Raino Mäkinen

On computer aided shape optimization

Kybernetika, Vol. 29 (1993), No. 3, 256--269

Persistent URL: `http://dml.cz/dmlcz/125098`

KYBERNETIKA — VOLUME 29 (1993), NUMBER 3, PAGES 256–269

ON COMPUTER AIDED SHAPE OPTIMIZATION

RAINO A. E. MÄKINEN

We discuss the implementation of shape optimization software. Algebraic sensitivity analysis for a Poisson problem is studied in detail. A numerical example is given.

1. INTRODUCTION

Optimal shape design may be defined as the rational establishment of a geometric design that is the best of all possible designs within a prescribed objective and a given set of geometrical and behavioural limitations.

During the last two decades optimal shape design has become a broad multidisciplinary field which finds applications in aeronautical, civil, mechanical, electrical, nuclear, and off-shore engineering, as well as in space technology. Although optimal shape design problems have attracted scientists over the centuries, it is modern high-speed computers combined with modern numerical algorithms (finite elements, nonlinear programming) that have made it possible to construct optimal shapes for non-trivial industrial applications. For an introductory treatment of the subject we refer to monographs [1] and [10]. More detailed treatment can be found in monographs [2], [9], [11] and [18].

Parameters chosen to describe the geometry of the system are called *design parameters.* The design parameters can be either finite dimensional (vector) or distributed parameters. Optimal shape design problems can be divided roughly into three classes: optimal sizing, domain optimization and topology optimization.

Optimal sizing usually deals with structural optimization. We assume that the layout of the structure is given and we try to find optimal sizes of the structural members. The sizes of the members are chosen as the design parameters that can be of a vector or distributed type. Two typical sizing problems in structural optimization are optimal sizing of a beam (distributed parameter) and of a frame (vector parameter).

In domain optimization, also known as variable boundary optimization, the shape of the two- or three-dimensional domain is sought. Usually the problem is reduced to finding a vector function which defines the unknown boundary.

Topology optimization deals with the search for optimal lay-out of the system. The problem is, for example, to find out which ones of the members of the truss should exist such that the weight of the truss is minimized and the truss can carry a given load without collapsing. Topology optimization problems have an on-off nature and are therefore extremely difficult to solve in the distributed case.

2. ABSTRACT SHAPE OPTIMIZATION PROBLEM

2.1. The continuous problem

Let us consider a general optimal shape design problem that can be stated semi-mathematically as follows: Suppose that we have a family U_{ad} of *geometrically admissible* design parameters α. For each $\alpha \in U_{ad}$ we have a state problem $A(\alpha; u(\alpha)) = 0$ which defines a state $u(\alpha)$. In this work A is assumed to be an elliptic differential operator. It is further required that the pair $(\alpha, u(\alpha))$ satisfies some *behavioural* (or *state*) *constraints* $(\alpha, u(\alpha)) \in S$. Finally, we have an optimization criterion (scalar function) $J(\alpha, u(\alpha))$. The problem is to find a design parameter α^* which minimizes $J(\alpha, u(\alpha))$ and satisfies the geometrical and behavioural constraints. Symbolically

$$\text{Minimize } J(\alpha, u(\alpha)) \tag{1}$$

subject to

$$A(\alpha; u(\alpha)) = 0, \tag{2}$$

$$\alpha \in U_{ad}, \tag{3}$$

$$(\alpha, u(\alpha)) \in S. \tag{4}$$

In many practical design problems, it is not obvious what the cost function should be and how it should be related to design parameters. Considerable engineering and mathematical insight and experience are needed to identify a proper cost function in large industrial design problems. In some situations a single cost function cannot be identified. For example we may want to minimize the weight of a structure and at the same time minimize the average stress within it. These are called *multicriteria optimization problems.* There is *no* general method for solving such optimum design problems. These problems are usually transformed into a sequence of scalar optimization problems by defining a composite cost function as a weighted sum of all the cost functions or selecting one criterion as the cost functional and treating the remaining ones as constraints.

The choice of the parametrization at the beginning of the optimal design process determines a restricted class of the optimal solutions that can be achieved. For example if a simply connected domain is assumed at the beginning, the optimal shape that is obtained, if it exists, is within this class of domains, although the true optimal shape may be doubly connected.

The state-of-the-art systems for computer aided optimal shape design of distributed structures require that the basic topology of the structure is given. However, the natural setting of many shape optimization problems in structural mechanics is the following: determine, for every point in the space, if there is material or not, i.e. find a characteristic function of the optimal domain. The idea proposed by Bendsøe and Kikuchi in [4] is to divide the optimum structural design problem into two subproblems. In the first subproblem, an approximately optimal topology is found by transforming the optimal shape design problem into a material distribution problem using composite materials. Two material constituents, substance and void, are considered, and the microscopic optimal void distribution is considered instead of shape optimization by boundary variations in the usual sense. In the second subproblem, boundary variation techniques are used to obtain conventional design. The solution of the previous step is used to define the basic topology of the structure for variable boundary optimization by considering only macroscopic voids as possible voids in the final structure. Using this approach very general shapes can be produced. In nonstructural applications, however, the meaning of voids is not clear and they are usually not allowed.

