Parallel Shape Optimization of Three-Dimensional Continua with High-Order Finite Elements

M. Baitsch, Y. Hu, D. Hartmann
Faculty of Civil Engineering, Ruhr-University of Bochum
matthias.baitsch@rub.de, ce.yphu@yahoo.de, hartus@inf.bi.rub.de

1. Abstract
This contribution is on shape optimization of 3D continua using high-order finite elements. Since in the $p$-version of FEM the discretization error is controlled by raising the polynomial degree, adaptive remeshing procedures during optimization are not required. The element geometry is constructed using the blending function method in conjunction with NURBS surfaces in order to efficiently describe the geometry in terms of only a few design variables. Design elements for high-order finite elements are introduced which facilitate the generation of geometrical optimization models largely independent from the finite element mesh. A two level parallel scheme is employed in order to run the optimization on a cluster computer with about 100 CPUs.

2. Keywords: High-order FEM, 3D shape optimization, NURBS, parallel computing.

3. Introduction
Shape optimization of continuous structures requires careful attention to the discretization error. In classical shape optimization, $h$-adaptive methods are utilized in order to attain reliable shape optimization results (for applications in two dimensions, see e.g. [15, 14]). However, the automatic construction of adapted meshes requires substantial effort, particularly in the case of three-dimensional problems. An alternative to $h$-adaptivity is the $p$-version of FEM where a small number of comparatively large elements is used. As high-order elements can have complex shape and are insensitive against a distorted geometry, the mesh can be left unmodified throughout the optimization procedure.

High-order finite element methods have been employed for optimization by several authors. For instance, Salagame and Belegundu utilize $p$-adaptivity for two-dimensional problems in [12]. Applications in three dimensions are given by Thanedar and King in [18] and more recently by Tanino et al. in [17]. However, the problem how to efficiently represent the three-dimensional geometry in terms of design variables has yet not addressed in detail. In this contribution, we present NURBS surfaces, design elements, connected surfaces and a special refinement technique which support the creation of three-dimensional optimization models based on high-order finite element methods. In order to solve the resulting optimization problems within tolerable time, parallel computing is mandatory. We present a two level parallelization scheme which addresses both optimization and structural analysis.

The outline of the paper is as follows: First, the basic principles of the $p$-version of FEM in three dimensions are outlined. Then, concepts for geometry representation and mesh refinement suitable for optimization purposes are presented. An approach to parallel optimization and analysis is described thereafter. Finally, two application examples (a shell structure and an arch-dam) exemplify our methods.

4. The $p$-version of FEM in Three Dimensions
In the $p$-version of FEM a mesh consisting of a small number of large elements is used while the polynomial degree $p$ is adapted in order to control the discretization error. The main differences to the traditional $h$-version are the element shape functions and the geometry model.

4.1. Element Shape Functions
Shape functions for high-order volume elements can be defined on various types of reference domains. Because of good approximation properties and a straightforward construction scheme, we use hexahedral elements. Following Szabó and Babuška [16], we start from the one-dimensional hierarchic shape...
functions

\[ N_1^1(\xi) = 0.5(1 - \xi), \quad N_2^1(\xi) = 0.5(1 + \xi), \]  
\[ N_3^1(\xi) = \sqrt{\frac{2i - 3}{2}} \int_{-1}^{\xi} P_{i-1}(t) \, dt, \quad i = 3, 4, \ldots, p + 1. \]  

defined on \( \hat{\Omega} = [-1, 1] \) for any polynomial degree \( p \geq 1 \). Here, \( P_i \) are the well known Legendre polynomials. The three-dimensional shape functions \( N_3^3 \) on the reference cube \( \hat{K} = [-1, 1]^3 \) are generated by the product

\[ N_{0(i,j,k)}^3(\xi_1, \xi_2, \xi_3) = N_1^1(\xi_1) N_1^1(\xi_2) N_1^1(\xi_3), \]  
\[(i, j, k) \in A \subset \{1, \ldots, p_1 + 1\} \times \{1, \ldots, p_2 + 1\} \times \{1, \ldots, p_3 + 1\}\]

of the one-dimensional hierarchic shape functions. The number of triples \((i, j, k)\) in \( A \) is the dimension \( d \) of the corresponding function space where \( p_i, i = 1, 2, 3 \) are the polynomial degrees for each coordinate direction. An indexing function \( o : A \rightarrow 1, 2, \ldots, d \) defines a certain ordering of the shape functions. Based on their properties, a classification of the shape functions into four groups is established:

