10^{-4}$ ...10^{-8} s) with a negative effect on the computer run time. This drawback can be overcome by parallel computing, where we can use the advantage, that explicit methods are excellent suited for parallel processing.
In general parallel computing is the only way to overcome the limits of single computers by magnitudes. The research challenge is to reformulate the given problem, develope parallel algorithms and computational strategies in order to fully utilise the capabilities of parallel machines. However, it can be realized that parallel computing is still an exotic technique in engineering and only some specialists are engaged in applying the parallel computing technology. But in recent years an increasing number of papers deals with concurrent scientific computation (for an overview see Adeli et al. 1992, 1993, Mesirov, 1991, Power, 1995, Valero et al. 1992) and also with applications in finite element analysis (e.g. Farhat et al. 1987, 1994, Valero et al. 1992, Le Tallec 1994, Lämmer et al. 1994). In our experience in the development of finite element software (Baumgarten et al. 1991, Blanke et al. 1994, Gabbert et.al. 1995,

1996) in addition to the efficiency the portability of the software is a major concern in parallel processing. With increasing different computer architectures, it is essential to adopt a program without rewriting most of the code or altering even the architectural basis of the software every time, a new parallel processor enters the market. Only some hardware dependent basic routines should be replaced to run a parallel finite element system on a massively parallel system as well as a workstation cluster or on a single processor machine. In relation to this, in parallel finite element analysis a mesh partitioning (substructure) technique combined with the message passing programming model seems to be a good basis for the development of general purpose parallel finite element software. The substructure technique is very popular and standard in large finite element applications and can be considered as the most natural approach in parallel processing of the finite element method. Within our development of the parallel finite element code PARFEM (Blanke et al. 1994, Gabbert et al. 1996) we have demonstrated, that based on a mesh partitioning/substructure concept most of the routines as well as the data structure of a serial finite element code (we used COSAR, see Baumgarten et al. 1991) can be used. For an automatic mesh partitioning numerous methods and powerful algorithms exist (see e.g. the shareware libraries *Chaco*: Hendrickson et al. 1995 and *Metis*: Karypis et al. 1995). PARFEM has been implemented on a ParSytec *GCPowerPlus 96* parallel computer using the basic routines *recv* and *send* only (see Haase et al. 1994). However, for a general industrial application current parallel processors are simply too expensive, and consequently, a network of relatively inexpensive workstations is a preferable alternative. Using PVM-like communication tools message passing software can be quickly adapted to a cluster of workstation. Based on the experiences of our former developments a new parallel finite element system for large scale non-linear analysis is under progress. As justified above, in difference to PARFEM, where the linear equation system of the discrete model is established and solved by an iterative solution technique (pcg with different preconditioner), with respect to very large scale problems we use an explicit time integration scheme also in the static case.

2 Basic equations of non-linear structural mechanics

The Hamilton principle

$$\int_{t=t_1}^{t=t_2} L\,dt \quad \Rightarrow \quad \text{Min.} \tag{1}$$

leads after spatial FE discretization to the following semidiscrete form of the equations of motion

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{f}_{int} &= \boldsymbol{f}_{ext} \\ \dot{\boldsymbol{q}}(t = t_0) &= \dot{\boldsymbol{q}}_0 \\ \boldsymbol{q}(t = t_0) &= \boldsymbol{q}_0 \end{aligned} \tag{2}$$

where L is the Lagrangian function, $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$ are the vectors of generalized displacements, velocities and accelerations, $\boldsymbol{M}$ and $\boldsymbol{C}$ are the mass and damping matrices respectively, $\boldsymbol{f}_{int}$ is the vector of the internal resisting forces and $\boldsymbol{f}_{ext}$ is the vector of the external applied forces. The internal forces include the shares of material and geometric nonlinearities. Therefore the internal force vector has to be updated at each time step as well as each iteration step during the time integration of the equations of motion. At the current position the internal forces may be evaluated from

$$\boldsymbol{f}_{int} = \int_V \boldsymbol{D}^T \boldsymbol{\sigma}(t, \boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}, \boldsymbol{h}) dV \tag{3}$$

