HIGH ORDER SOLID ELEMENTS FOR THIN-WALLED STRUCTURES: NO TRICKS? - NO CRIMES!

E. Rank, H. Bröker, A. Düster, V. Nübel

1/2001

in:
TRENDS IN COMPUTATIONAL STRUCTURAL MECHANICS, CIMNE, Barcelona, Spain, 2001.
HIGH ORDER SOLID ELEMENTS FOR THIN-WALLED STRUCTURES: NO TRICKS? - NO CRIMES!

Ernst Rank*
Lehrstuhl für Bauinformatik
Technische Universität München
Arcisstr. 21
D-80290 Munich, Germany
e-mail: rank@bv.tum.de
web page: http://www.inf.bv.tum.de

Henrike Bröker, Alexander Düster,
Vera Nübel
Lehrstuhl für Bauinformatik
Technische Universität München
Arcisstr. 21
D-80290 Munich, Germany
e-mail: broeker,duester,nuebel@bv.tum.de
web page: http://www.inf.bv.tum.de

Abstract. A p-version finite element approach for a strictly three-dimensional analysis of thin-walled structures will be discussed in this paper. As key issues, anisotropic high order ansatz spaces, a consistent coupling to a geometric model and an efficient integration scheme for high order elements are identified. Results of this approach are shown in several numerical examples.

Key words: Thin-walled structures, finite element analysis, p-version, vector integration, solid elements.

1 INTRODUCTION

The development of accurate and efficient element formulations for thin-walled structures has been in the focus of research in computational mechanics since the event of the finite element method. It seemed to be clear very early, that an investigation of plate or shell problems with tetrahedral or hexahedral elements is not feasible for practical problems, as a sufficient accuracy could only be obtained by a prohibitively large amount of degrees of freedom and computational effort. The major reasons for this observation are the mapping requirements of isoparametric, low order elements. Accurate solutions can only be obtained, if the aspect ratio of an element is close to one, resulting in an enormous amount of elements, even if only a few layers are used over the thickness of the structure. A very natural consequence of this observation was to use dimensionally reduced models like Reissner-Mindlin plates or Naghdi shells, and
to build element formulations on top of these theories. Yet, it turned out that naive displacement type elements for these models lead to notorious numerical problems like locking or spurious energy modes, giving rise to the development of numerous improvements, like mixed elements or enhanced strain formulations (e.g. [3]). Although many of these 'tricks' have shown to be very successful in practice and are now well understood from a mathematical point of view, they are often connected with 'crimes' being sometimes of pure mathematical nature, but often also due to a violation of the underlying model assumptions, like the kinematics of the displacement field over the thickness of the structure. To overcome these problems, shell elements with higher order kinematics giving a better approximation to the three-dimensional stress state have been presented [2]. As an alternative, we will investigate in this paper the feasibility of strictly three-dimensional models, using high order elements being coupled to a precise geometric description of the structure. As a key issue the p-version of the FEA is used, offering a consistent and accurate way to implement solid elements having very large aspect ratio (up to a few hundred) and to represent much more general shapes of element surfaces than those available in the usual isoparametric approach. A transition from thin- to thick-walled constructions is thus possible without the necessity to couple models of differing dimensions and without imposing any restrictions on the (three-dimensional) kinematics of the structure. We will focus our discussion especially on the question of an efficient implementation of these elements, i.e. on an advantageous choice of anisotropic higher order ansatz spaces, and on an adaptive integration technique for higher order elements. Also very important is an efficient coupling of the finite element analysis to a geometric model, which has been described elsewhere [6]. Using these 'tricks' of the p-version, which are often not known or not necessary for low order elements, one obtains an efficient code being consistent by construction to the three-dimensional theory of elasticity.

