Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXII, págs. 2637-2654 (artículo completo)
Carlos G. García Garino, Aníbal E. Mirasso, Mario A. Storti, Miguel E. Tornello (Eds.)
Mendoza, Argentina, 19-22 Noviembre 2013

SHAPE OPTIMIZATION OF AXISYMMETRIC SOLIDS USING AUTOMATIC DIFFERENTIATION AND NURBS PARAMETRIZATION

Linn, Renato Vaz[a], Espath, Luis Felipe da Rosa[b] and Awruch, Armando Miguel[c]

[a]*Graduate Student PPGEC/UFRGS renatolinn@gmail.com*

[b]*Graduate Student PPGEC/UFRGS espath@gmail.com*

[c]*Graduate Program in Civil Engineering (PPGEC), Federal University of Rio Grande do Sul (UFRGS), Av. Osvaldo Aranha, 99, 3o andar, 90035-190, Porto Alegre, RS, Brazil, amawruch@ufrgs.br*

Keywords: Shape Optimization, NURBS, Automatic Differentiation (AD), Axisymmetric Solids.

Abstract. Shape optimization of axisymmetric solids is performed in this work using a NURBS parametrization of the geometry coupled with a Sequential Quadratic Programming (SQP) algorithm where the sensitivity analysis is performed by automatic differentiation (AD). The structural analysis is evaluated by the Finite Element Method. The framework proposed is computationaly efficient and accurate and in addition remeshing techniques are also avoided. Large thickness variation are allowed with the methodology. As a result, structures with improved structural performance are obtained for linear elastic structural problems.

1 INTRODUCTION

In structural mechanics, axisymmetric structures have a wide range of applications such as components of submarines and aircraft fuselages, rockets, metallic silos, pressure vessels, spherical domes, cooling towers, among others. As the stiffness of these structures depend on their geometry and material properties, shape optimization is necessary in order to obtain improvement in their behaviour for fixed materials. Axisymmetric shape, loads and boundary conditions of these structures also lead to two-dimensional simplification in the analysis of the problem. Thus, shape optimization of the three-dimensional solid geometry can be performed through shape optimization of the cross section of the structure. Shape optimization deals with the modification of the structure using an optimization algorithm by the structural analysis and its sensitivity analysis. Therefore, different areas must be coupled together to establish a structural optimization system. Both the accuracy and efficiency of the optimization depends on all these areas.

Many different structural models and approaches have been focused in structural optimization. Some works of Ramm et al. (1993) and Bletzinger et al. (2008) use a thick and thin shell element formulation, respectively, with an elastic formulation. Shape optimization of shells to prevent buckling are analyzed by Khosravi et al. (2008) and Aubert and Rousselet (1998), and the same problem, taking into account imperfections of the structure, were studied by Reitinger and Ramm (1995). Axisymmetric thin shell structures optimization was investigated by Mota Soares et al. (1994) and a shell optimization by Rousselet et al. (1995). Shape optimization of axisymmetric structures was analyzed by Özakça et al. (1993) using an adaptative finite element procedure and by Csonka and Kozák (1995) using a higher-order shear deformation theory. AD with NURBS was investigated by the authors of this work (Espath et al., 2011) to optimize shape of shell structures, obtaining a good performance. In the present work, AD, SQP and NURBS are also used for shape optimization. However, the major limitation addressed to shape optimization of shells is the difficult to deal with thin and thick regions in the same structure. Since axisymmetric solids naturally deal with these strong differences in thickness, in the optimization process a large thickness variation is allowed. Thus, in the optimal shape is expected solids with three different behaviors related to thin, thick shells as well as solids structures.

For optimization problems, evaluation of derivatives are of great importance since it is one of the most important information used by the gradient-based optimization algorithm to find the optimum point. The need for obtaining derivatives in optimization problems is directly related to its formulation. Therefore, the choose of the differentiation method should combine both precision and efficiency to be well suitable for optimization procedures. The Automatic Differentiation method (AD) is a numerical-computational method for evaluating derivatives with the advantage of having analytical precision, limited only by the truncation error of the machine. Evaluation of gradients of functionals by AD have generally a much lower computational cost compared to other numerical methods (such as the finite differences method), being well suitable for optimization applications, where those evaluations take massive computation efforts. However, additional computational memory can be needed when using AD.

The optimization algorithm used in the present work is the Sequential Quadratic Programming (SQP). It is a robust algorithm for deterministic non-linear optimization with continuous variables being well suitable for the current shape optimization purpose, where the geometrical constraints adopted are usually highly nonlinear. Also, as the SQP algorithm uses a quasi-Newton approach, only the gradient information is needed instead of a complete Hessian matrix

information required by a Newton approach. This feature is well suitable for the coupled use of AD, with faster and more accurate evaluation of derivatives.

The characteristic variables of shape optimization process are geometrical parameters that define the geometry of the structure. The number of variables can be dramatically reduced if Computer Aided Geometry Design (CAGD) concepts are used. According to these concepts, free-form geometries can be described through a set of few points called control points. These control points are used as optimization parameters in the present work. Thus, a smooth and precise geometry description is looked for in the shape optimization context. A Non Uniform Rational B-Splines (NURBS) parametrization promotes an easy shape modification through the manipulation of the control points. Furthermore, it can precisely represent complex geometries with an efficient mathematical implementation. Such description is standard to describe and to model curves and surfaces in CAGD. A two-dimensional NURBS description is used for shape description and modification in the present work. More details of NURBS can be found in Piegl and Tiller (1997).