The question of existence and uniqueness of the solution is important from both theoretical and practical points of view. Usually, the existence is proved under some compactness assumptions. In many cases these assumptions are quite artificial. As the mapping $\alpha \mapsto u(\alpha)$ is usually nonconvex the uniqueness may fail. In any case, to prove the uniqueness is much more difficult than to prove the existence.

2.2. The discrete problem

In order to solve the design problem (1) – (4) on a computer we must define a finite dimensional approximation of it. In this work we assume that the approximation is done using the finite element method (FEM). Let h stand for the *discretization parameter* (mesh size). The finite dimensional approximation of (1) – (4) could be read symbolically as follows

$$\text{Minimize } J_h(\alpha_h, u_h(\alpha_h)) \tag{5}$$

subject to

$$A_h(\alpha_h; u_h(\alpha_h)) = 0, \tag{6}$$

$$\alpha_h \in U^h_{ad}, \tag{7}$$

$$(\alpha_h, u_h(\alpha_h)) \in S_h. \tag{8}$$

Equation (6) is a system of (non)linear algebraic equations arising from the FEM-approximation of (2) and J_h, U^h_{ad} and S_h are finite dimensional approximations of J, U_{ad} and S, respectively.

The approximate optimization problem and the original one should be checked for compatibility in the sense that the solution of the approximate optimization problem is "close" in some sense to the solution of the original problem.

The question of convergence of a sequence of solutions of the approximate optimal design problems is almost completely overlooked by the engineering community. This is probably due to the fact that engineers consider the problem (5) – (8) as an approximation of the physical problem instead of the mathematical problem (1) – (4). However, it is rather easy to find examples where the approximate problem always has a solution but the sequence of these solutions does not converge. For example in plane elasticity it is perfectly legimate to impose point loads and consider pointwise stresses in the finite element model for fixed h. However, point loads prevent the convergence of the finite element method and pointwise stresses have no meaning in the continuous case unless the stress field is smooth.

As optimal shape design problems are usually nonconvex one should apply methods of global optimization. As these methods are extremely expensive in the case of problem (1) - (4) one is usually satisfied with a possible local optimum. On the other hand, in practice one is often satisfied with *feasible* design, i.e., to find a such that (2) - (4) hold.

In the numerical computation of local optimum there are two basic techniques. In the first technique the optimality conditions of (1) - (4) are solved. This is called the *indirect method.* The other technique is to use mathematical programming methods to find a sequence of designs which converge to a local optimum. This method is called *direct.* In practice the direct method is usually used due to its better stability. However, in certain situations, as in the case of large scale optimal sizing problems, the indirect method has proved to be useful.

3. REMARKS ON SOFTWARE AND PROGRAMMING

3.1. Solving shape optimization problem with existing software

Let us consider the general nonlinear programming problem

$$\min_{\mathbf{a}\in\mathrm{R}^M} \tilde{F}_0(\mathbf{a}) \tag{9}$$

subject to constaints

$$\tilde{F}_i(\mathbf{a}) = 0 \quad i = 1,\dots,N_c. \tag{10}$$

Here $\tilde{F}_i : \mathrm{R}^M \to \mathrm{R}$ are nonlinear, possibly nonsmooth functions. Bound constraints and linear constraints are omitted for simplicity. As shape optimization problems usually are highly non-linear, possibly non-smooth and have large number of constraints the use of well-tested, published computer codes is preferred. Program codes for the solution of (9) - (10) assume user written subroutines for the calculation of functions $\tilde{F}_i$ and their gradients (or subgradients) at a point $\mathbf{a}$ given by the optimization algorithm. The cost and the constraint functions usually are implicit functions of the design variable vector $\mathbf{a}$. Therefore, the calculation of their gradients is not straightforward.

Let us assume that the functions $\tilde{F}_i$ are smooth and that the state problem is linear. The form of problem (9) - (10) in this specific case reads:

$$\min_{\mathbf{a}\in\mathrm{R}^M} \left\{\tilde{F}_0(\mathbf{a}) \equiv F_0(\mathbf{a}, \mathbf{q}(\mathbf{a}))\right\} \tag{11}$$

subject to

$$\mathbf{K}(\mathbf{a})\,\mathbf{q}(\mathbf{a}) = \mathbf{f}(\mathbf{a}), \tag{12}$$

$$\tilde{F}_i(\mathbf{a}) \equiv F_i(\mathbf{a}, \mathbf{q}(\mathbf{a})) = 0, \quad i = 1,\dots,N_c. \tag{13}$$

Here $\mathbf{K}$ and $\mathbf{f}$ stand for the stiffness matrix and force vector of (6), respectively. Vector $\mathbf{q}$ contains the nodal values of u_h.