where $\boldsymbol{D}$ is the incremental strain-displacement matrix, $\boldsymbol{\sigma}$ is the vector of Cauchy stresses, which may depend on the time t, strain $\boldsymbol{\varepsilon}$, strain rate $\dot{\boldsymbol{\varepsilon}}$ and internal parameters $\boldsymbol{h}$. If linearity is assumed with respect to the displacements, the internal forces can be written in the form

$$\boldsymbol{f}_{int} = \boldsymbol{K}\boldsymbol{q}(t) \tag{4}$$

where $\boldsymbol{K}$ is the stiffness matrix of the system. The equations hold for the time t. To get a new equilibrium state at $t + \Delta t$ we use an updated Lagrangian formulation with second Piola-Kirchhoff stresses and Green-Lagrangian strains. Different types of constitutive relations can be used. In ANSALT we use e.g. an additive combination model for the simulation of rock materials.

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{el} + \dot{\boldsymbol{\varepsilon}}^{th} + \dot{\boldsymbol{\varepsilon}}^{cr} + \dot{\boldsymbol{\varepsilon}}^{vp} \tag{5}$$

This formulation consists of an elastic and a thermal part, creep deformations and a fracture describing model, where the fracture behaviour is represented by a viscoplastic model with an extended Drucker/Prager criterion using an associated flow rule (see Gabbert et al. 1995).

3 Explicit time integration

Contrary to implicit schemes the generation and factorisation of system matrices, which are very memory and time consuming, may be avoided by explicit schemes (lumped mass and damping matrices). Working with system vectors (instead of system matrices), which may be added up by the finite element contributions, for the computation of the state variables $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, it is possible to increase the number of degrees of freedom and thus large engineering problems can be treated. That is also a reason why in spite of the shortcomings, explicit algorithms are often preferred to the analysis of very complex structures (Underwood, Park 1982). From a software development point of view the application of the explicit time integration schemes provides the opportunity to create an uniform software concept both for the solution of static and dynamic problems. To this end a static problem has to be transformed into a dynamic one by adding an artificial acceleration and an artificial damping. This method is known as dynamic relaxation (Wood 1971 and Papadrakadis 1981, for a parallel version see Topping et al. 1994). If static problems are solved by dynamic relaxation both the mass and the damping matrices lose their physical background and become fictitious quantities which control the iteration process. The central difference method to approximate $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ has proved to be a very effective procedure to integrate the initial value problem (1).

$$\begin{aligned} \dot{\boldsymbol{q}} &= \frac{1}{\Delta t}\left(\boldsymbol{q}^{t+\frac{1}{2}\Delta t} - \boldsymbol{q}^{t-\frac{1}{2}\Delta t}\right) \\ \ddot{\boldsymbol{q}} &= \frac{1}{\Delta t}\left(\dot{\boldsymbol{q}}^{t+\frac{1}{2}\Delta t} - \dot{\boldsymbol{q}}^{t-\frac{1}{2}\Delta t}\right) \end{aligned} \tag{6}$$

The stability consideration of the central difference scheme gives a limitation of the time step length of

$$\Delta t_{krit} \le 2/\omega_{max} \tag{7}$$

where ω_{max} is the highest frequency of the FE model. If we insert the Eqs. (6) in Eq. (1), accept $\boldsymbol{C} = c\boldsymbol{M}$ with a lumped mass matrix $\boldsymbol{M}$, the explicit solution scheme of Eq. (1) may be written as

$$\dot{q}_i^{t+\frac{1}{2}\Delta t} = \frac{2 - c^t\,\Delta t}{2 + c^t\,\Delta t}\,\dot{q}_i^{t-\frac{1}{2}\Delta t} + \frac{2\,\Delta t}{m_{ii}\,(2 + c\,\Delta t)}\left(f_{ext_i}^t - f_{int_i}^t\right)$$
$$q_i^{t+\Delta t} = q_i^t + \Delta t\,\dot{q}_i^{t+\frac{1}{2}\Delta t} \tag{8}$$

