A client-server concept for integration of the p-version finite element analysis with geometric modelling

E. Rank, H. Bröker, A. Düster, M. Rücker

Februar 2000

submitted to:
ICCCBE-VIII, Stanford 2000

A client-server concept for integration of the p-version finite element analysis with geometric modelling

Ernst Rank, Henrike Bröker, Alexander Düster, Martin Rücker [1]

Abstract

We present an implementation of a three-dimensional p-version for structural problems of solids with almost arbitrarily curved surfaces. The blending function technique for geometric mapping allows to design a client server software structure, completely seperating the geometric model from the finite element analysis part of the program. Numerical examples show the efficiency of this approach.

Introduction

While in the standard h-version of the finite element method the mesh is refined to achieve convergence, the polynomial degree of the shape functions remains unchanged. Usually a low order approximation of degree $p = 1$ or $p = 2$ is chosen. This principle enforces a loose coupling of a geometric and the corresponding finite element model, as for each refined approximation, a new finite element mesh has to be generated starting from the geometric model. The p-version of the finite element method, on the other hand, leaves the mesh unchanged and only increases the polynomial degree of the shape functions locally or globally. In our implementation, a hierarchical set of shape functions is applied, providing a simple and consistent possibility of implementation in 1-, 2- or 3-dimensional analysis. Guidelines to construct these meshes a priori can often be given much easier for the p-version than for the h-version. For linear elliptic problems it was also proven, that a sequence of meshes can be constructed [10] so that the approximation error only depends on the polynomial degree p and not on the order

[1] All authors: Lehrstuhl für Bauinformatik, Technische Universität München, Arcisstr. 21, D-80290 München, Germany

of singularities in the exact solution, yielding a very high accuracy of the computation, which could not be achieved by the classical h-version in many cases of practical importance [1].

Our p-version implementation [3] uses a set of hierarchical basis functions, which can easily be implemented up to any desired polynomial degree [9]. These shape functions can be grouped into four classes: nodal, edge, internal and, for three-dimensional hexahedral elements, face modes. Whereas the nodal modes are identical to the well-known classical bi- or tri-linear shape functions, the other classes offer the key for an efficient parallelisation of the p-version as well as for a close coupling to a geometric model in a distributed software system [7, 8].

The blending function method

An important difference between h- and p-version finite element methods lies in *mapping requirements.* Because in the p-version the element size is not reduced as the polynomial degree is increased, the description of the geometry has to be independent of the number of elements. This results in the necessity to construct elements with an exact representation of the boundary. The isoparametric mapping, used in standard finite element formulations, can be seen as a special case of mapping using the *blending function method* [4, 5, 9]. Our formulation of this method is based on the work of G. KIRÁLYFALVI and B. SZABÓ [5]. Following these ideas, element boundaries can be implemented as (almost) arbitrarily curved edges and faces.

Before describing the software structure taking advantage of the blending function method, some basic concepts of element matrix computation should be reviewed in the light of this mapping technique. The definition of an element stiffness matrix

$$\mathbf{K}^e = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{M}(\mathbf{N}, \mathbf{C}, \mathbf{Q}, \mathbf{J}^{-1}, \mid \mathbf{J} \mid)\, d\xi d\eta d\zeta, \tag{1}$$

is given by an integral of a matrix function $\mathbf{M}$, which depends on the shape functions $\mathbf{N}$, the material properties $\mathbf{C}$, the mapping function $\mathbf{Q}$ and the Jacobian matrix $\mathbf{J}$, with its inverse and determinant $\mathbf{J}^{-1}, \mid \mathbf{J} \mid$.

In general, this integration has to be performed numerically in the local coordinate system (ξ, η, ζ) of a standard hexahedral element domain $\Omega_{\mathrm{st}}^{\mathrm{h}} = [(-1,1) \times (-1,1) \times (-1,1)]$. Therefore, the terms $\mathbf{N}, \mathbf{C}, \mathbf{Q}, \mathbf{J}^{-1}, \mid \mathbf{J} \mid$ have to be computed only at the integration points (ξ^l, η^m, ζ^n).