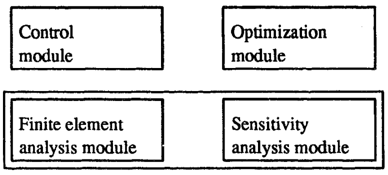

Fig. 1.

As the whole computer code needed to solve a shape optimization problem is rather long and complicated, the use of existing software is encouraged. One usually needs the modules shown in Figure 1. It is clear, however, that the standard software has some limitations. The user does not necessarily have access to the source code of all modules. This is the case when an optimization subroutine of some large numerical mathematics package is used. Some module, usually the FE-analysis module, may be a separate package that can be executed only at the operating system level.

3.2. Sensitivity analysis

The search for optimality conditions of the problem (11)–(13) in optimal shape design is called *design sensitivity analysis.* In the case of domain optimization, design sensitivity analysis is often considered the most tedious step in the numerical procedure for finding the optimal shape. A substantial amount of literature has been developed in the field of design sensitivity analysis for domain optimization problems. In the case of optimal sizing the sensitivity analysis is more straightforward.

Although the geometric sensitivity analysis is one of the most crucial steps in numerical shape optimization, it is still considered extremely elaborate and difficult even for linear problems. This is probably due to the bad form in which most of the sensitivity formulae are presented. In these formulae there are usually too much explicit dependence on the certain application or element type. This implies nonstructured programs which are difficult to debug and maintain.

Analytical differentation

Contributions to this field have been made using two fundamentally different approaches. The first approach, called algebraic differentation, uses the discretized model, based on FEM analysis, and proceeds to carry out design sensitivity analysis by differentiating the algebraic finite element equations. The second approach uses the *material derivative* of continuum mechanics to account for changes in the shape of the domain.

Algebraic formulae are obtained by differentiating the finite element equations with respect to the design variables. This approach requires that the mesh topology is fixed. The partial derivative of $\tilde{F}_i$ with respect to a_j, $j = 1, \ldots, M$ is given by

$$\frac{\partial \tilde{F}_i(\mathbf{a})}{\partial a_j} = \frac{\partial F_i(\mathbf{a}, \mathbf{q}(\mathbf{a}))}{\partial a_j} + (\nabla_{\mathbf{q}} F_i(\mathbf{a}, \mathbf{q}(\mathbf{a})))^{\mathrm{T}} \frac{\partial \mathbf{q}(\mathbf{a})}{\partial a_j}, \tag{14}$$

If $\mathbf{K}(\mathbf{a})$ and $\mathbf{f}(\mathbf{a})$ depend smoothly on $\mathbf{a}$, we may use the implicit function theorem and differentiate (12) to obtain

$$\mathbf{K}(\mathbf{a}) \frac{\partial \mathbf{q}(\mathbf{a})}{\partial a_j} = \frac{\partial \mathbf{f}(\mathbf{a})}{\partial a_j} - \frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_j} \mathbf{q}(\mathbf{a}). \tag{15}$$

The form of equation (14) is not suitable when the number of constraints is less than the number of design variables as it requires M solutions of the linear system (15). Employing the standard adjoint equation technique of optimal control theory to eliminate $\mathbf{q}a_j$ we obtain

$$\frac{\partial \tilde{F}_i(\mathbf{a})}{\partial a_j} = \frac{\partial F_i(\mathbf{a}, \mathbf{q})}{\partial a_j} + (\mathbf{p}^i(\mathbf{a}))^{\mathrm{T}} \left(\frac{\partial \mathbf{f}(\mathbf{a})}{\partial a_j} - \frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_j} \mathbf{q}(\mathbf{a}) \right), \tag{16}$$

where $\mathbf{p}^i(\mathbf{a})$, $i = 0, \ldots, N_c$, are the solutions of the adjoint equations

$$\mathbf{K}(\mathbf{a})\, \mathbf{p}^i(\mathbf{a}) = \nabla_{\mathbf{q}} F_i. \tag{17}$$

Now the computation of $\nabla_{\mathbf{a}} \tilde{F}_i$, $i = 0, \ldots, N_c$, requires only $N_c + 1$ solutions of the linear system (17).

In the material derivative method, the sensitivity information is expressed as integrals of the solution of the continuous state and adjoint state. For details see [11], [21] and [24], for example. The formulae for directional derivatives in the continuous problem contain only boundary integrals. Applying this approach in the discrete case one may change the topology of the finite element mesh during optimization iterations.