1.1 The one-dimensional hierarchic basis

Our p-version implementation is based on the one-dimensional hierarchic basis, proposed by Szabó and Babuška [8]. Using this basis, ansatz functions can be implemented up to any desired polynomial degree. In Table 1 the one-dimensional basis is plotted for \( p = 1, 2, 3 \). An important property of the hierarchic basis functions is, that all lower order shape functions are kept in the higher order basis.

\[
\begin{align*}
  p = 1 & \quad \begin{array}{c}
  \end{array} \\
  p = 2 & \quad \begin{array}{c}
  \end{array} \\
  p = 3 & \quad \begin{array}{c}
  \end{array}
\end{align*}
\]

Table 1: Set of one-dimensional hierarchic shape functions for \( p = 1, 2, 3 \)
The two- and three-dimensional $p$-version implementation in our research code is based on the set of one-dimensional hierarchic shape functions

\begin{align}
N_1(\xi) &= \frac{1}{2}(1 - \xi) \\
N_2(\xi) &= \frac{1}{2}(1 + \xi) \\
N_i(\xi) &= \phi_{i-1}(\xi), \quad i = 3, 4, \ldots, p + 1
\end{align}

where the linear functions $N_1(\xi), N_2(\xi)$ are the so-called nodal modes. The higher order modes

\begin{equation}
\phi_j(\xi) = \sqrt{\frac{2j - 1}{2}} \int_{-1}^{\xi} P_{j-1}(t) \, dt = \frac{1}{\sqrt{4j^2 - 2}} (P_j(\xi) - P_{j-2}(\xi))
\end{equation}

for $j = 2, 3, \ldots$ are based on the Legendre polynomials

\begin{equation}
P_k(\xi) = \frac{1}{2^k k!} \frac{d^k}{d \xi^k} (\xi^2 - 1)^k, \quad k = 0, 1, \ldots
\end{equation}

Because of

\begin{equation}
N_i(-1) = N_i(1) = 0, \quad i = 3, 4, \ldots
\end{equation}

the functions $N_i(\xi), i = 3, 4, \ldots$ are called internal modes. Due to the orthogonality property of the Legendre polynomials it follows

\begin{equation}
\int_{-1}^{1} \frac{dN_i}{d\xi} \frac{dN_j}{d\xi} \, d\xi = \delta_{ij}, \quad i, j \geq 3,
\end{equation}

having an immediate impact on the condition number of the stiffness matrix. In [10] it is demonstrated, that for two-dimensional isotropic elasticity the condition number is reduced by one order of magnitude if hierarchic shape functions instead of the classical Lagrangian shape functions are used. Two- and three-dimensional \textit{ansatz functions} can be easily constructed by simply building the tensor product of one-dimensional hierarchic shape functions.

1.2 Hierarchic shape functions for hexahedral elements

For a three-dimensional discretization we consider a hexahedral element formulation (see Figure 1) also based on the one-dimensional \textit{ansatz functions}. 
\[ \Omega_h = \left[ (-1, 1) \times (-1, 1) \times (-1, 1) \right] \]

Figure 1: Standard hexahedral element \( \Omega_h \): definition of nodes, edges, faces and polynomial degree

Three different types of \textit{ansatz} spaces have been implemented: the \textit{trunk space} \( S_{ts}^{p_\xi,p_\eta,p_\zeta}(\Omega_h) \), the \textit{tensor product space} \( S_{ps}^{p_\xi,p_\eta,p_\zeta}(\Omega_h) \) and the \textit{anisotropic tensor product space} \( S^{p,p,p}(\Omega_h) \). A detailed description of the three \textit{ansatz} spaces is given in [6] and the literature listed there. For the definition of the spaces \( S_{ts}^{p_\xi,p_\eta,p_\zeta}(\Omega_h) \) and \( S^{p,p,p}(\Omega_h) \) see also Szabó and Babuška [8].

In three dimensions, the shape functions can be classified into four groups: \textit{nodal modes}, \textit{edge modes}, \textit{face modes} and \textit{internal modes} (see [8, 6]). The difference between the trunk space and the tensor product space is relevant for the face modes and the internal modes, only.