The structural analysis is performed using a standard Constant Stress Triangle (CST) finite element for static elastic linear axissymetric structural problems. Avoiding large mesh distortion and entanglement during the geometry modification, a mesh update scheme is used.

2 THEORETICAL ASPECTS

Many fields have to be coupled together in order to create an optimization system: geometry description, sensitivity analysis, optimization algorithm and structural analysis. The optimization of the geometry is performed by modification of the control points from a NURBS parameterization of the structure. The sensitivity analysis is done using the Automatic Differentiation (AD) method. The optimization algorithm used here is the Sequential Quadratic Programming (SQP) and the structural analysis is performed using a CST finite element for axisymmetric problems.

2.1 Automatic Differentiation (AD)

The first-order sensitivity analysis is performed using Automatic Differentiation (AD) in the present work. This method is based on the graph theory. Computationally, all evaluations can be described as a trace (sequence) of calculus. This trace contains all the sequence of elemental evaluations performed by the computer in order to achieve the final result (the functional evaluation, for example). This means that to calculate a given functional $\mathcal{F}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$, it is possible to rewrite it by breaking the original functional into a functional evaluated by parts:

$$\mathcal{F}(\mathbf{x}) \equiv \bigodot_{i=1}^{l} v_i = v_1 \circ \ldots \circ v_l \tag{1}$$

where these v_i parts are intermediate calculations, each of them evaluated by means of one or more previous intermediate calculations, making a sequence of operations to evaluate the functional. We also call each of these v_i steps of evaluations as variables. The initial steps of evaluation are called initial variables, with i ranging from $1-n$ to 0. The initial variables are also the independent variables, $\mathbf{x}$. The internal steps are called internal variables, with i ranging from 1 to $l-1$ and the final variable is the functional itself, with i defined as l. We can group all the variables v_i in a set $\mathbf{V}$:

$$\mathbf{V} = \{\underbrace{v_{1-n}, \ldots, v_0}_{\mathbf{x}}, v_1, v_2, \ldots, v_{l-1}, \underbrace{v_l}_{\mathcal{F}(\mathbf{x})}\} \tag{2}$$

Each of these v_i variables are the vertices of a graph (Figure 1). The vertices v_i with $i \geq 1$ are obtained by applying an elementar function φ_i to a given set or arguments v_j, with $i > j$. This can be stated as:

$$v_i = \varphi_i(v_j)_{j \prec i} \tag{3}$$

where the symbol $j \prec i$ means that a given variable v_i is directly depended on a set of given variables v_j, with $i > j$. An elementar function φ_i is any function associating one or two variables v_j to a new variable v_i, for example, the sum, the division and the multiplication of two variables and the exponential and the secant of one variable. It is obvious that any complicated functional can be expressed as a sequence of elementary operations over the independent variables. More generally this means that a functional evaluation can be interpreted as a graph $\mathcal{G}$ containing vertices $\mathbf{V}$ and edges $\mathbf{E}$. The vertices are the variables v_i whose relations are the edges, given by $\varphi_i(v_j)_{j \prec i}$, where $i > j$. In a more systemic notation it can be stated as (see Figure 1):

$$\mathcal{G}\left(\mathbf{V}, \mathbf{E}\right) \quad \begin{matrix} \mathbb{R}^n \to \mathbb{R}^1 \\ \mathbf{x} \to \mathcal{F}(\mathbf{x}) \end{matrix} \quad v_i = \begin{cases} x_i & i = 1-n \ldots 0 \\ \varphi_i(v_j)_{j \prec i} & i = 1 \ldots l-1 \\ \mathcal{F}(\mathbf{x}) & i = l \end{cases} \tag{4}$$

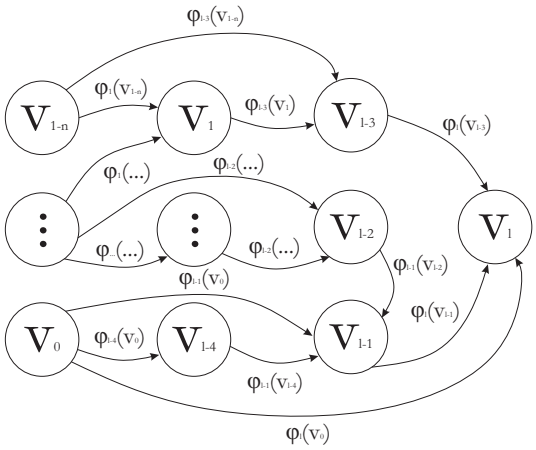

Figure 1: An example of graph variables and its vertices.