The straightforward application of this method, however, may lead to severe inaccuracies in the numerical computations of the sensitivity information as was shown in [16]. This is due to the fact that numerical approximation of the optimality conditions for the continuous problem does not usually give good approximation to the optimality conditions of the approximate design problem. Boundary formulae usually contain terms like boudary fluxes or stresses which are difficult to calculate accurately by using the standard finite element method. One may try to avoid this difficulty by using mixed or hybrid finite elements to improve the accuracy of the above mentioned boundary terms. This complicates the analysis phase, however.

Finite differences

The simplest way from the programmers point of view to obtain partial derivatives of the functions is to use finite differences. For example

$$\frac{\partial \tilde{F}_0(\mathbf{a})}{\partial a_j} = \frac{\tilde{F}_0(\mathbf{a} + \delta \mathbf{e}^{(j)}) - \tilde{F}_0(\mathbf{a})}{\delta} + \mathcal{O}(\delta), \tag{18}$$

where $\mathbf{e}^{(j)}$ is the unit coordinate vector. It is clear that this approach has many drawbacks. The accuracy of the derivatives depends strongly on the right choice of the parameter δ. Instead of one full analysis and $N_c + 1$ adjoint analyses one has to perform $M + 1$ full analyses. Thus there is a substantial increase in computational costs. One can use higher order difference formulae to get better accuracy but the computational burden is increased. However, the method is general and there are situations where it is the only possible one to get the sensitivities.

Other methods

The so called *semi-analytical* formulae are widely used by the engineering community. They can be considered as a compromise between finite difference and algebraic differentation. In this case the sensitivity of $\mathbf{q}(\mathbf{a})$ is calculated but the derivatives on the right hand side of (16) are approximated by finite differences.

One may also apply the material derivative method directly to the *discrete* problem. The directional derivatives then contain area integrals that can be evaluated exactly. The formulae obtained are exact as in algebraic differentation. The method itself, however, does not give any advantage compared to algebraic differentation.

3.3. Question of parallelization

Unfortunately, the nonlinear mathematical programming using a gradient type method is a recursive process. Namely, each approximation $\mathbf{a}^{(k)}$ depends on the previous ones. Therefore the parallelization of the program code cannot occur at a very high level of granularity.

Calculation of the cost and constraint functions and their gradients at a given point $\mathbf{a}$ provide an opportunity for parallellization. Consider the following algorithm that computes $\tilde{F}_i(\mathbf{a})$ and $\mathbf{g}^i \equiv \nabla \tilde{F}_i(\mathbf{a})$, $i = 1, \ldots, N_c$, using $N_c + 1$ processors:

```
1. Compute $\mathbf{K}(\mathbf{a})$ and $\mathbf{f}(\mathbf{a})$
2. Factorize $\mathbf{L}\mathbf{L}^{\mathrm{T}} \leftarrow \mathbf{K}(\mathbf{a})$
3. Solve $\mathbf{y} \leftarrow \mathbf{L}^{-1}\mathbf{f}(\mathbf{a}), \quad \mathbf{q}(\mathbf{a}) \leftarrow \mathbf{L}^{-\mathrm{T}}\mathbf{y}$
4. parallel do $i = 0, \ldots, N_c$
5.     Set $\mathbf{g}^i \leftarrow 0$
6.     Compute $F_i(\mathbf{a}, \mathbf{q}(\mathbf{a}))$, $\mathbf{r}^i \leftarrow \nabla_{\mathbf{q}} F_i(\mathbf{a}, \mathbf{q}(\mathbf{a}))$
7.     Solve $\mathbf{y} \leftarrow \mathbf{L}^{-1}\mathbf{r}^i, \quad \mathbf{p}^i \leftarrow \mathbf{L}^{-\mathrm{T}}\mathbf{y}$
8.     do $j = 1, \ldots, M$
9.         Compute $\frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_j}$ and $\frac{\partial \mathbf{f}(\mathbf{a})}{\partial a_j}$
10.        Compute $g_j^i \leftarrow g_j^i + \frac{\partial F_i(\mathbf{a},\mathbf{q})}{\partial a_j} + (\mathbf{p}^i)^{\mathrm{T}} \left( \frac{\partial \mathbf{f}(a)}{\partial a_j} - \frac{\partial \mathbf{K}(\mathbf{a})}{\partial a_j} \mathbf{q}(\mathbf{a}) \right)$
11.    end do
12. end parallel do
```

In the level of parallelism indicated in the above algorithm the steps 1–3 are sequential operations. As step 2 is usually the most computationally expensive step the speedup achieved is modest.

As each step of the previous algorithm consists of basic linear algebraic operations there is a lot of lower level parallelism. This parallelism is exploited in the best way by using a computer with one or more vector processors. Thus the question of parallelization of shape optimization is reduced to the parallelization of the finite element method.

4. ALGEBRAIC SENSITIVITY ANALYSIS FOR THE POISSON EQUATION

In this Section, we develop algebraic sensitivity analysis in matrix form for the Poisson equation. We assume that the continuous problem is discretized using isoparametric Lagrangian elements. A sensitivity analysis of this type for linear elasticity problems has been done by Brockman [7].