The \textit{nodal modes} \( N_{i_{1},i_{2},i_{3}} \) are the standard trilinear shape functions, well known from the isoparametric eight-noded brick element, where \( (\xi, \eta, \zeta) \) are the local coordinates of the \( i \)-th node. \textit{Edge modes} are defined for each individual edge separately. If we consider e.g. edge \( E_i \), the corresponding edge modes read: \( N_{i_{1},i_{2},i_{3}}^{E_i} = \frac{1}{8} (1 + \xi_i) (1 + \eta_i) (1 + \zeta_i) \), \( i = 1, \ldots, 8 \) are the standard trilinear shape functions, well known from the isoparametric eight-noded brick element, where \( (\xi, \eta, \zeta) \) are the local coordinates of the \( i \)-th node. \textit{Edge modes} are defined for each individual edge separately. Considering the trunk space \( S_{ts}^{p_\xi,p_\eta,p_\zeta}(\Omega_h) \) and e.g. face \( F_i \), the corresponding face modes read: \( N_{i_{1},i_{2},i_{3}}^{F_i} = \frac{1}{2} (1 - \zeta_i) \phi_i(\xi) \phi_j(\eta) \), \( i + j = 4, \ldots, \max\{p_\xi, p_\eta\} \). The \textit{internal modes} (considering again the trunk space) \( N_{i_{1},i_{2},i_{3}}^{int}(\xi, \eta, \zeta) = \phi_i(\xi) \phi_j(\eta) \phi_k(\zeta), i = 2, \ldots, p_\xi - 4, j = 2, \ldots, p_\eta - 4, k = 2, \ldots, p_\zeta - 4, i + j + k = 6, \ldots, \max\{p_\xi, p_\eta, p_\zeta\} \) are purely local and vanish at the faces of the hexahedral element. The indices \( i, j, k \) of the shape functions denote the polynomial degrees in the local directions \( \xi, \eta, \zeta \). Due to the fact, that internal degrees of freedom are purely local to the element they are eliminated by static condensation. It was shown by different authors (e.g. [1]), that this condensation of internal degrees of freedom can be interpreted as an efficient preconditioning. To
solve the remaining part of the overall equation system a PCG-solver with SSOR preconditioning is applied.

The ansatz space $S^{p,q,s}(\Omega^h_{sl})$ defines an anisotropic set of shape functions being determined by two polynomial degrees $p$ and $q$ (see Figure 1). All shape functions being of higher order in $\xi$ and $\eta$-direction are associated with the polynomial degree $p$. These shape functions correspond to the edges 1, 2, 3, 4, 9, 10, 11, 12, to the faces 1 and 6 and to all internal modes. Shape functions for faces 1 and 6 are equal to the ones of the trunk space $S^{p_\xi,p_\eta,p_\zeta}(\Omega^h_{sl})$ with $p = p_\xi = p_\eta$. $q$ defines the degree of all shape functions being of higher order in $\zeta$-direction, which are associated with the edges 5, 6, 7, 8, with the faces 2, 3, 4, 5 and with all internal modes. The modes corresponding to the faces 2, 3, 4, 5 are equal to the ones of the tensor product space $S^{p_\xi,p_\eta,p_\zeta}(\Omega^h_{sl})$ with $p = p_\xi = p_\eta$ and $q = p_\zeta$. The number of internal modes of $S^{p,q,s}(\Omega^h_{sl})$ with $p = q = p_\xi = p_\eta = p_\zeta$ is higher than the one of the trunk space $S^{p_\xi,p_\eta,p_\zeta}(\Omega^h_{sl})$ and less than the one of the tensor product space $S^{p_\xi,p_\eta,p_\zeta}(\Omega^h_{sl})$ (see [6]). Furthermore, it can be noted, that the internal modes of the space $S^{p,q,s}(\Omega^h_{sl})$ are polynomials in which the direction $\zeta$ is preferred.

Due to the built-in anisotropic behaviour of the ansatz space $S^{p,q,s}(\Omega^h_{sl})$ it is important to consider the orientation of the local coordinates of a hexahedral element. In Figure 2 it is shown, how hexahedral elements should be orientated, when three-dimensional thin walled structures are to be discretized. The local coordinate $\zeta$ of the hexahedral element corresponds with the thickness direction $z$. If the orientation of all elements is equal, then it is possible to construct discretizations where the ansatz for the in-plane and thickness direction of thin walled structures can be treated differently. The numerical examples will demonstrate that anisotropic ansatz spaces lead to efficient discretizations.

![Figure 2: Modelling plate-like structures with hexahedral elements](image)

Our implementation of the $p$-version allows not only to vary the polynomial degree
for the three different local directions but also to choose a different degree for each primary variable.