Once a graph (and also an entire algorithm which evaluates a given functional) is stated as above, it can be derived by two distinct ways. The first methodology is simply the application of the chain rule with respect to a given variable t:

$$\frac{\partial \mathcal{F}(\mathbf{x})}{\partial t} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} \tag{5}$$

where the term $\frac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian matrix of $\mathcal{F}(\mathbf{x})$. This is called forward mode or tangent mode of differentiation in AD (Griewank and Walther, 2008). The forward mode implies that for each variable of the graph also its corresponding derivative with respect to t must be evaluated. This way, the total size of this process is exactly twice as the single evaluation of the functional. Applying the chain rule (5) on the graph (4) and taking into account (1) the following derivatives arise:

$$\frac{\partial v_i}{\partial t} = \begin{cases} \frac{\partial x_i}{\partial t} & i = 1-n \ldots 0 \\ \sum_{j \prec i} \frac{\partial \varphi_i(v_j)_{j \prec i}}{\partial v_j} \frac{\partial v_j}{\partial t} & i = 1 \ldots l-1 \\ \frac{\partial \mathcal{F}(\mathbf{x})}{\partial t} & i = l \end{cases} \tag{6}$$

The derivative of the elementary function $\frac{\partial \varphi_i}{\partial v_j}$ is also called tangent function associated to the elementar function φ_i and it is obtained by simple differentiation of the original function. For example, the elementar function associated with the multiplication of two variables $\varphi = a \cdot b$ has the tangent function $\frac{\partial \varphi}{\partial t} = \frac{\partial a}{\partial t} \cdot b + a \cdot \frac{\partial b}{\partial t}$ associated.

The second methodology to propagate derivatives is called reverse mode of propagation in AD. In this mode not only a directional derivative but the complete gradient of the functional is evaluated. It can be stated:

$$\nabla^t \mathcal{F}(\mathbf{x}) = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}} \tag{7}$$

The total size of this procedure is no more then twice the original functional evaluation because there is a dependence on the particular graph structure. To apply the gradient on the graph, the propagation of evaluations over the variables v_i must be performed in a reverse way, from the functional to the independent variables $\mathbf{x}$:

$$\frac{\partial \mathcal{F}(\mathbf{x})}{\partial v_i} = \begin{cases} 1 & i = l \\ \sum_{i \succ j} \frac{\partial \varphi_i(v_j)_{j \prec i}}{\partial v_j} \frac{\mathcal{F}(\mathbf{x})}{\partial v_i} & i = l-1 \ldots 1 \\ \nabla^t \mathcal{F}(\mathbf{x}) & i = 0 \ldots 1-n \end{cases} \tag{8}$$

In the statement above, the need of evaluation of the contribution of all the successors $j \succ i$ of a given variable v_i can be computationally inappropriate because of the need of information not directly available from the elementary functions φ_i and its arguments. This way, it it necessary to obtain for each variable v_j a list of all elementary functions depending of the variables v_j. This approach is also called non-incremental reverse mode. However, the most practical for computational applications is the incremental reverse mode, which is identically to the non-incremental mode except that the intermediate variables are evaluated as follows:

$$\frac{\partial \mathcal{F}(\mathbf{x})}{\partial v_i} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial v_i} + \frac{\partial \varphi_i(v_j)_{j \prec i}}{\partial v_j} \frac{\mathcal{F}(\mathbf{x})}{\partial v_i} \quad \forall j \prec i, \quad i = l-1 \ldots 1 \tag{9}$$

The computational complexity of a evaluation procedure can be measured in terms of the number of computational flops involved and it is directly related with the computational cost (physical time required by the machine to execute the evaluation). The complexity involved in forward mode is (Griewank, 1993):

$$\frac{flops\{\frac{\partial \mathcal{F}(\mathbf{x})}{\partial t}\}}{flops\{\mathcal{F}(\mathbf{x})\}} \leq 1 + 3n \tag{10}$$

where $flops\{\mathcal{F}(\mathbf{x})\}$ is the number of flops needed to evaluate the original functional $\mathcal{F}(\mathbf{x})$ and $flops\{\frac{\partial \mathcal{F}(\mathbf{x})}{\partial t}\}$ is the number of flops needed to evaluate the forward propagation of derivatives in AD. For the reverse mode the complexity is:

$$\frac{flops\{\nabla^t \mathcal{F}(\mathbf{x})\}}{flops\{\mathcal{F}(\mathbf{x})\}} \leq 5 \tag{11}$$

where $flops\{\nabla^T \mathcal{F}(\mathbf{x})\}$ is the number of flops needed to evaluate the gradient of a scalar functional $\mathcal{F}(\mathbf{x})$ in the reverse mode of AD. It is explicit in the statement above that it holds to be independent of any length of $\mathbf{x}$. This independence with respect to the number of independent variables is high efficient in optimization problems where the number of variables can be very large and the computational costs involved are very high. Although the reverse complexity

was stated in the present work only to scalar-valuable functionals, it can be also extended to vectorial-valuate functionals $\mathcal{F}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$:

$$\frac{flops\{\nabla^t \mathcal{F}(\mathbf{x})\}}{flops\{\mathcal{F}(\mathbf{x})\}} \leq 1 + 4m \tag{12}$$

being the scalar case $m = 1$ a particular case of the general complexity relation. It is evident that when the number of required first-order derivatives of the functional is larger compared with the total number of independent variables the reverse mode is preferable instead of the forward mode due to less computational cost involved.

Although the reverse mode is very computationally cheaper compared with other numerical procedures to evaluate derivatives, it has a drawback: the computational memory needed. This occurs due to the requirement to store all the computational graph in the memory and also its derivatives. Special treatment can be performed to optimize the memory usage in this process, with allocation and deallocation of variables and taking into account the Jacobian matrix sparsity (Griewank, 2003).

In the present work, automatic differentiation tool TAPENADE AD INRIA (2002), developed by the INRIA in France, is employed to evaluate the derivatives with reverse automatic differentiation.

2.2 Non-Uniform Rational B-*Spline* (NURBS)

In order to represent a complex surface, a parametric representation is used. NURBS parameterization is well suitable for shape optimization in any physical problem involving curves, surfaces and solids. For a bidimensional representation of structures, a plane surface representation in the form

$$\mathbf{S}(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta)) \qquad (\xi, \eta) \in [0, 1] \times [0, 1]$$

is looked for.