Consider the following Poisson problem

$$\begin{cases} -\nabla \cdot (\rho(x)\nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{19}$$

Here $\Omega \subset \mathrm{R}^n$, $n = 2, 3$ is a sufficiently smooth bounded domain, $f \in L^2(\Omega)$ and $\rho : \mathrm{R}^n \to \mathrm{R}$ is a given smooth function. We assume that for given data the problem (19) is an elliptic problem and has an unique (weak) solution.

We discretize the problem (19) using Lagrangian finite elements of order m. Then the discrete analoque of problem (19) reads as

$$u_h \in V_h : \quad \int_{\Omega_h} \rho(x)\nabla u_h \cdot \nabla v_h \, dx = \int_{\Omega_h} f v_h \, dx \quad \forall\, v_h \in V_h, \tag{20}$$

where $V_h = \{\varphi \in C^0(\Omega_h) \mid \varphi|_{T_e} \in P^m(T_e),\ \varphi|_{\partial\Omega_h} = 0\}$ is the piecewise polynomial finite element space and $\Omega_h = \cup T_e$ is the finite element mesh. The matrix form of problem (20) is the system of linear algebraic equations

$$\mathbf{K}\,\mathbf{q} = \mathbf{f}. \tag{21}$$

The unknown vector $\mathbf{q}$ contains the nodal values of u_h.

Our aim is to find the sensitivities of $\mathbf{K}$ and $\mathbf{f}$ with respect to parameters a_j, $j = 1, \dots, M$, i.e. to find $\frac{\partial \mathbf{f}}{\partial a_j}$ and $\frac{\partial \mathbf{K}}{\partial a_j}$. In what follows we denote $(\cdot)' = \partial(\cdot)/\partial a_j$. The terms $\mathbf{K}'$ and $\mathbf{f}'$ can be computed element by element using the relations

$$\mathbf{K} = \sum_e \mathbf{P}^e \mathbf{K}^e \quad \text{and} \quad \mathbf{f} = \sum_e \mathbf{P}^e \mathbf{f}^e. \tag{22}$$

Here $\mathbf{P}^e$ is the "local-to-global" expanding matrix.

In the case of isoparametric elements, each element T_e is obtained from the parent element $\hat{T}$ ($[-1, 1]^n$, for example) by the mapping $\xi \in \hat{T} \mapsto x(\xi) \in T$. Let

$$\mathbf{N} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_m \end{pmatrix} \quad \text{and} \quad \mathbf{L} = \begin{pmatrix} \partial\varphi_1/\partial\xi_1 & \cdots & \partial\varphi_m/\partial\xi_1 \\ \vdots & \ddots & \vdots \\ \partial\varphi_1/\partial\xi_n & \cdots & \partial\varphi_m/\partial\xi_n \end{pmatrix} \tag{23}$$

be the matrices containing the values of the shape functions and their derivatives for the parent element. Denote by $\mathbf{J} = \left[\frac{\partial x_j}{\partial \xi_i}\right]_{i,j=1}^{n}$ the Jacobian of the mapping $\xi \mapsto x(\xi)$. Finally let

$$\mathbf{X}^e = \begin{pmatrix} X_1^1 & \cdots & X_n^1 \\ \vdots & \ddots & \vdots \\ X_1^m & \cdots & X_n^m \end{pmatrix} \tag{24}$$

be the matrix containing the nodal coordinates of the e:th element. (In what follows, we omit the superscript e as we are now working with the e:th element). At a point $x(\xi)$ the Cartesian derivatives of the shape functions are now given by $\mathbf{B} = \mathbf{J}^{-1}\mathbf{L}$ and the Jacobian by $\mathbf{J} = \mathbf{L}\mathbf{X}$.

The Gaussian quadrature with integration points and weights (ξ^k, W_k), $k = 1, \ldots, K$, see [22], is then used to perform the numerical integration needed for computing the element stiffness matrix and force vector, resulting in

$$\mathbf{K}^e = \sum_{k=1}^{K} W_k\, \rho(x^k)\, \mathbf{B}_k^{\mathrm{T}} \mathbf{B}_k |\mathbf{J}_k|, \tag{25}$$

$$\mathbf{f}^e = \sum_{k=1}^{K} W_k f(x^k) \mathbf{N}_k |\mathbf{J}_k|, \tag{26}$$

where $x^k = x(\xi^k)$, $\mathbf{B}_k = \mathbf{B}(\xi^k)$, $\mathbf{J}_k = \mathbf{J}(\xi^k)$ and $|\mathbf{J}_k| = \det \mathbf{J}_k$.