In Eq. (8) $f_{int\,i}^t$ is the internal force related to the degree of freedom i, which is calculated by using Eq. (3), and $f_{ext\,i}^t$ is the given external force at the degree of freedom i. If we solve a static problem the stability criterion of the central difference method Eq. (7) is used to estimate the diagonal element m_{ii} of the lumped mass matrix. The largest eigenvalue of a matrix $\boldsymbol{A}$ is always smaller than any matrix norm (e.g. maximum sum of the absolute elements of a matrix row, also known as Gerschgorin´s theorem). From this follows the estimation (note that in our case $\boldsymbol{A} = \boldsymbol{M}^{-1}\,\boldsymbol{K}$):

$$\omega_i^2 \le \frac{1}{m_{ii}} \sum_{j=1}^{n} \left|k_{ij}\right| \tag{9}$$

Substitution of Eq. (7) in Eq. (9) gives an estimation of m_{ii}:

$$m_{ii} \ge \frac{1}{4}\,\Delta t^2 \sum_{j=1}^{n} \left|k_{ij}\right| \tag{10}$$

The damping coefficient c included in (8) may be calculated from the condition of an aperiodic oscillation, i.e.

$$\vartheta = 1 = \frac{c}{2\omega_0} \tag{11}$$

where ϑ, c, ω_0 are the decrement of the damping, the damping coefficient and the smallest eigenfrequency, respectively. The smallest eigenfrequency ω_0 is approximated by the Rayleigh´s quotient of the FE system, i.e.

$$\omega_0^2 \le \frac{\boldsymbol{q}^{t\,T}\,\boldsymbol{K}\,\boldsymbol{q}^t}{\boldsymbol{q}^{t\,T}\,\boldsymbol{M}\,\boldsymbol{q}^t} \tag{12}$$

The substitution of Eq. (12) in Eq. (13) gives the damping factor as

$$c^t = 2\sqrt{\frac{\boldsymbol{q}^{t\,T}\,\boldsymbol{K}\,\boldsymbol{q}^t}{\boldsymbol{q}^{t\,T}\,\boldsymbol{M}\,\boldsymbol{q}^t}} \tag{13}$$

It should be noted, that the quadratic forms in Eqs. (12) and (13) are simply calculated at the element level by adding up the shares of each element.

4 Concept of a parallel explicit time integration and dynamic relaxation

Fig. 1 shows the structure of the time stepping algorithm by using an explicit integration in dynamics or the dynamic relaxation in static cases. Each row represents the equation for one degree of freedom of the finite element system. Obviously, this form is excellent suited for parallel computing.

The communication via a network is the main disadvantage of the virtual parallel computer. Because the computers of a cluster can not use a shared memory like vector computers or high speed links like transputers all data have to pass through the network. Although the per-

formance of the data transfer by networks is speeded up in the last years (100 Mbits/s and more by CDDI and FDDI networks), the transfer of data is slow compared to massively parallel computers. The absolute minimisation of the data transfer is the consequence of this fact. In our concept of a parallel explicit FE system based on the master-slave model (Fig. 2) the master generates the finite element model from the input data, configures the virtual parallel computer (find out the number of computers in the cluster and bind them to the task) and generates the mesh partitions of the FE model of the solution domain. The partitioning is done by using standard techniques (see e.g. Hendrickson 1995, Karypis 1995). We tested different methods at large 3D engineering models and got best results with the multi-level techniques of the Metis library (see Deutsch 1995). The master maps the mesh partitions as well as all mesh data (geometry, material, forces, boundary conditions etc.) onto the slaves. The slaves get the data from the master and create their own data arrays for the analysis, which is done only once.