Consider now a hexahedral element, as pictured in Figure 1. $\mathbf{X}_i = (X_i, Y_i, Z_i)$, $i = 1, ..., 8$ denote the global coordinates of the nodes. $\mathbf{E}_i = (E_{ix}, E_{iy}, E_{iz})$, $i = 1, ..., 12$ are functions which depend on local coordinates

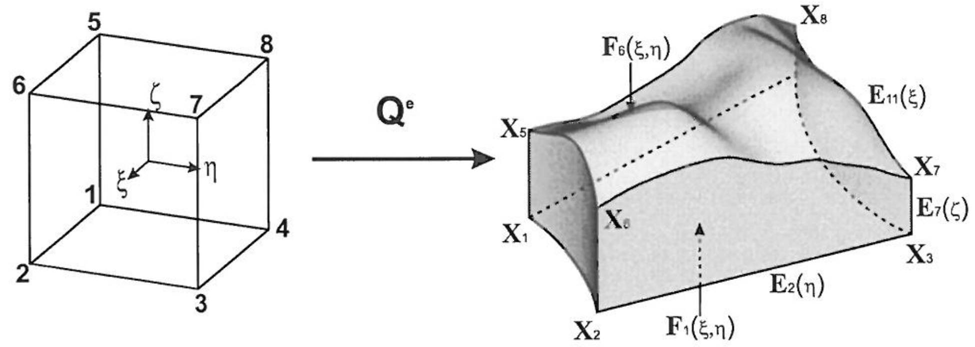


Figure 1: Mapping by the blending function method

(ξ, η, ζ) and describe the shape of each edge. $\mathbf{F}_i = (F_{ix}, F_{iy}, F_{iz})$, $i = 1, ..., 6$ denote functions describing the shape of each face. The mapping function $\mathbf{Q}^e(\xi, \eta, \zeta)$ from local coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T$ to global coordinates $\mathbf{x} = (x, y, z)^T$ is obtained by

$$\mathbf{x} = \mathbf{Q}^e(\xi, \eta, \zeta) = \sum_{i=1}^{8} N_i(\xi, \eta, \zeta)\, \mathbf{X}_i + \sum_{i=1}^{6} \mathbf{f}_i(\xi,\, \eta,\, \zeta) - \sum_{i=1}^{12} \mathbf{e}_i(\xi,\, \eta,\, \zeta)\,. \qquad (2)$$

The first term is the standard mapping of isoparametric eight-noded hexahedral elements using tri-linear shape functions N_i. The second term is referred to as face blending and is a linear function of the face mappings $\mathbf{F}_i$, whereas the third term corresponds to the edge blending being a linear function of the edge mappings $\mathbf{E}_i$ (see [3]).

It can now readily be seen, that the Jacobian matrix $\mathbf{J}$ containing the derivatives of the mapping function $\mathbf{Q}^e$ with respect to the local coordinates ξ, η and ζ depends on the following three groups of geometric information:

- the coordinates of the nodes $\mathbf{X}_i$ for $i = 1, ..., 8$
- the (twelve) tangent vectors $\dfrac{\partial\,\mathbf{E}_i}{\partial\, r}$
- the six tangential planes $\left(\dfrac{\partial\,\mathbf{F}_i}{\partial\, r}, \dfrac{\partial\,\mathbf{F}_i}{\partial\, s}\right)$, r and s denoting generic local coordinates.

Therefore the Jacobian matrix at any interior point $\mathbf{x}(\xi, \eta, \zeta)$ of the element can be computed only from nodal, edge and surface data which have to be evaluated at points on the surfaces and edges corresponding in their local coordinates to those of $\mathbf{x}$. Figure 2 shows schematically the essential information on the boundary and surface of a hexahedral element for computing **J**. All these data are typically available in geometric modellers supplying a boundary representation model [2] for the description of solids.