1. **Vertex modes** take a value of one at one vertex and vanish at all other vertices.
2. **Edge modes** vanish at all vertices and at each edge but one.
3. **Face modes** vanish at all edges and at each face but one.
4. **Internal modes** vanish at all faces and are thus purely local to the element.

Fig. 1(a) shows one instance of the shape functions for each mode group. Based on the choice of the index set \( A \), different function spaces can be constructed. Three types are commonly used: (i) the tensor product space (PS) spanned by \((p_1 + 1)(p_2 + 1)(p_3 + 1)\) shape functions, (ii) the trunk space (TS) having fewer face and internal modes and (iii) the anisotropic tensor product space (APS) which prefers one distinct coordinate direction. Details are given by Düster et al. in [3]. The dimensions of the three spaces are shown for an isotropic polynomial degree pattern \((p_1 = p_2 = p_3)\) in Fig. 1(b). As a consequence of the large number of shape functions, elements in 3D \( p \)-FEM can have up to 1000 degrees of freedom and more. For the analysis of thin-walled structures, an anisotropic distribution of polynomial degrees with \( p_1, p_2 > p_3 \) (see [3] and the application example in Sec. 6.1) reduces the computational effort.

4.2. Geometry Model – The Blending Function Method

Because of the large element size used in the \( p \)-version of FEM, the geometrical mapping from the reference domain into physical coordinates has to be able to represent complex element shapes. The blending function method by Gordon and Hall [4] provides a framework to construct the geometry mapping \( Q : \hat{K} \to \mathbb{R}^n \) in terms of surfaces \( F_i : \hat{Q} \to \mathbb{R}^3, i = 1, \ldots, 6 \), edges \( E_i : \hat{\Omega} \to \mathbb{R}^3, i = 1, \ldots, 12 \)
and vertices $X_i \in \mathbb{R}^3, i = 1, \ldots, 8$. The general formulation is

$$Q(\xi_1, \xi_2, \xi_3) = + \sum_{i=1}^{8} N_i^3(\xi_1, \xi_2, \xi_3)X_i$$

$$- N_1^3(\xi_2)N_1^3(\xi_3)E_1(\xi_1) - N_2^3(\xi_1)N_3^3(\xi_3)E_2(\xi_2) - N_2^3(\xi_2)N_1^3(\xi_3)E_3(\xi_1)$$

$$- N_1^3(\xi_1)N_3^3(\xi_2)E_4(\xi_3) - N_2^3(\xi_1)N_1^3(\xi_2)E_5(\xi_3) - N_2^3(\xi_2)N_1^3(\xi_1)E_6(\xi_3)$$

$$- N_2^3(\xi_1)N_2^3(\xi_2)E_7(\xi_3) - N_1^3(\xi_1)N_2^3(\xi_2)E_8(\xi_3) - N_1^3(\xi_2)N_2^3(\xi_1)E_9(\xi_3)$$

$$- N_2^3(\xi_1)N_3^3(\xi_2)E_{10}(\xi_3) - N_2^3(\xi_2)N_3^3(\xi_1)E_{11}(\xi_3) - N_1^3(\xi_1)N_3^3(\xi_2)E_{12}(\xi_3)$$

$$+ N_1^3(\xi_3)F_1(\xi_1, \xi_2) + N_1^3(\xi_2)F_2(\xi_1, \xi_3) + N_2^3(\xi_1)F_3(\xi_2, \xi_3)$$

$$+ N_3^3(\xi_2)F_4(\xi_1, \xi_3) + N_1^3(\xi_1)F_5(\xi_2, \xi_3) + N_2^3(\xi_3)F_6(\xi_1, \xi_2)$$

(4)

where $N_i^3, i = 1, \ldots, 8$ are the vertex modes of the three-dimensional shape functions (3). If only some of the faces and edges are curved, Eqn. (4) simplifies significantly.