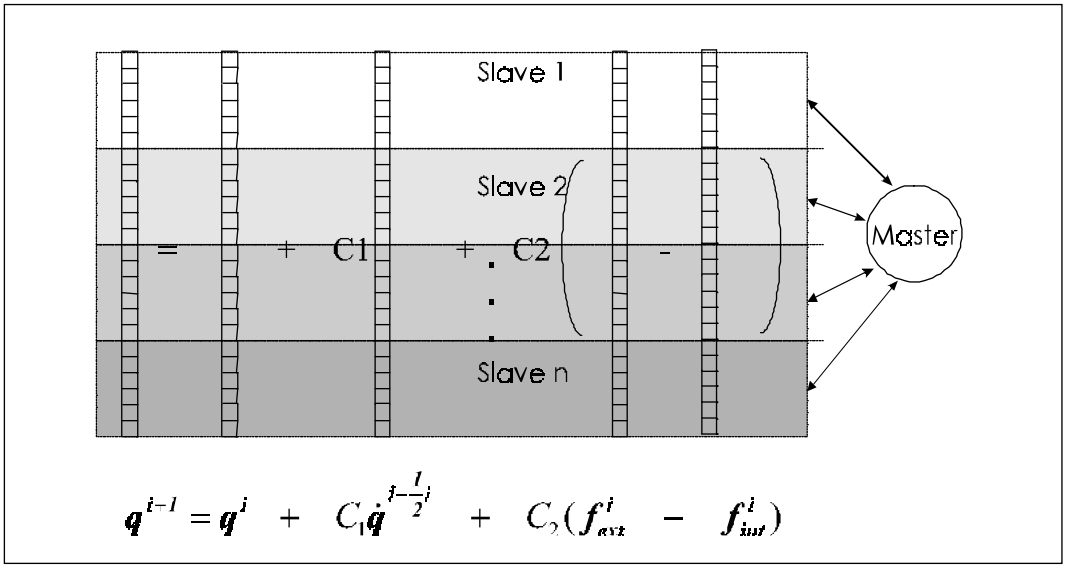

Figure 1 Structure of the explicit time integration (e.g. dynamic relaxation) scheme

The advantage of the explicit method and the dynamic relaxation is that no system matrices have to be assembled. The algorithms can be implemented on element basis only. Consequently, each slave is able to calculate the vector of displacements for his own mesh partition. A fraction of it, the external degrees of freedom, which share common boundaries with other domains, are sent for an update to the master. Here also the global control parameters for the solution scheme (Eqs. 10 and 13) are calculated. These parameters and the globally updated fractions of the solution vector (external degrees of freedom) are sent back to the slaves to start the next time or iteration step. Local results like element stresses and strains are also calculated separately on each processor without any communication. At the end all required results are sent to the master. Using these results of the complete FE model the master can determine other solution components of interest. The implementation of the concept briefly described above on an heterogeneous computer cluster composed of 4 HP-workstations C120, one 2-processor workstation J210 and one Pentium PC 166 under Windows-NT (see[14]) is under progress. As programming language we use FORTRAN 77 and PVM (Geist 1994) for the realisation of the virtual parallel computer, the de facto standard for heterogeneous network computing. This makes the concept open for the integration of other computer types or for an implementation on other designed computer clusters.