Lemma 1. The sensitivity of the matrix $\mathbf{B}_k$ is given by

$$\mathbf{B}_k' = -\mathbf{B}_k \mathbf{X}' \mathbf{B}_k. \tag{27}$$

Proof. As $\mathbf{B}_k = \mathbf{J}_k^{-1}\mathbf{L}_k$, we have

$$(\mathbf{J}_k \mathbf{B}_k)' = \mathbf{J}_k' \mathbf{B}_k + \mathbf{J}_k \mathbf{B}_k' = \mathbf{L}_k' = 0,$$

and therefore

$$\mathbf{B}_k' = -\mathbf{J}_k^{-1} \mathbf{J}_k' \mathbf{B}_k = -\mathbf{J}_k^{-1} \mathbf{L}_k \mathbf{X}' \mathbf{B}_k = -\mathbf{B}_k \mathbf{X}' \mathbf{B}_k. \qquad \square$$

Lemma 2. The sensitivities of x^k, $\rho(x^k)$ and $|\mathbf{J}_k|$ are given by

$$(x^k)' = (\mathbf{X}')^{\mathrm{T}} \mathbf{N}_k, \tag{28}$$

$$(\rho(x^k))' = (\nabla_x \rho(x^k))^{\mathrm{T}} (\mathbf{X}')^{\mathrm{T}} \mathbf{N}_k. \tag{29}$$

$$|\mathbf{J}_k|' = |\mathbf{J}_k| \sum_{j=1}^{m} (\nabla \varphi_j(x^k))^{\mathrm{T}} (X^j)'. \tag{30}$$

Proof. The result immediately follows from the relation

$$x^k = \mathbf{X}^{\mathrm{T}} \mathbf{N}_k$$

and the definition of the determinant. $\square$

Combining the results of Lemmas 1 – 2 we have:

Theorem 1. The sensitivities of $\mathbf{K}^e$ and $\mathbf{f}^e$ are given by

$$(\mathbf{K}^e)' = \sum_{k=1}^{K} \left(D_k (\mathbf{B}_k')^{\mathrm{T}} \mathbf{B}_k + D_k \mathbf{B}_k^{\mathrm{T}} \mathbf{B}_k' + E_k \mathbf{B}_k^{\mathrm{T}} \mathbf{B}_k + F_k \mathbf{B}_k^{\mathrm{T}} \mathbf{B}_k \right), \tag{31}$$

$$(\mathbf{f}^e)' = \sum_{k=1}^{K} W_k \left(\nabla_x f(x^k)^{\mathrm{T}} (x^k)' \mathbf{N}_k |Jk| + f(x^k) \mathbf{N}_k |k|' \right), \tag{32}$$

where

$$\begin{aligned} D_k &= W_k |\mathbf{J}_k| \rho(x^k), \\ E_k &= W_k |\mathbf{J}_k| (\nabla_x \rho(x^k))^{\mathrm{T}} (\mathbf{X}')^{\mathrm{T}} \mathbf{N}_k, \\ F_k &= W_k |\mathbf{J}_k|' \rho(x^k). \end{aligned}$$

Remarks. In the equations (27)–(32) the only matrix depending on a specific application (mesh topology, design parametrization, etc.) is $\mathbf{X}'$. All other matrices are available from the assembly of the system (21).

The previous analysis can be done in the same way for other state problems and nonlinear problems too. For further details see [8], [14].

A simple example. Consider the following geometry. The domain $\Omega(\mathbf{a})$ is the rectangle $(0, a_1) \times (0, a_2)$. The domain is divided into elements as shown in Figure 2. Let us take the element number 5. The matrix $\mathbf{X}$ of nodal coordinates corresponding to this element is

$$\mathbf{X} = \begin{pmatrix} a_1/3 & a_2/3 \\ 2a_1/3 & a_2/3 \\ 2a_1/3 & 2a_2/3 \\ a_1/3 & 2a_2/3 \end{pmatrix}.$$

Then the only matrices needed for the calculation for the sensitivity of $\mathbf{q}(\mathbf{a})$ are

$$\frac{\partial \mathbf{X}}{\partial a_1} = \begin{pmatrix} 1/3 & 0 \\ 2/3 & 0 \\ 2/3 & 0 \\ 1/3 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial \mathbf{X}}{\partial a_2} = \begin{pmatrix} 0 & 1/3 \\ 0 & 1/3 \\ 0 & 2/3 \\ 0 & 2/3 \end{pmatrix}.$$

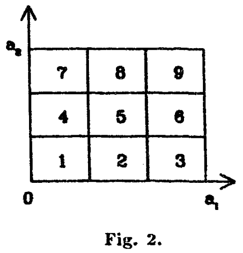

Fig. 2.