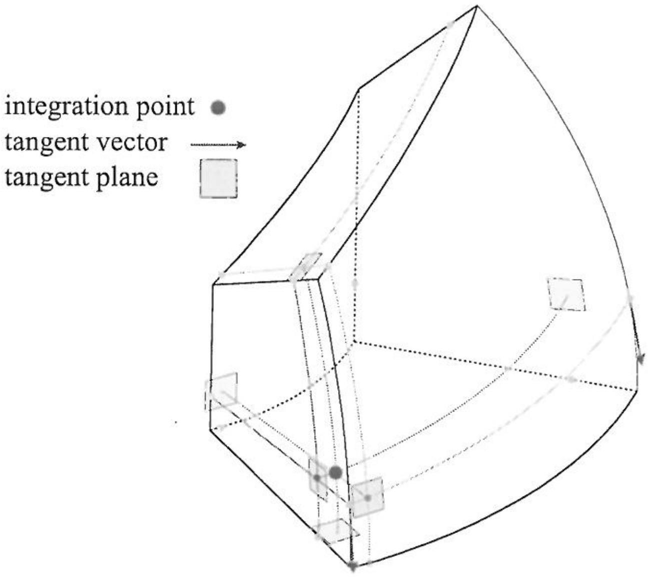


Figure 2: Geometric information of the edges and faces of an integration point

Summarizing, this blending function technique for mapping the geometry of p-elements offers the possibility to completely separate all geometric computations involved in a finite element analysis from the non-geometric part. Furthermore, it is possible to design a distributed software system, where the geometric model of a CAD-program, although running in a different process or even on a different computer, is directly linked to a finite element kernel. This software structure offers the advantage of using all state-of-the-art CAD-techniques like geometric editing or parametric design in a finite element analysis, immediately. The increase of efficiency for practical work may be dramatic, as such a system for computer integrated engineering relieves from the necessity to transfer geometric data from CAD to FEA, which is usually very time-consuming, even if only some geometric parameters of the model change.

Our implementation is based on a client-server architecture, as shown in Figure 3. The finite element code is the client, requesting all information on the boundary or surface of an element from a geometry server. We are using AutoCAD with its ACIS-kernel and the ARX-interface to provide the geometry information. Assuming that the structural model already exists in the CAD-environment, the user can define the boundary conditions for the finite element computation on the geometric model. After having started the finite element computation in a different process the relevant data will be received from the CAD-system. Data of interest are topological information of the finite element mesh, as well as Neumann and Dirichlet boundary conditions and edge and surface data to compute the Jacobian matrix **J** as described above.

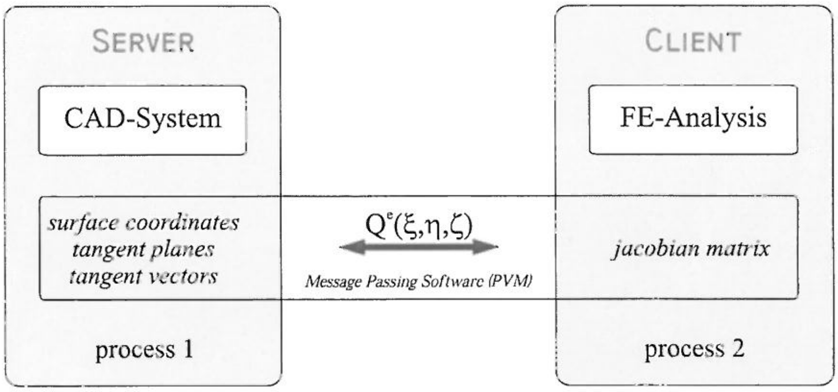
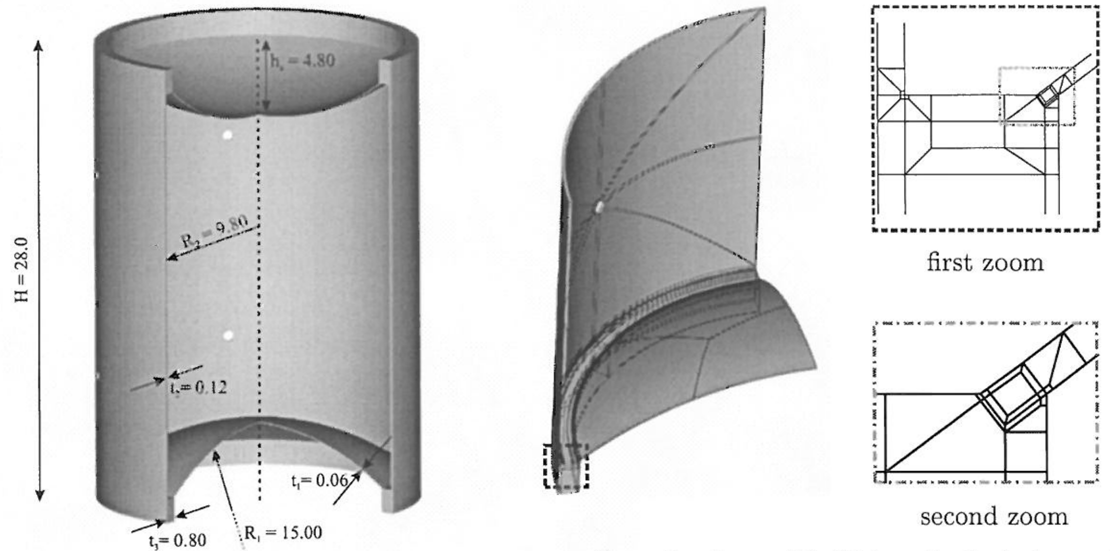