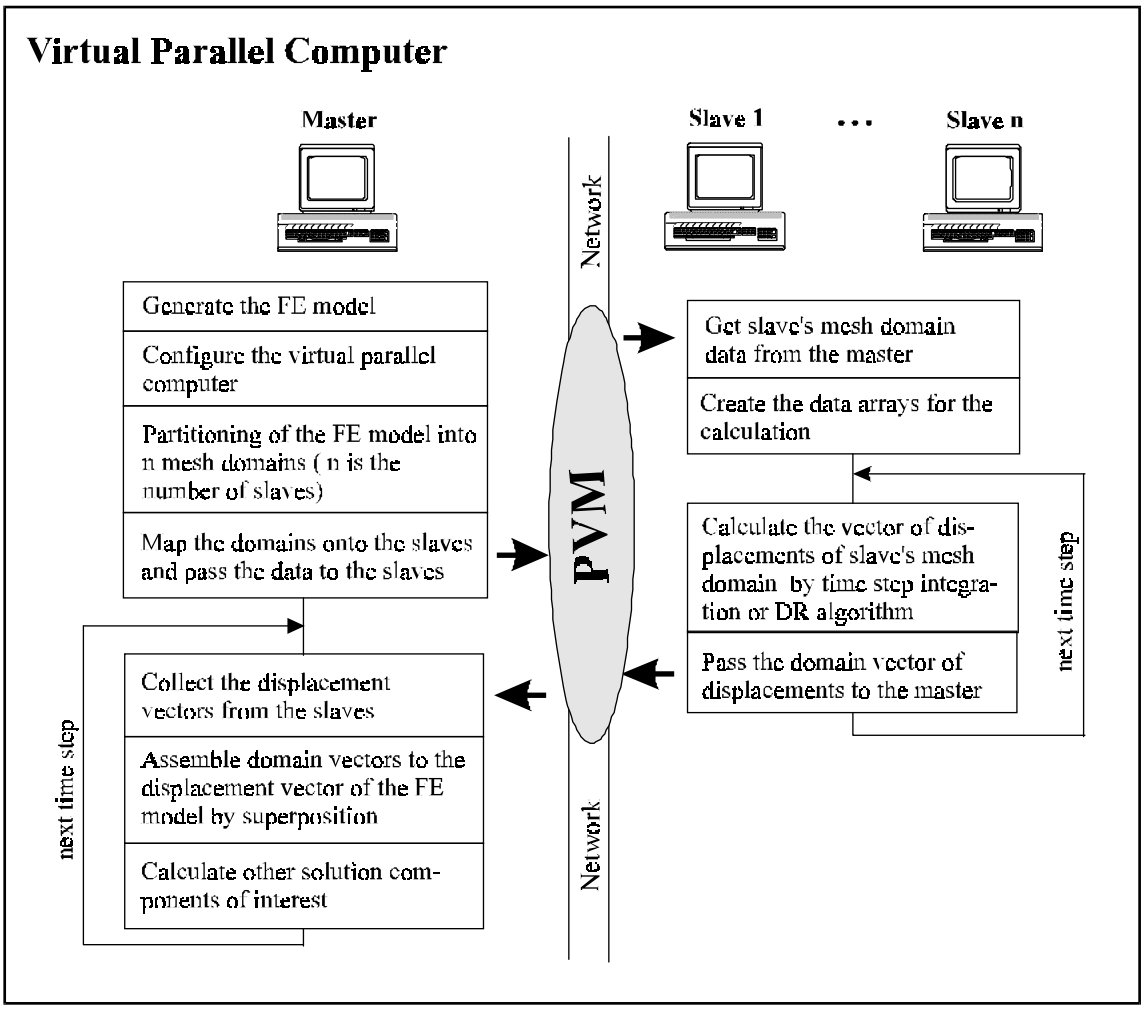

Figure 2 Virtual parallel computer concept of an explicit finite element system

5 Example

The explicit time integration for dynamic problems as well as the dynamic relaxation method has been tested at several large scale engineering problems (e.g. transient process of mining out a cavity in a deep level rock salt formation, see Gabbert et al. 1995). Also a parallel version of the dynamic relaxation method has been developed and tested on a ParSytec *GCPowerPlus 96* parallel computer. Due to the limited length of the paper we only present as a simple test examples a problem from linear crack mechanics (Fig. 3), which has been calculated with different degrees of freedom and 2, 4, 8, 16, 32 and 64 parallel processors. As a result in Fig. 3 the speed up of the total elapsed time with respect to the number of processors and the increase of the communication time is demonstrated.