5. SHAPE REPRESENTATION

5.1. Node movement technique

The classical technique is to use as design variables the coordinates of the boundary nodes of the FE-mesh [3], [23]. This type of parametrization allows nonsmooth boundaries even for coarse meshes. One can also obtain conforming finite element approximation as the parametrized boundary is exactly followed by the finite elements. Unfortunately, a fine finite element mesh implies a huge number of design variables. An annoying property of optimal approximate design is that it very often oscillates for a "large" h. Many authors have introduced heuristic methods to prevent these oscillations, see [18] for example, or restricted the design space [12]. There are three main reasons why such oscillations can exist: the solution of the continuous design problem is oscillating, the continuous design problem has no solution but the approximate ones have for each h, the rate of convergence of the approximate designs is slow. An ideal situation arises if the designer is able to solve the design problem with different h and by comparing the results to conclude if the sequence of designs is converging before restricting the design space further.

5.2. Design elements

Rather than trying to express the boundary using only one curve the so called *design element* technique is often used [5], [6]. The domain to be optimized is divided into several patches which are to be meshed separately.

A popular way in the engineering community is to use *design elements*, one or more sides of which are Bezier or B-spline curve segments. In this case design variables are the coordinates of the "keypoints" which are not necessarily boundary points. The reason for the popularity of spline curves is obvious: the boundary is always smooth, the number of design variables moderate and spline-curves (surfaces) are basic tools in a CAD-environment. Moreover, C^1-continuity of the curve segments across design element boundaries can be achieved using simple geometrical rules.

Let us consider the construction of a simple design element. Let $P^1, P^2, \dots, P^{M+2}$ be the keypoints of the design element shown in Figure 3. One side of the design element is curved while the other ones are straight lines. The curved side is given by a Bezier-curve

$$\beta(s) = \sum_{k=1}^{M} P^k B_k^M(s). \tag{33}$$

defined by the keypoints $P^1, P^2, \dots, P^M$ and the blending functions B_k^M. Let $0 = s_1 < s_2 < \dots < s_n = 1$ be a uniform partition of $[0,1]$ and let $L^i = \beta(s_i)$, $U^i = (1-s_i)P^{M+2} + s_i P^{M+1}$; $i = 1, \dots, n$.

Each keypoint is required to lie on a segment of a line, i. e.

$$P^k = (1-a_k)P_L^k + a_k P_U^k, \quad 0 \le a_k \le 1;\ k = 1, \dots, M+2. \tag{34}$$

The design variables are now the numbers a_k. Let $0 = \sigma_1 < \sigma_2 < \dots < \sigma_m = 1$ be another uniform partition of $[0,1]$. Now a general gridpoint x_{ij} in the non-Cartesian

$n \times m$ grid shown in Figure 4 can be given by

$$x_{ij} = (1 - \sigma_j)L^i + \sigma_j U^i, \quad i = 1, \ldots, n; \; j = 1, \ldots, m. \tag{35}$$

It is important to notice that the dependence of the gridpoints on the design variables is *linear*. Thus the construction of the matrix $\mathbf{X}'$ needed in the sensitivity analysis in the previous section is easy to perform.

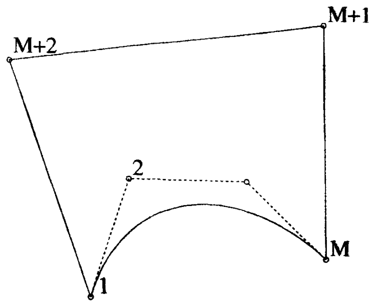

Fig. 3. Design element geometry

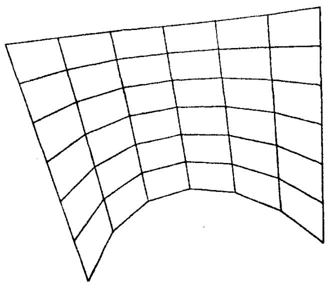

Fig. 4. Meshed design element

5.3. A typical example

Consider the shape optimization of a hole in a cantilever. The volume of the cantilever is fixed and we want to find the shape of a hole in the beam which minimizes the compliance. The geometry is described by four Bezier-design elements discussed above. The shape of the hole is subjected to several geometrical constraints, the specification of which we omit for brevity. We optimized the shape using the sequential quadratic programming algorithm from the NAG-library ([15]). The gradient of the cost function was calculated by using algebraic sensitivity analysis formulae. The initial and final shapes are shown in Figures 5,6. The reduction in compliance was about 23%.