Figure 3: Client-Server structure of our implementation

The major advantage of this software structure is the *direct* coupling of FEA and geometric model. The usual *loose* coupling of these models consists of a transfer of the boundary or surface *description* by some parameters, requiring that the geometric type of the boundary is not only known in the geometric modeller but also in the FEA-program. Therefore, the surface description of the boundary has either to be coded in the finite element program in the same way as in the geometric modeller, or the boundary must be approximated by some other type of surface [5]. In contrast to these approaches, our structure does not need any knowledge on the type of surface, it obtains all necessary information directly by requesting the mapping data at discrete points. Thus, any change in geometry of a structure in the geometric model is directly inherited to the finite element model. The client-server structure is implemented using PVM [6] as message passing software between the CAD-program and the FEA-code.

A numerical example

As a complex three-dimensional construction we consider a reactor pressure vessel, loaded by interior pressure (see Figure 4). It is composed of a spherical and a cylindrical shell-like structure which are merged in a cylindrical solid. The radius of the spherical shell is $R_1 = 15$ and the thickness equals $t_1 = 0.06$, while the thickness ratio of the cylindrical part is $t_2/R_2 = 0.12/9.80$. A mesh consisting of 85 hexahedral elements, taking advantage of symmetry is chosen to discretize the structure. The classical approach to this problem would demand for special elements in order to model the transition from shell- to solid elements (see right part of Figure 4). Due to the use of three-dimensional continuum p-version elements the whole structure can be modelled with the same type of discretization and no transition elements are needed.



structure loaded by interior pressure

discretization with 85 hexahedral elements, zoomed part of a vertical cut through the structure with corner refinement

Figure 4: System and mesh of a complex shell model

A series of computations is performed, based on an isotropic trunk space $\mathcal{S}_{\text{ts}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ with polynomial degrees $p = p_\xi = p_\eta = p_\zeta = 3, ..., 8$ (for a detailed description of the ansatz space see either [3] or [9]. The relative error in energy norm is plotted in Figure 5.

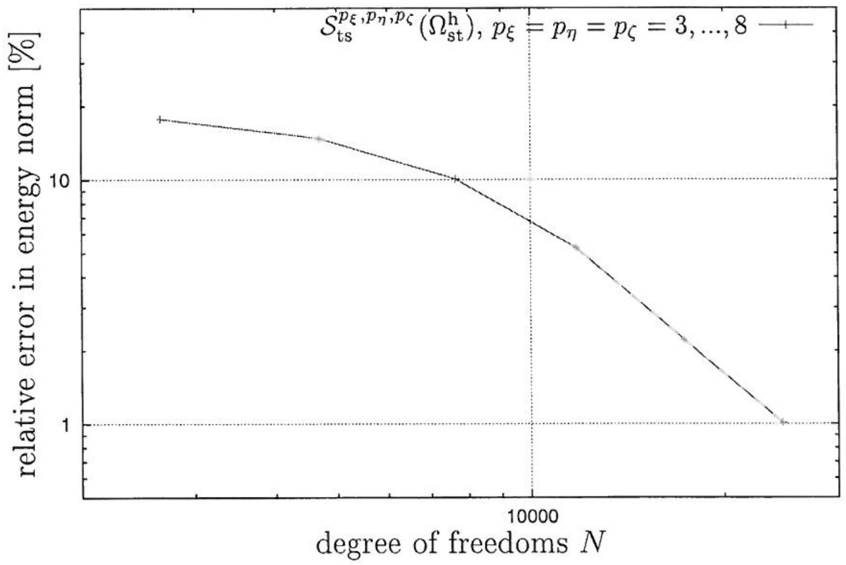


Figure 5: Relative error in energy norm of a complex shell model

Using a polynomial degree of $p = 8$ with a corresponding number of 24610 degrees of freedom an accuracy with approximately 1.0% error in energy norm is achieved. This error is estimated from an extrapolation of the strain energies obtained with polynomial degrees $p = 6, 7, 8$ (see [9]).

The deformed structure (scaling factor=1500) and the von Mises stress are sketched in Figure 6, where the results are evaluated on a fine post-processing mesh, being obtained by a subdivision of each p-element.