5. Optimization-Oriented Geometry Representation

In an abstract setting, the shape optimization problem is to find a generalized design function $s$ from the set of admissible design functions $D$ minimizing an objective functional $\mathcal{J} : D \rightarrow \mathbb{R}$,

$$\min_{s \in D} \mathcal{J}(s).$$

(5)

Here, $s$ represents the shape of the structure, $\mathcal{J}$ reflects the primary design intention and the requirements on the structural design are expressed in terms of $D$. An extended formulation of problem (5) including an additional state function $u$ is given in [9]. However, in the continuous form (5), the optimization problem can not be solved with the methods at hand. In fact, the search for the best design is restricted to a finite dimensional space $D_h \subset D$ where the design function $s_h \in D_h$ depends on a design vector $x \in \mathbb{R}^n$. By that, the original problem (5) is transformed into the standard constrained optimization problem

$$\min_{x \in S} f_0(x) \quad \text{with} \quad S = \{x \in \mathbb{R}^n : f_j(x) \leq 0, i = 1, \ldots, m, x_1 \leq x \leq x_n\}$$

(6)

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function and $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \ldots, m$ are constraint functions. The computational resources required to solve (6) and the quality of the design corresponding to the solution is governed by the geometrical discretization. Two contradictory criteria have to be balanced:

1. A large number of design variables slows down the solution procedure.
2. If $D_h$ is too small, potentially desired results are excluded.

Unlike finite element meshes, geometrical models for design optimization can not be generated automatically. For practical applications it is therefore important to provide the designer with tools that support the creation of such models efficiently.

In the following sections, we present NURBS surfaces for the discretization of the design function $s$ and design elements for high-order FEM that support the construction of adequate geometry models.

5.1. NURBS Surfaces

Today, Non Uniform Rational B-Splines (NURBS) are the de facto standard in computer aided geometric design (CAGD). In high-order FEM, NURBS surfaces are used for the face function $F_i$ in Eqn. (4).

We start with the one-dimensional B-Spline basis functions. Let the vector of parameter knots $k = [k_0, \ldots, k_m]$ be a non-decreasing sequence of real numbers (in order to simplify the integration with the blending function method, we choose $k_0 = -1$ and $k_m = 1$). Based on the B-Spline basis functions of degree zero

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{for } k_i \leq \xi < k_{i+1} \\ 0 & \text{otherwise,} \end{cases}$$

(7)
Figure 2: Design element for high-order FEM

the basis functions of higher polynomial degree

$$N_{i,p}(\xi) = \frac{\xi - k_i}{k_{i+p} - k_i} N_{i,p-1}(\xi) + \frac{k_i + 1 - \xi}{k_{i+p+1} - k_{i+1}} N_{i+1,p-1}(\xi),$$

are defined recursively. The NURBS surface $S: \hat{\Omega} \rightarrow \mathbb{R}^3$ of degree $(p_1, p_2)$ is

$$S(\xi_1, \xi_2) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p_1}(\xi_1)N_{j,p_2}(\xi_2)w_{i,j}P_{i,j}$$

where $P_{i,j} \in \mathbb{R}^3$ are control points and $w_{i,j}$ are control point weights which give additional control on the shape of the surface. A comprehensive survey on NURBS surfaces can be found in [10].

Using NURBS, the shape of individual finite elements is accurately described in terms of the control points of which the coordinates are design variables in the optimization problem. For shape optimization, NURBS have various properties that make them appealing: The basis functions (7) have local support and form a partition of unity. The rational term makes it possible to construct regular as well as irregular surfaces in a unified fashion. Furthermore, the number of control points can be chosen arbitrarily such that the accuracy of the geometry representation can be adapted to the problem at hand.

Using individual NURBS surfaces for each curved face of a finite element is possible but has severe disadvantages. The problem is that the finite-element mesh is designed according to analysis requirements such that additional dependencies between control points have to be introduced in order to ensure a smooth geometry.