The NURBS surface in homogeneous coordinates is defined as

$$\mathbf{S}^w(\xi, \eta) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(\xi) N_{j,q}(\eta) \mathbf{P}_{i,j}^w \tag{13}$$

where (p, q), (n, m), $(N_{i,p}(\xi), N_{j,q}(\eta))$ are the degrees, the numbers of basis functions, and the basis functions in (ξ, η) directions, respectively. $\mathbf{P}_{i,j}^w = (w_{i,j}x_{i,j}, w_{i,j}y_{i,j}, w_{i,j})$ are the control points in homogeneous coordinates.

The basis functions in recursive form are defined as

$$N_{i,0}(\xi) = \begin{cases} 1 \text{ if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 \text{ otherwise} \end{cases} \tag{14}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{15}$$

$$\tag{16}$$

over the following knot vectors, in (ξ, η) directions

$$\Xi = \{\underbrace{0, \ldots, 0}_{p+1}, \xi_{p+1}, \ldots, \xi_{r-p-1}, \underbrace{1, \ldots, 1}_{p+1}\} \quad \text{with} \quad r = n + p + 1$$

$$\mathcal{H} = \{\underbrace{0, \ldots, 0}_{q+1}, \eta_{q+1}, \ldots, \eta_{s-q-1}, \underbrace{1, \ldots, 1}_{q+1}\} \quad \text{with} \quad s = m + q + 1$$

In the shape modification, when a control point, a weight and/or a knot position is modified the parametric mesh remains the same; however, the mesh in the Euclidean space assumes a new position according with the new geometry. This mapping may be not the best choice, because the mesh distortion is not controlled, although this problem is somehow minimized with a convenient initial mesh. An additional update scheme to control mesh distortion is also used in the present work and it is explained in the section 3.

The geometry description used in the present work uses a weight $w = 1$ which leads the general NURBS description to a particular Bézier description case. In the optimization process, the external control points (lying on the boundary) $\mathbf{P}_{i,j}$ of the geometry are used as optimization variables and the internal ones are defined as dependent of the external control points (this dependence is also the mesh distortion control scheme).

A complete overview of NURBS and Bézier theories can be found in Piegl and Tiller (1997).

2.3 Numerical optimization

In the context of the numerical shape optimization, the SQP method is used. The optimization problem is stated as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \mathcal{F}(\mathbf{x}) \qquad \text{subject to} \begin{cases} \mathcal{C}_i(\mathbf{x}) = 0, & i \in \mathcal{E} \\ \mathcal{C}_i(\mathbf{x}) \geqslant 0, & i \in \mathcal{I} \end{cases} \tag{17}$$

where $\mathcal{F}$, $\mathcal{C}_i$ are defined in $\mathbb{R}^n$; and $\mathcal{E}$ and $\mathcal{I}$ are two finite index sets. $\mathbf{x}$ are the independent variables, $\mathcal{F}$ is the objective function, $\mathcal{C}_i, i \in \mathcal{E}$ are equality constraints and $\mathcal{C}_i, i \in \mathcal{I}$ are inequality constraints.

The active set of inequality constraints is expressed as

$$\mathcal{A}(\mathbf{x}) = \mathcal{E} \cup \{i \in \mathcal{I} \mid \mathcal{C}_i(\mathbf{x}) = 0\} \tag{18}$$

The Lagrangian functional is defined as

$$\mathcal{L}(\mathbf{x}, \lambda) \equiv \mathcal{F}(\mathbf{x}) - \sum_{i\in\mathcal{A}(\mathbf{x})} \lambda_i \mathcal{C}_i(\mathbf{x}) \tag{19}$$

where λ_i are the Lagrange multipliers. If the linear independence constraint qualification holds, then the optimum $(\mathbf{x}^*, \lambda^*)$ must satisfy the *Karush-Kuhn-Tucker* (*KKT*) condition, i.e.,

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*) = \mathbf{0}, \tag{20a}$$
$$\mathcal{C}_i(\mathbf{x}^*) = 0, \quad \forall\, i \in \mathcal{E}, \tag{20b}$$
$$\mathcal{C}_i(\mathbf{x}^*) \geqslant 0, \quad \forall\, i \in \mathcal{I}, \tag{20c}$$
$$\lambda_i^* \geqslant 0, \quad \forall\, i \in \mathcal{I}, \tag{20d}$$
$$\lambda_i^* \mathcal{C}_i(\mathbf{x}^*) = 0, \quad \forall\, i \in \mathcal{E} \cup \mathcal{I} \tag{20e}$$

The proof and the complete theory related is shown by Nocedal and Wright (1999). The *KKT* conditions, Eqs. (20a) to (20e), must be satisfied for a given tolerance range.