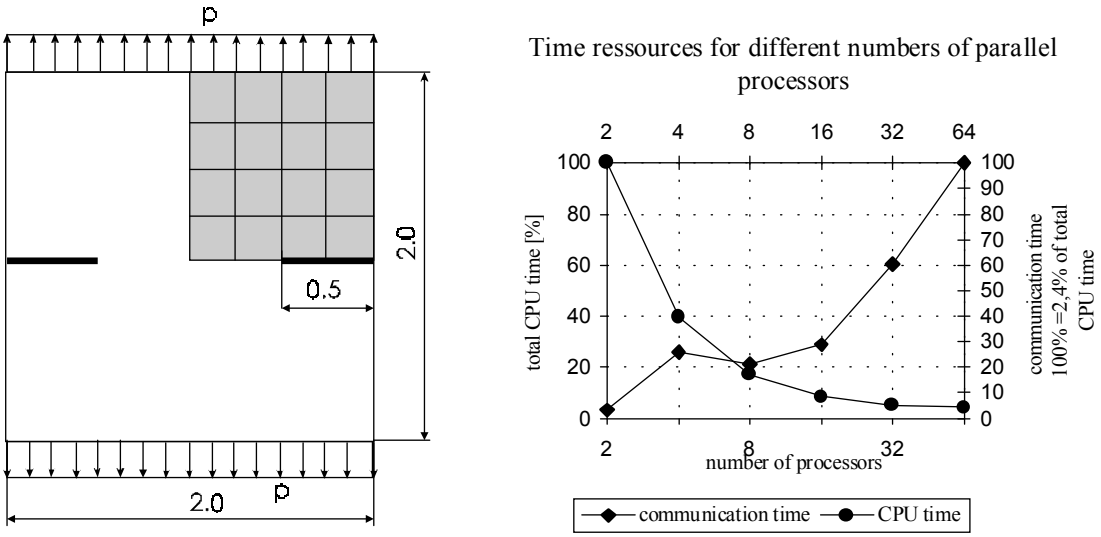

Figure 3 Parallel finite element calculation of a plane crack problem
Parameters: E = 1 N/mm^2, ν = 0.3, p = 1 N/mm
FE mesh: 16x32 8-node plane elements (3266 degrees of freedom)

6 Conclusion

The exploitation of parallelism within algorithms and the efficient use of hardware calls for new strategies in parallel finite element software. The combination of the domain decomposition (substructure technique respectively) with the explicit time integration in transient problems and dynamic relaxation in static cases discussed in this paper gives a general and uniform concept for the simulation of large scale non-linear problems in mechanics, which is very well suited for parallel processing. In the paper at first we give a brief review of the theoretical background of the mechanical modelling and the dynamic-explicit technique for the solution of the semi-discrete equations of motion. For the implementation, which is under progress, we use a heterogeneous computer cluster. The programming language is FORTRAN 77 and as software for the realisation of the virtual parallel computer we use PVM. A test example concludes the paper. However, there is remains a lot to do to make it a tool for real complex engineering problems.

Acknowledgement: This work has been partially supported by the Deutsche Forschungs Gemeinschaft (DFG) under the project number GA 480/2-1. The support is gratefully acknowledged.

References

1. Adeli,H. (Ed.): Supercomputing in Engineering Analysis. Marcel Dekker, Inc. New York, Basel, Hong Kong 1992
2. Adeli,H., Kamal,O.: Parallel Processing in Structural Engineering. Elsevier Applied Science, London, New York 1993