2 EFFICIENT NUMERICAL INTEGRATION

To make three-dimensional computations of complex structures affordable, the numerical quadrature, which dominates the overall computation time of such problems, has to be accelerated. Our adaptive vector quadrature follows the scheme of H.E. Hinnant [5]. The basic idea is to separate the integrand into two parts. Considering the 1-D case, the element stiffness matrix is given by

\[ k_{ij} := \int_{-1}^{1} \frac{B_{kl}(x)}{g_i(x)} \frac{D_{kl} B_{lj}(x) \det J}{h_j(x)} \, dx \quad \Rightarrow \quad k_{ij} := \int_{-1}^{1} g_i(x) h_j(x) \, dx. \tag{8} \]

\( B_{kl}(x) \) is the standard strain-displacement matrix, \( D_{kl} \) the elasticity matrix and \( \det J \) the Jacobian. In the remaining part of this paper, the indices \( i \) and \( j \) are omitted for simplicity. \( g(x) \) and \( h(x) \) can be represented by a series of orthogonal polynomials \( P_l(x) \). This leads to

\[ \int_{-1}^{1} g(x) h(x) \, dx = \int_{-1}^{1} \left[ \sum_{l=0}^{\infty} c_l P_l(x) \right] \left[ \sum_{j=0}^{\infty} d_j P_j(x) \right] \, dx = \sum_{l=0}^{\infty} c_l d_l = \mathbf{c} \cdot \mathbf{d}, \]

where \( \mathbf{c} = (c_l)_{l=0,\ldots,\infty} \) and \( \mathbf{d} = (d_l)_{l=0,\ldots,\infty} \). The integral of the product of \( g(x) \) and \( h(x) \) can thus be replaced by the dot product of the two vectors \( \mathbf{c} \) and \( \mathbf{d} \), which result from the use of the least squares method:

\[ c_l = \int_{-1}^{1} g(x) P_l(x) \, dx \quad \text{and} \quad d_j = \int_{-1}^{1} h(x) P_j(x) \, dx. \tag{10} \]

Assuming \( g(x) \) and \( h(x) \) to be polynomials, the order of \( g(x) \) determines the (finite) dimension of \( \mathbf{c} \), whereas the length of \( \mathbf{d} \) corresponds to the order of \( h(x) \). Thus, the quadrature scheme integrates each part of the integrand individually to combine the results by performing the dot product for each combination \( ij \) of the element stiffness entry \( k_{ij} \). The order of both integrands varies with \( i \) and \( j \). This can be used in two ways. On the one hand the computational effort to form the vector integrals depends on the individual order of each integrand. On the other hand the number of operations being necessary to compute the dot product depends on the dimension of the vector integral (\( \mathbf{c} \) or \( \mathbf{d} \)), arising from the lower order polynomial (either \( g(x) \) or \( h(x) \)). Using the hierarchic structure of the strain-displacement matrix, the method reduces the
number of floating point operations without taking advantage of any sparsity of the element stiffness matrix.

Whereas the work of Hinnant primarily deals with the integration method itself, our investigations focus on the question how to implement this quadrature scheme into an existing finite-element code and especially, how to deal with strongly distorted elements in order to compute efficiently and accurately the corresponding element stiffness matrices.

Let us first consider the question how to implement vector quadrature into an existing $p$-version finite-element program. Entirely different from the standard Gaussian quadrature, where the whole strain-displacement matrix is computed at each Gaussian point, vector integration works separately on each entry in the matrix, adapting the number of integration points to the order of the function that has to be integrated. To avoid any change of the complex structure of our $p$-version code, the whole $B$-matrix is still computed at each integration point. As the source of efficiency lies in the computation of the dot product, the loss due to this compromise is very small.

The second question to be addressed is how to account for distorted elements, where the order of the vector integrals is not known in advance. The integrand corresponding to distorted elements is not yet a polynomial but a ratio of polynomials, which is due to the mapping being given by the inverse of the Jacobian matrix $J^{-1}$. In contrast to rectilinear elements, it is not a priori known which entries of the vector integrals $c_I$ and $d_I$ are almost equal to zero. As the entire $B$-matrix is computed at all Gaussian points, we calculate each entry of both vector integrals — their dimensions are first determined by the chosen integration order due to the $FE$-discretization. Afterwards we find those components below a limiting value near zero. This strategy leads to an a-posteriori knowledge about the numerically necessary dimension of the vector integrals. The computational effort to perform the dot product depends now only on the number of entries in the vector integral which are different from zero.