The numerical approach for the optimization problem using a SQP algorithm starts from an initial point $\mathbf{x}_0$, and then the SQP algorithm generates a sequence of iterations $\{\mathbf{x}_k\}_{k=0}^{\infty}$ that ends when no more progress can be made or when this point is the approximate solution with sufficient accuracy, i.e., the KKT conditions are satisfied for a given tolerance range. The SQP

algorithm approximates the objective function $\mathcal{F}(\mathbf{x}_k)$ by a quadratic form and the restrictions $\mathcal{C}_i$ (of both equality and inequality) by a linear form. Therefore, applying the Newton method to the Lagrangian function with these approximations for the objective function and restrictions, the SQP method arrives to a quadratic programming problem. An algorithm for solving quadratic problems is then applied in order to obtain a search direction $\delta_k = \mathbf{x}_k - \mathbf{x}_{k-1}$, being δ_k the solution of the quadratic problem. A linear search method is then used to obtain a new position trough this search direction with lower value, which is a unidimensional minimization problem. While the optimal conditions are not satisfied, new points and new directions with new length are evaluated on each step and the process is repeated. Every new point $\mathbf{x}_{k+1}$ is updated by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k \tag{21}$$

While the optimum is not reached, the Hessian matrix required by the quadratic program is updated by the Broyden-Fletcher-Goldfarb-Shanno (*BFGS*) method. The BFGS method is an algorithm which generates an approximation for the Hessian based on the gradients of the current and previous iteration. This mean that BFGS is a quasi-Newton method, i.e., the Hessian is not evaluated exactly, but approximated by gradients evaluations. These gradients evaluations are performed using AD.

2.4 Structural analysis

The structural analysis is performed using the finite element method for an axisymmetric problem. The Constant Stress Triangle (CST) for linear elastic static analysis, commonly related in the literature, is employed.

3 SHAPE MODIFICATION AND SHAPE OPTIMIZATION

The coupling of the optimization procedure to shape optimization is presented in this section. The shape modification is performed by moving control points $\mathbf{P}_{i,j}$ belonging to the NURBS description of the geometry. Only the control points on the boundary Γ need to be chosen as optimization variables and the internal control points within the domain Ω are updated in order to prevent interpenetration of the control points and also to avoid large mesh distortions. Thus:

$$\mathbf{x} = \{\mathbf{P}_{i,j} \in \Gamma\} \tag{22}$$

The boundary control points which are optimization variables are allowed to move in just one given direction. An upper and lower bound are imposed over this control point movement as an optimization constraint. Also, a minimal distance is imposed as optimization constraint to avoid interpenetration of the control points. When the new position of these optimization variables are obtained by the optimization algorithm, intermediate control points are then updated by linearly interpolation over the distance between the boundary control points along the moving direction. This process improves the mesh update and is represented in Figure 2 and Figure 3. In these figures, the red control points represent control points which are chosen as optimization variables and the blue ones the other control points. Figure 2 represents the initial condition of the mesh and geometry. Thus, the two middle optimization nodes (red) on the top and bottom boundaries are moved in one direction. As a result, intermediate points (blue) between points representing the variables (red) are also updated, holding an equal distance along the moved direction. In Figure 3 it can be seen that this procedure moves the mesh following the change of the geometry. The chosen objective function to be minimized $\mathcal{F}$ is the internal strain energy

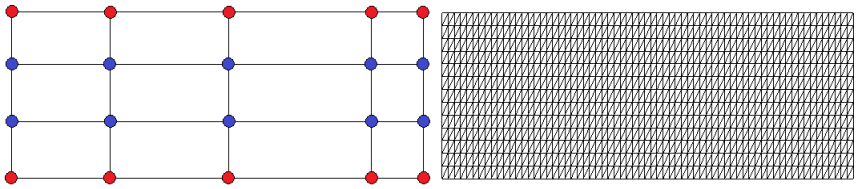

Figure 2: Original geometry and mesh.

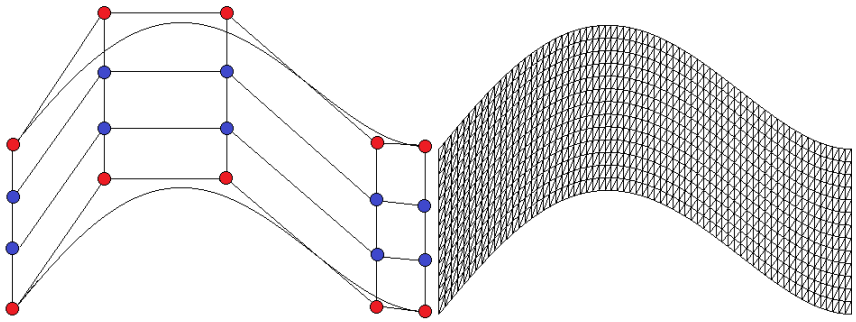

Figure 3: New geometry and mesh.

in linear elastic problems, which is given by

$$\mathcal{F} = \int_{\Omega} \underline{\underline{\sigma}} : \underline{\underline{\epsilon}} \, \mathrm{d}\Omega = U^T K U \tag{23}$$

where $\underline{\underline{\sigma}}$ is the stress tensor, $\underline{\underline{\epsilon}}$ is the strain tensor, U is the displacement vector and K is the stiffness matrix. The optimization problem is performed using relative objective functions values, i.e., referred to an initial value.