5.2 Design Elements for High-Order FEM

Design elements have been adopted early in shape optimization, see e.g. Imam [6] for an application to three-dimensional problems. The basic idea of design elements is to introduce geometry functions (often Bézier splines or Coons patches) that adequately define the shape of the structure in terms of only a few design variables. In the conventional design element technique, nodes of the FE model are attached to the parametric functions and after a change of the design, nodal coordinates are updated. For high-order finite elements, this simple approach is clearly not applicable (see Sec. 4.2).

In order to apply the design-element technique to high-order finite elements, an extension to the original concept is proposed where, in addition to the nodes, also elements edges and faces are tied to the design element. The idea is to define element edge functions $E_j$ and face functions $F_k$ in terms of a single higher-level geometrical mapping. The high-order design element is basically a reference domain $Q_D \subset \mathbb{R}^2$, a mesh $M$ on $Q_D$ and a geometrical mapping $D: Q_D \rightarrow \mathbb{R}^3$ which is the actual shape of the design element. For constructing the face mapping $F_k$, we establish the linear transformation $\tilde{F}_k: \hat{\Omega} \rightarrow Q_k$ from the reference domain $\hat{\Omega}$ into the mesh cell $Q_k$ and obtain the face mapping by the composition

$$F_k = D \circ \tilde{F}_k.$$ (10)

In the same way, we construct the edge function

$$E_j = D \circ \tilde{E}_j.$$ (11)
Figure 3: Geometry of a shell with and without design elements

where $\tilde{E}_j : \tilde{\Omega} \rightarrow \Omega_d$. Figure 2 illustrates these relations.

The mesh on the domain of the design element is conveniently defined in terms of parameters $(\xi_1, \xi_2)$ corresponding to the vertices of the mesh cells $Q_k$ and thus determine the mappings $\tilde{E}_j$ and $\tilde{F}_k$. In many situations, it is most practicable to select the parameters $(\xi_1, \xi_2)$ and to leave them unchanged during optimization. However, in order to ensure a regular geometry for all elements, it may be required to adapt parameters during the optimization procedure. Such requirements are formulated in terms of a nonlinear equation

$$f(\xi_1, \xi_2) = 0, \quad f : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (12)$$

which are solved after each change of the design element using a Newton procedure.

The advantage of design elements is clearly visible from the example given in Fig. 3 which shows a shell-like structure modeled by means of 10 hexahedral elements. Using the design element technique as shown in Fig. 3(a), the shape of the whole shell is described in term of one NURBS surface (of which the control polygon is shown in blue color). In contrast, if a single NURBS surface is used for each element, additional relations amongst the control points would have to be introduced in order to avoid an inappropriate geometry like that shown in Fig. 3(b).

5.3 Mesh refinement

Numerous engineering structural analysis problems in three dimensions involve stress concentrations or singularities at inclined edges and clamped boundary conditions. Such stress concentrations require additional mesh refinement either to resolve stresses with sufficient accuracy or in order to avoid pollution effects (see [1]). To facilitate the use of $hp$-adaptivity in design optimization, an automatic refinement procedure for hexahedral elements based on the element map $Q$ of the refined element has been developed.

The basic idea for such an automatically generated refinement (see Fig. 4) is closely related to the design elements from Sect. 5.2. First, a mesh is created on the reference domain $K$ using two specified refinement factors $\sigma_1, \sigma_2$. Then, the geometrical mappings $Q_i$ for refined elements are obtained by combining a linear transformation $\tilde{Q}_i : K \rightarrow K_i$ with the original mapping $Q$ as

$$Q_i = Q \circ \tilde{Q}_i. \quad (13)$$
In order to obtain the additional edge and face functions, three design elements, one for each affected face, are created. Of course, this refinement procedure can be applied recursively as shown in Fig. 4.