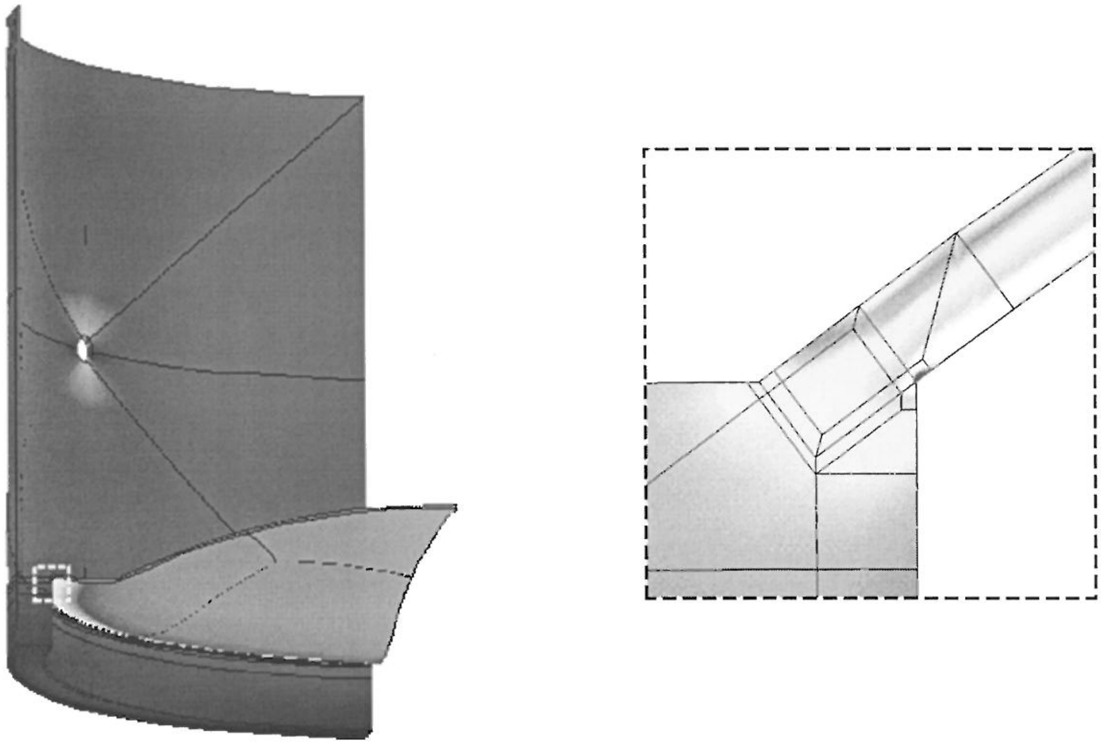


Figure 6: Von Mises stress on the deformed structure (scaling factor 1500) of a complex shell model

References

[1] I. Babuška, B. Szabó. On the rates of convergence of the finite element method. *Int. J. Numer. Methods Engrg.*, **18**, 323-341, 1982.

[2] J. Corney. *3D modeling with the ACIS kernel and toolkit.* John Wiley and Sons Ltd, Chister, 1997.

[3] A. Düster, H. Bröker, E. Rank. The p-version of the finite element method for three-dimensional curved thin walled structures. *Technical Report, Lehrstuhl für Bauinformatik, TU München,* Germany, February 2000.

[4] W. J. Gordon, Ch. A. Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *Int. J. Numer. Methods Engrg.*, Vol. 7, 461-477, 1973.

[5] G. Királyfalvi, B. Szabó. Quasi-regional mapping for the p-version of the finite element method. *Finite elements in analysis and design*, **27**, 85-97, 1997.

[6] http://www.epm.ornl.gov/pvm.

[7] E. Rank, A. Düster, M. Krafczyk, M. Rücker. Some aspects of coupling structural models and p-version finite element models. *Proceedings of the 4th World Congress on Computational Mechanics*, Buenos Aires, Argentina, 1998.

[8] E. Rank, M. Rücker, A. Düster, H. Bröker. The efficiency of the p-version finite element method. in Proceedings of *ECCM '99, European Conference on Computational Mechanics,* München, Germany, CD-ROM, 1999.

[9] B. Szabó, I. Babuška. *Finite element analysis.* John Wiley and Sons, Inc., New York, 1991.

[10] B. Szabó. Mesh design for the p-version of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, **55**, 181-197, 1986.