This minimization of $\mathcal{F}$ is then constrained by geometrical or mechanical functions, such as constant volume restriction, limiting upper and lower bound of the points and the minimum distance between boundary control points in the given optimized direction. These restrictions may be written as follows:

$$\begin{aligned} \mathcal{C} &= 1 - \frac{V}{V_{ini}} = 0 \\ \mathcal{C} &= \left(x_i^a - x_i^b\right) - tol \geq 0 \\ x_{min} &\leq x_i \leq x_{max} \end{aligned} \tag{24}$$

where V is the current volume, V_{ini} is the initial volume, x_{min} is the lower bound, x_{max} is the upper bound, $x_i^a - x_i^b$ indicate the distance between control points (in a given direction) of each boundary and tol is the minimum distance adopted. These functions are evaluated by FEM, and the gradients by AD. The complete algorithm of shape optimization can be stated as follows:

1. Perform a structural analysis on initial geometry by FEM,

2. Evaluate the required gradients by reverse AD,

3. Check *KKT* conditions. If convergence is not obtained, find a new search direction and perform unidimensional minimization along this direction in order to obtain a new position $\mathbf{x}_{n+1}$ for the optimization variables $\mathbf{P}_{i,j}$,

4. Update internal control points of the geometry by linear interpolation of the distance between opposite boundaries in a given direction,

5. Perform a structural analysis on the new geometry by FEM.

The algorithm loops over the second and fifth steps if the convergence is not achieved. When convergence is achieved, the algorithm just performs the final structural analysis (step 5).

4 NUMERICAL APPLICATIONS

In order to demonstrate and validate the shape optimization algorithm, some examples are presented in this section. The material properties for all the examples are: Young's elastic modulus $E = 3.10 \times 10^{10}$, Poison coefficient $\nu = 0.2$ and specific mass $\gamma = 2.5 \times 10^4$. The values used are in the SI. The objective function to be minimized in all examples is the strain energy and the equality constraint of constant volume is imposed in all examples. Other constraints adopted for each example are properly specified in the description of each case. For all examples, the initially uniform load remains constant on all geometries through the optimization process.

4.1 Example 1

The initial geometry of this example is a disk shape. An initial height of $h = 1$ and radius $R = 5$ are used. The disk is simply supported on its external point and a uniformly distributed load of $q = 750 \times 10^3$ is applied on the upper boundary (see Figure 4). No body forces are considered in this example.

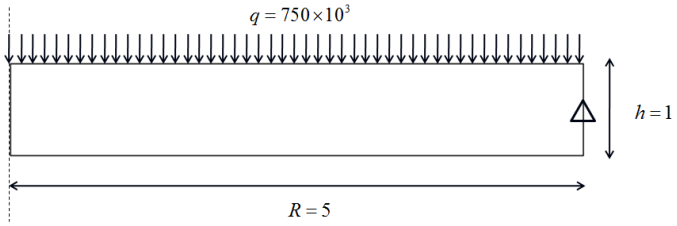

Figure 4: Initial geometry and boundary conditions for example 1.

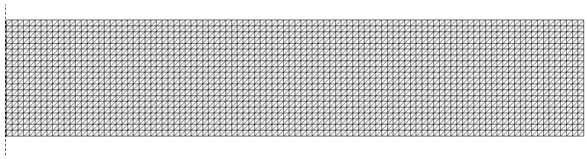

Figure 5: Finite element mesh for example 1.

The finite element mesh has 2121 nodes and 4000 elements as shown in Figure 5. The NURBS parametrization is done with 16 control points, where 8 of them were adopted as optimization variables (4 in the upper boundary and 4 in the bottom). Figure 6 shows the adopted optimization variables with red color and the other with blue color. The basis function are defined over the knot vectors

$$\Xi = \{0, 0, 0, 0, 1, 1, 1, 1\} \tag{25}$$

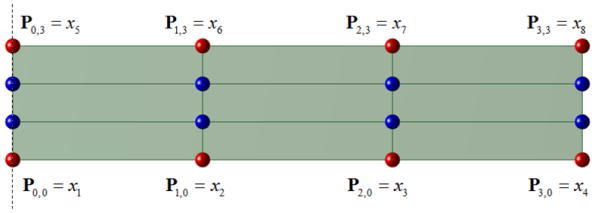

Figure 6: Surface parametrization and adopted optimization variables for example 1.

$$\mathcal{H} = \{0, 0, 0, 0, 1, 1, 1, 1\} \tag{26}$$

The control points are allowed to move in the vertical direction bounded by an upper limit of 25 and lower limit of -25. Also, vertical distance between upper and lower boundary control points are limited to be greater or equal to 0.15 (to avoid interpenetration of one point into other one). The initial coordinates of the 16 control points are shown in Table 1.

$\mathbf{P}$	x	y
(0,0)	0.00	0.00
(1,0)	1.67	0.00
(2,0)	3.33	0.00
(3,0)	5.00	0.00
(0,1)	0.00	0.33
(1,1)	1.67	0.33
(2,1)	3.33	0.33
(3,1)	5.00	0.33
(0,2)	0.00	0.67
(1,2)	1.67	0.67
(2,2)	3.33	0.67
(3,2)	5.00	0.67
(0,3)	0.00	1.00
(1,3)	1.67	1.00
(2,3)	3.33	1.00
(3,3)	5.00	1.00

Table 1: Initial coordinates of control points for example 1.