5.4 Displaced Surface Functions
For thin-walled structures, it is adequate to describe the overall shape and thickness independently. Using the design elements described above, this is realized by translating one surface along the thickness direction. Such a displaced surface function $D_d : Q_d \rightarrow \mathbb{R}^3$ is constructed based on a thickness function $t : Q_d \rightarrow \mathbb{R}$ and direction vectors $d_i$ using

$$D_d(\xi_1, \xi_2) = D(\xi_1, \xi_2) + t(\xi_1, \xi_2) \sum_{i=1}^{n} d_i N^2_i(\xi_1, \xi_2)$$

(14)

where $D$ is the reference geometry and $N^2_i$ are Lagrangian shape functions.

6. Parallel Optimization and Analysis
Parallel computing has a long history in design optimization and various solution techniques have evolved. For instance, in multilevel structural optimization, the original optimization problem is decomposed into a number of smaller non-interacting subproblems coupled on a coordination level [19]. However, when using the $p$-version of FEM for design optimization, a simpler approach where inherent parallelism is exploited directly also seems to be promising. Basically, there are three potential targets for parallelization: The optimization procedure, the computation of element matrices and the solution of the linear system.

Parallelization on the optimization level hinges on the type of optimization strategy used. The class of evolutionary algorithms and related techniques like particle swarm optimization are trivial to parallelize because in each iteration step, a large number of function values can be computed without inter-process communication [5,8]. However, gradient-based algorithms, if applicable to the problem at hand, are preferable because tremendously fewer function evaluations are required. The potential for parallelization here is in the approximation of derivatives using finite difference-schemes and in a parallel variant of the line-search algorithm [13]. In this context, the maximum number of parallel processes depends on the number of design variables $n$.

For typical engineering design optimization problems, the number of design variables $n$ is too small to take full advantage of today’s cluster computers with 100 and more CPUs. Therefore, we introduce a second level of parallelization by performing a parallel structural analysis. In the $p$-version, the overall analysis time is dominated by the computation of element matrices and the solution of the linear system. Since the computation of element matrices takes a large amount of the overall analysis time (between 50% and 90% depending on the problem and the polynomial degree) and is easy to parallelize, we do not consider parallel equation solvers in this contribution. In the future, improvements in parallel efficiency could be obtained by incorporating a domain decomposition algorithm as presented in [11].

In our parallel algorithm, function values

$$f(x_1) = [f_0(x_1), \ldots, f_m(x_1)] \quad \ldots \quad f(x_q) = [f_0(x_q), \ldots, f_m(x_q)]$$

are computed for a set of $q$ design vectors whereas each function evaluation includes a finite element analysis which is itself performed in parallel. The algorithm is based on the well known Manager-Worker pattern and uses point-to-point communication. The outline of the procedure for one parallel function evaluation is:

1. The manager process receives the set of $q$ design vectors $x_1, \ldots, x_q$. The available processes are arranged in $q$ groups and the manager sends the design vector $x_i$ to the $i$-th process group.

2. Each processor in the $i$-th group updates the finite-element model according to $x_i$. Collectively, all processors in the group assemble the global stiffness matrix.

3. One process in each group solves the linear system of equations. This processor also computes the function values $f_0(x_i), \ldots, f_m(x_i)$ based on the analysis results and sends the values back to the manager process.
A schematic view on the algorithm is given in Fig. 5 where two use cases are depicted: Parallel evaluation for one design vector and parallel evaluation for several design vectors. In the figure, one gray box represents one parallel process; blue boxes represent groups of processes. The main advantage of the proposed scheme is its simplicity and the straightforward implementation. However, because of the structure of the optimization problems studied in this contribution, reasonable parallel efficiency is achieved.

7. Numerical Examples

The methods described in this paper have been implemented into a FEM system developed by the authors [2]. Parallel computations have been executed on a 112 CPU Linux cluster (AMD Opteron, 2.4GHz) equipped with a standard Gigabit Ethernet communication infrastructure.