3. Baumgarten,H., Gabbert,U., Zehn,M., et al.: COSAR - Universelles FEM-System: Nutzerhandbuch. Forschungsgesellschaft für Technische Mechanik FEMCOS mbH Magdeburg, 1991
4. Blanke,G., Gabbert,U.: Ein Konzept zur Parallelisierung universeller FEM-Systeme. Berichte 39. Int. Wiss. Kolloquium, 27.-30. September 1994, TU Ilmenau, Bd1, S. 495-498
5. Boersma,A.: Lösung von linearen und nichtlinearen Problemen in der FEM. Berichte des Workshops Parallelisierung, 27.-18. Januar 1993, TU Darmstadt, S.159-166
6. Deutsch,G.: Bwertung und Weiterentwicklung intelligenter Netzpartitionierungsalgorithmen für parallele FEM-Systeme. Studienarbeit, Univ. Magdeburg, Institut für Mechanik, 1995
7. Dongarra, J., Wasniewski, J.: Parallel Scientific Computing. First International Workshop, PARA´94, Lyngby, Denmark, June 1994, Proceedings, Springer Verlag 1994
8. Farhat,C., Wilson,E.: A new finite element concurrent computer program architecture. Int. J. for Num. Meth. in Engn., Vol. 24, 1987, pp. 1771-1792
9. Farhat,C., Roux,F.-X.: Implicit parallel processing in structural mechanics. Computational Mechanics Advances, Vol 2, No. 1, 1994, pp. 1-124
10. Gabbert,U., Zehn,M., Blanke,G.: Intelligent mesh generation and domain decomposition for a parallel general purpose finite element system. H.Power (Ed.): *Applications of High-Performance Computing in Engineeing IV*, Computational Mechanics Publications, Southampton, Boston 1995, pp. 123-130
11. Gabbert,U., Honecker,A., Köppe,H., Nipp,H.: Numerical Analysis of Large Geomechanical Systems in Civil Engineering by the Dynamic Relaxation Technique. P.J.Pahl, H.Werner (Eds.): *Computing in Civil and Building Engineering*. A.A. A.A.Balkema Publishers Rotterdam, Netherlands 1995, pp. 1275-1281
12. Gabbert,U., Blanke, G.: Das parallele FEM-System PARFEM - Konzeption und Realisierung, Pre-Print IFME 96/3, July1996, Institut für Mechanik, Universität Magdeburg, 1996
13. Geist, A.: PVM - Parallel Virtual Machine. A User´s Guide and Tutorial for Networked Parallel Computing. MIT Press 1994
14. GMD - Forschungszentrum Informationstechnik GmbH: HPCN stärkt die europäische Industrie. Workshop zur Förderung des parallelen Rechnens, Mai 1996, Schloß Birlinghoven
15. Hendrickson,B., Leland,R.: The Chaco user´s guide - Version 2.0. Sandia National Laboratories, Albuquerque, NM 87185-1110, 1995
16. Haase,G., Hommel,T., Meyer,A., Pester,M.: Bibliotheken zur Entwicklung paralleler Algorithmen. Preprint SPC95_20, TU Chemnitz-Zwickau, Fakultät für Mathematik 1994
17. Karypis,G., Kumar,V.: Metis Unstructured graph partitioning and sparse matrix ordering system. Version 1.0, 1995, Univ. of Minnesota 1995
18. Lämmer,L., Meißner,U., Mücke,R., Olden,J.: Adaptive finite element analysis of plate problems for parallel computers. Topping,B.H.V., Papadrakakis,M. (Eds.): *Advances in Parallel and Vector Processing for Structural Mechanics*, CIVIL-COMP Ltd., Edingburgh, Scotland, 1994, pp. 185-190
19. Le Tallec,P.: Domain decomposition methods in computational mechanics. Computational Mechanics Advances, Vol 1, No. 2, 1994, pp. 121-220
20. Malone,J.G.: Automated mesh decomposition and concurrent finite element analysis for hypercube multiprocessor computers. Comp. Meth. in Appl. Mech. Eng. 70, 1988, pp.27-58
21. Meißner,U., Gabbert,U.: Automatisierte Steuerung der Modell- und Ergebnisqualität für Finite-Elementsysteme. Preprint IFME 96/1, Januar 1996, Universität Magdeburg, Institut für Mechanik 1996
22. Mesirov,J.P.:Very Large Scale Computation in the 21st Century. Society for Industrial and Applied Mathematics, Philadelphia, 1991
23. Papadrakakis,M.: A method for automated evaluation of the dynamic relaxation parameters. Computer Methods in Applied Mechanics and Engineering, 1981, Vol. 25, pp. 35-48
24. Power,H. (Ed.): Applications of High-Performance Computing in Engineeing IV, Computational Mechanics Publications, Southampton, Boston 199
25. Topping,B.H.V., Khan,A.I.: Parallel computation schemes for dynamic relaxation. Engineering Computations, Vol. 11, 1994, pp. 513-548
26. Underwood,P.G.,Park, K.C.: A stand-alone explicit time integration package for structural dynamic analysis, Int. J. for Num. Meth. in Engineering,1982, Vol. 18, pp. 609-622
27. Valero,M., Onate,E., Jana,M., Larriba,J.L., Suarez,B.: Parallel Computing and Transputer Applications. Part I and II, CIMNE International Center for Numerical Methods in Engineering, Barcelona, IOS Press, Amsterdam, Washington 1992
28. Wood,W.L.: Note on dynamic relaxation. Int. J. for Num. Meth. in Eng., 1971,Vol. 3, pp. 145-147