The imposed constraint of constant volume forces the geometry to search the best mass distribution in order to minimize the strain energy with fixed boundary conditions and load, avoiding a simple mass increase in order to obtain a lower strain energy. Figure 7 shows the evolution of the geometry along the iterations until an optimized shape is obtained. At the 28th iteration, the optimum condition is reached. The final relative objective function obtained $\mathcal{F}_{opt}$ is $\mathcal{F}_{28}/\mathcal{F}_{ini} = 0.114$. A comparison between the displacements and the von Mises stresses in initial and final optimized geometry is shown in Figure 8. The decrease of the relative objective function in terms of the iterations is presented in Figure 9. Also it is presented the final geometry. The final coordinates of the 16 control points are shown in Table 2. It can be noticed from Figure 9 that the initial iterations imply in the greatest reduction of the objective function. The final iterations do not reduce the objective function so much, but they play a role in fixing the curvature, as can be observed in Figure 7. From the comparison between displacement and

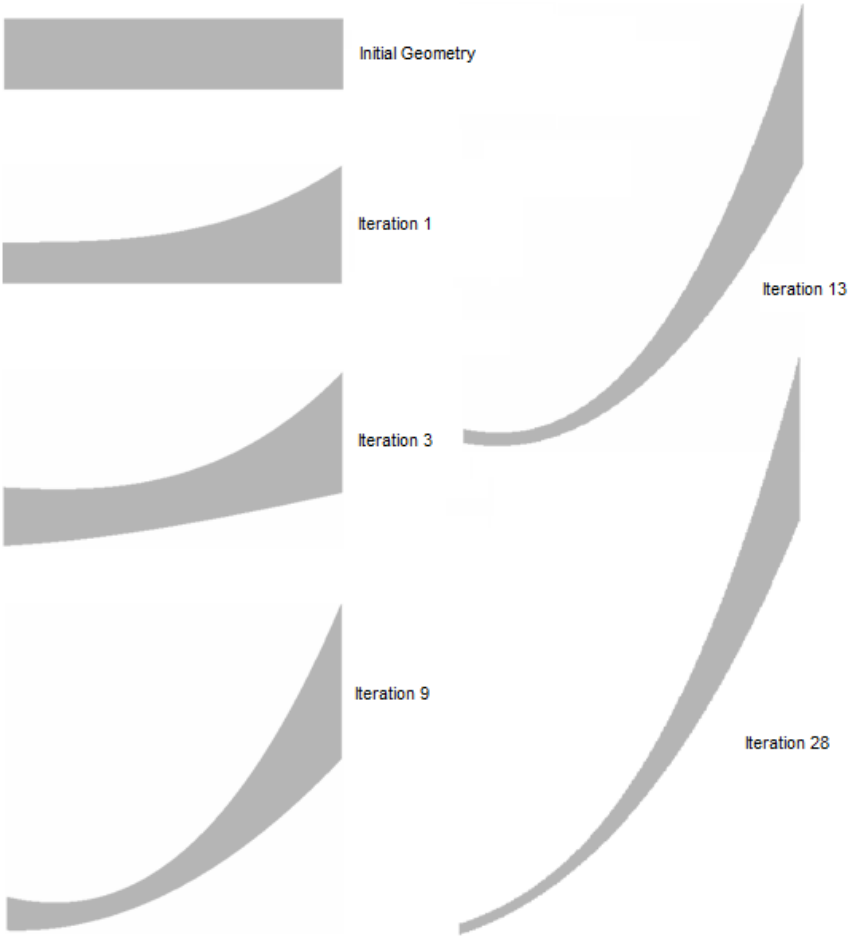

Figure 7: Sequence of shapes corresponding to different number of iteration for example 1.

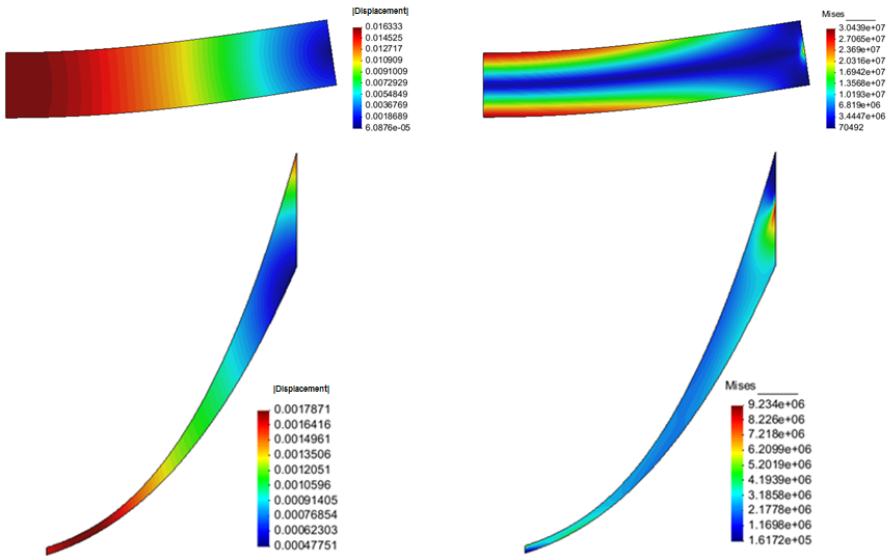

Figure 8: Comparison between displacements and von Mises stresses in the initial and the final optimized geometries for example 1.

von Mises stresses it can be noted the significant reduction of both in the optimized geometry.

P	x	y
(0,0)	0.00	-14.98
(1,0)	1.67	-14.45
(2,0)	3.33	-13.02
(3,0)	5.00	-9.26
(0,1)	0.00	-14.93
(1,1)	1.67	-14.40
(2,1)	3.33	-12.97
(3,1)	5.00	-8.50
(0,2)	0.00	-14.88
(1,2)	1.67	-14.35
(2,2)	3.33	-12.92
(3,2)	5.00	-7.74
(0,3)	0.00	-14.83
(1,3)	1.67	-14.30
(2,3)	3.33	-12.87
(3,3)	5.00	-6.98

Table 2: Final coordinates of control points for example 1.