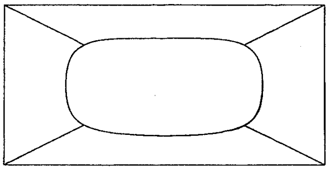
Fig. 5. Initial design

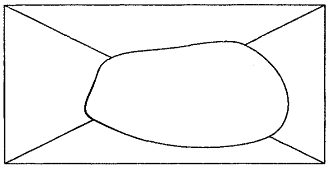
Fig. 6. Final design

6. CONCLUSIONS

Shape optimization of systems governed by linear state problems has reached a level of maturity that has made it possible to implement the methods in CAE (Computer Aided Engineering) systems for production use. A general sensitivity analysis capability built into a FEM system and its CAD type preprocessor is the key to a fully computerized design cycle. Programs like CAOS ([19]), OASIS ([17]) and SAMCEF ([20], [13]) can be considered "complete" optimal design systems. Moreover, some commercially available large FEM systems provide optimization or sensitivity analysis modules for optimal sizing. In practice, a FEM system having shape optimization capability should be distributed over various types of computers: a powerful mainframe for finite element analyses and design sensitivity analyses (batch jobs), and a graphical workstation for pre- and post-processing.

Despite the amount of existing software a great deal of research work is still to be done. Design problems governed by nonlinear state problems (contact problems, nonlinear materials, Navier–Stokes, etc.) are still seldomly solved due to the need for an enormous computing capacity. Moreover, it should be noted that the user of the above mentioned systems is still responsible for setting up the optimization problem correctly. As a designer of practical systems is seldom an expert in optimization or numerical analysis, an expert system should be available to guide the user in formulating the problem.

(Received May 7, 1992.)

REFERENCES

[1] J. Arora: Introduction to Optimum Design. McGraw-Hill, New York 1989.

[2] N. Banichuk: Problems and Methods of Optimal Structural Design. Plenum Press, London 1983.

[3] D. Begis and R. Glowinski: Application de la méthode des éléments finis à l'approximation d'un problème de domaine optimal. Appl. Math. Optim. *2* (1975), 130–169.

[4] M. P. Bendsøe and N. Kikuchi: Generating optimal topologies in structural design using a homogenization method. Comput. Methods Appl. Mech. Engrg. *71* (1988), 197–224.
[5] V. Braibant and C. Fleury: Shape optimal design using B-splines. Comput. Meths. Appl. Mech. Engrg. *44* (1984), 247–267.
[6] V. Braibant and C. Fleury: An approximation-concepts approach to shape optimal design. Comput. Meths. Appl. Mech. Engrg. *53* (1985), 119–148.
[7] R. A. Brockman: Geometric sensitivity analysis with isoparametric finite elements. Comm. Appl. Numer. Methods *3* (1987), 495–499.
[8] J. Haslinger and R. A. E. Mäkinen: Shape optimization of elasto-plastic bodies under plane strains: sensitivity analysis and numerical implementation. Struct. Optim. *4* (1992), 133–141.
[9] J. Haslinger and P. Neittaanmäki: Finite Element Approximation for Optimal Shape Design. J. Wiley, Chichester 1988.
[10] E. J. Haug and J. Arora: Applied Optimal Design. J. Wiley, New York 1979.
[11] E. J. Haug, K. Choi and V. Komkov: Design Sensitivity Analysis of Structural Systems. Academic Press, Orlando 1986.
[12] I. Hlaváček and R. Mäkinen: *On the numerical solution of axisymmetric domain* optimization problems. Appl. Math. *36* (1991), 284–304.
[13] D. Liefooghe and C. Fleury: An interactive capability for shape optimal design. Finite Elements in Analysis and Design *5* (1989), 39–55.
[14] R. Mäkinen: Finite-element design sensitivity analysis for non-linear potential problems. Commun. Appl. Numer. Methods *6* (1990), 343–350.
[15] NAG Fortran library, Numerical Algorithms Group Ltd, Oxford.
[16] P. Neittaanmäki and T. Tiihonen: Sensitivity analysis for a class of optimal shape design problems. University of Jyväskylä, Dept. of Math., REPORT 29, 1985.
[17] OASIS-ALADDIN optimization system, ALFGAM optimering AB, Stockholm.
[18] O. Pironneau: Optimal Shape Design for Elliptic Systems. Springer-Verlag, New York 1984.
[19] J. Rasmussen: The structural optimization system CAOS Struct. Optim. *2* (1990), 109–115.
[20] SAMCEF, Système d'Analyse des Milieux Continus par Elements Finis, Aerospace Laboratory, University of Liège, Liège.
[21] J. Sokolowski and J.-P. Zolésio: Shape sensitivity analysis of unilateral problems. SIAM J. Math. Anal. *18* (1987), 1416–1437.
[22] O. C. Zienkiewicz: The Finite Element Method. McGraw-Hill, London 1977.
[23] O. C. Zienkiewicz and J. S. Campbell: Shape optimization and sequential linear programming. In: Optimum Structural Design (R. Gallagher and O. C. Zienkiewicz, eds.), J. Wiley, New York 1973, pp. 109–126.
[24] J.-P. Zolésio: Identification de domaines par déformations. Thése d'ètat, Université de Nice, 1979.

Dr. Raino A. E. Mäkinen, Department of Mathematics, University of Jyväskylä, P. O. Box 35, SF-40351 Jyväskylä. Finland.