![Figure 3: Discretization of three-dimensional plate with 194 hexahedral elements](image-url)
Vector integration always demands an even number of evaluation points. Therefore, the integration order for an odd \( p \) — assuming \( p \) being the order given by the FE-discretization — is \( p + 1 \), whereas the integration order used in case of an even \( p \) is automatically set to \( p + 2 \).

Concerning three-dimensional computations of thin-walled structures, our adaptive version of the vector integration leads even in case of strongly distorted elements to a significant speed-up. This will be demonstrated by the following example concerning a plate with columns. This structure is discretized by 194 hexahedron elements, whereas each column is meshed by two elements, the plate itself consists of one element in thickness direction. The discretization is constructed by sweeping a two-dimensional \( p \)-version mesh.

In the following, our adaptive vector integration is compared to the standard Gaussian quadrature. In Figure 4 the ratio of the CPU-time of both methods, needed for computing the element stiffness matrices, is plotted versus the polynomial degree of the FE-discretization. Applying three different ansatz spaces, the trunk space \( S_{n_{\text{f}}n_{\text{p}}}^{p,\varphi} (\Omega^h) \), the tensor product space \( S_{p_{\text{f}}p_{\text{p}}}^{p,\varphi} (\Omega^h) \) and the anisotropic tensor product space \( S_{p_{\text{f}}p_{\text{p}}}^{p,\varphi} (\Omega^h) \). All computations have been performed on a digital (alpha) machine (processor ev5 21164 with 500 MHz).

![Figure 4: ratio of CPU-times of Gaussian quadrature and vector integration](image)

In Figure 5 the relative error in energy norm is plotted versus the CPU-time needed to calculate the element stiffness matrices. Considering a FE-analysis with approximately 6\% error in energy, a reduction of CPU-time (element stiffness matrices) by one order of magnitude is obtained, when using the new quadrature scheme. The 'exact' strain energy has been estimated by a Richardson extrapolation of the last three results obtained from the calculations with the tensor product space \( S_{p_{\text{f}}p_{\text{p}}}^{p,\varphi} (\Omega^h) \), \( p = 1, \ldots, 7 \).
3 A numerical example with curved surfaces

Figure 6 shows a CAD-model of Favinis famous gas station near Mailand (see [7]). A significant difficulty in a classical finite element analysis for this construction is the transition from thin- to thick-walled construction parts. Whereas elements of different dimension types (shells, beams, bars and solids) would have to be used in the classical h-version, the p-version, coupled to the geometric model, can discretize the complete structure strictly three-dimensional.

Figure 6: CAD-model of gas-station (Favini)

In Figure 7 the p-version hexahedron mesh using 178 elements with a zoomed detail is plotted. Von Mises stresses are depicted in Figure 8 on the upper surface and for a detail in the transition from the roof to a column of the construction. For a computation with polynomial degree \( p = 6 \) an overall computational time of about 45 minutes on a digital (alpha) machine (processor ev6 21264 with 500 MHz) yielding an estimated error in energy norm of approximately 10% was necessary. More detailed results for this example can be found in [4].
4 CONCLUSIONS

One of the significant features of the p-version finite element analysis is the robustness of high order elements. They are free of locking and accurate results can be obtained even for elements with very high aspect ratio. This robustness opens the possibility to use high order solid elements to simulate thin-walled structures, which can usually only be discretized by plate or shell elements. Yet, a naive implementation of high order elements yields a code, which will need prohibitively large computational resources even for small sizes of three-dimensional numerical models. Using
anisotropic ansatz spaces, efficient preconditioning of the system matrix and special integration techniques for high order element matrices together with adequate data structures and a careful implementation, finite element codes can be developed, which are able to analyse realistic thin-walled 3D-problems with only moderate computational effort. As a major advantage of this approach no a priori assumptions on the kinematics of dimensionally reduced models are necessary and no special elements have to be introduced in regions, where thin-walled structures are joint to general three-dimensional continua.

REFERENCES