7.1. Stiffness Optimization of a Shell Structure

The first numerical example is the optimization of the shell structure shown in Fig. 6(a). A similar problem has been investigated by Kimmich in [7] using shell elements. The concrete shell covers an area of 40 m by 40 m, has an initial height of 14.2 m and an initial thickness of 15 cm. Linear elastic material with $E = 3 \times 10^7 \text{kN/m}^2$, $\nu = 0.2$ and $\gamma = 25 \text{kN/m}$ is assumed. As loading, dead weight, an additional surface load of $q_d = 4 \text{kN/m}^2$ and a wind pressure of $q_d = 1.6 \text{kN/m}^2$ from an angle of 35° degrees are applied. At the four vertices, the shell is softly supported at the lower edge. The structure is discretized with 96 curved hexahedral elements whereas the polynomial degrees of the individual directions are $p_1 = p_2 = 9$ and $p_3 = 3$. The resulting linear system has 30816 unknowns. For the initial design, the elastic strain energy is $167.28 \text{kNm}$.

In the shape optimization, the strain energy is minimized which is equivalent to the maximization of the overall structural stiffness. The only constraint is an upper limit of 800 tons to the total weight of the shell. In the geometrical model, the shape of the upper surface is described by one design element in terms of a NURBS surface with $7 \times 7$ control points. Because of symmetry conditions, there are 10 corresponding design variables. Using the displaced function (14), the shape of the lower surface is defined in terms of the upper surface. The direction vectors $d_i$ in (14) in normal direction to the upper surface and are prescribed at 9 points in conjunction with quadratic Lagrangian shape functions. As thickness function $t$, a NURBS with $2 \times 2$ control points is utilized such that three design variables specify the shell thickness.

For optimization, Schittkowski’s NLPQLP algorithm [13] is utilized because it incorporates a parallel line-search algorithm. In Fig. 6(b), the improved shape is depicted, the corresponding strain energy is $52.16 \text{kNm}$ which is less than a third of the initial value. Fig. 7 shows a comparison of the equivalent stress for the initial and the optimized design. Note the unsymmetric stress distribution due to the wind load. In the optimization, the overall height of the structure has been slightly reduced to 13.6 m.
which reduces the influence of the wind load. Furthermore, the curvature of the free edge is increased in the middle section. It can clearly be seen that the overall stress-level is significantly lower for the new geometry.

The optimization procedure required 649 function evaluations and took 5356 s computing time on 100 CPUs. One function analysis carried out sequentially takes 270 s such that the overall speedup is 32. Taking into account the fact that our implementation has not been tuned for performance yet, we consider this as a good result. However, substantial improvements in parallel efficiency are expected for the future.

7.2. Weight Minimization of an Arch Dam
In this application example, the arch-dam shown in Fig. 8 is considered. The structure is modeled using 172 hexahedral elements arranged in two layers. At the clamped boundary, two refinement layers along both edges are introduced in order to isolate the local stress concentration (see details in Fig. 8). For the analysis, a uniform polynomial degree $p = 7$ is utilized resulting in 31032 degrees of freedom. Material parameters are the same as in the previous example. Dead weight of the dam and the water pressure are applied as loading. The volume of the initial design is $1.480 \cdot 10^6 \text{m}^3$.

The objective in shape optimization is the minimization of the volume with respect to a maximum stress of $15000 \text{ kN/m}^2$. For the evaluation of stress constraints, elements in the boundary layer (marked blue in the detail in Fig. 8) are excluded. This is necessary because of the purely local stress concentration.
at the clamped boundary. The geometry of the dam is defined by one design element with a NURBS having $5 \times 5$ control points. The position of the control points are linked to a NURBS curve described by 5 design variables. Using 5 more thickness design variables, the outer surface of the dam is realized by the translated function (14).

Again, the NLPQLP algorithm is used for the optimization. Fig. 9 depicts the optimized shape of the structure which has a volume of $1.042 \cdot 10^6 \text{m}^3$ which corresponds to a weight reduction by 30%. In the optimization, the curvature of the upper edge is increased and adapted to the asymmetric shape of overall structure. From the comparison of the stress distribution for the initial and the optimized design, it can be seen that the material is used more efficiently.

8. Conclusions
In this paper, an approach to shape optimization of three-dimensional continua using the p-version of FEM has been presented. Advanced concepts for the geometrical model and mesh refinements are applied to engineering structures. Particularly, the concept of design elements substantially simplifies the generation of the optimization models with only a small number of design variables. The two numerical examples show the applicability of the methods to engineering problems.

9. References