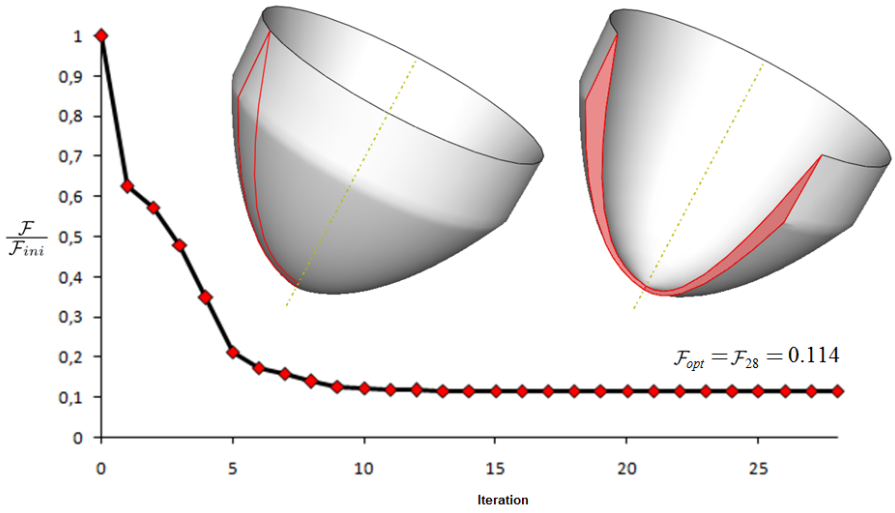

Figure 9: Relative objective function as a function of the iterations and final shape for example 1.

Along the optimization iterations, the original high flexural and shear stresses are progressively reduced, while the membrane stresses are increased. The final geometry configuration tends to approach a parabolic arc shape. It can be noticed that using the optimized shape with an applied load in the inverse direction, i.e. oriented from the bottom to the external surface, the structure will work primary with compression stresses. This is an important result and has been extensively used for many engineering applications (such as the arc bridge). This type of geometry is a well-known established result of minimization of strain energy. For the one-dimensional case, this minimization leads to a catenary shape configurations. Analogous results for shell structures are presented by Ramm and Wall (1961). An experimental work on this area was performed by Isler (1961), where the form was found by hanging models and a membrane configuration under compression was also obtained.

4.2 Example 2

The initial geometry of this example is analogous to the example 1 but with a central hole. An initial height of $h = 1.5$, internal radius $R_i = 5$ and external radius $R_e = 10$ were adopted using the same boundary condition of example 1 (simply supported) and a uniformly distributed load of $q = 268 \times 10^3$ is applied on the upper boundary (see Figure 10). No body forces are considered in this example.

The finite element mesh has 952 nodes and 1742 elements as shown in Figure 11. The NURBS parametrization is done with 42 control points, where 14 of them were adopted as optimization variables (7 in the upper boundary and 7 in the bottom). Figure 12 shows the adopted optimization nodes with red color and the others with blue color. The basis function are defined over the knot vectors:

$$\Xi = \{0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1\} \tag{27}$$

$$\mathcal{H} = \{0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1\} \tag{28}$$

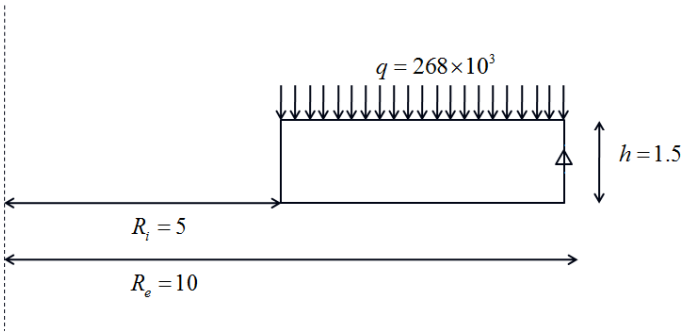

Figure 10: Initial geometry and boundary conditions for example 2.

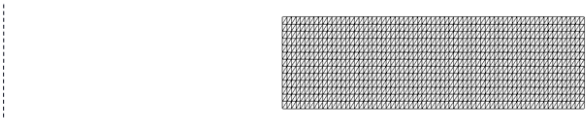

Figure 11: Finite element mesh for example 2.

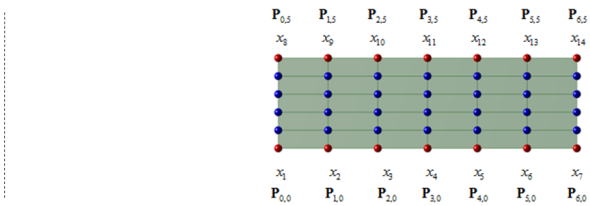

Figure 12: Surface parametrization and adopted optimization variables for example 2.

The 14 optimization variables are allowed to move in the vertical direction bounded by an upper limit of 7.5 and a lower limit of -7.5. Also, vertical distance between upper and lower

boundary control points are limited to be great or equal to 0.15. Figure 13 shows the geometry optimization evolution along the iterations.

At the 39th iteration, the optimum condition is reached. The final relative objective function obtained $\mathcal{F}_{opt}$ is $\mathcal{F}_{39}/\mathcal{F}_{ini} = 0.068$. Figure 14 shows a comparison between the initial and the final optimized geometry for displacements and von Mises stresses. The decrease of the

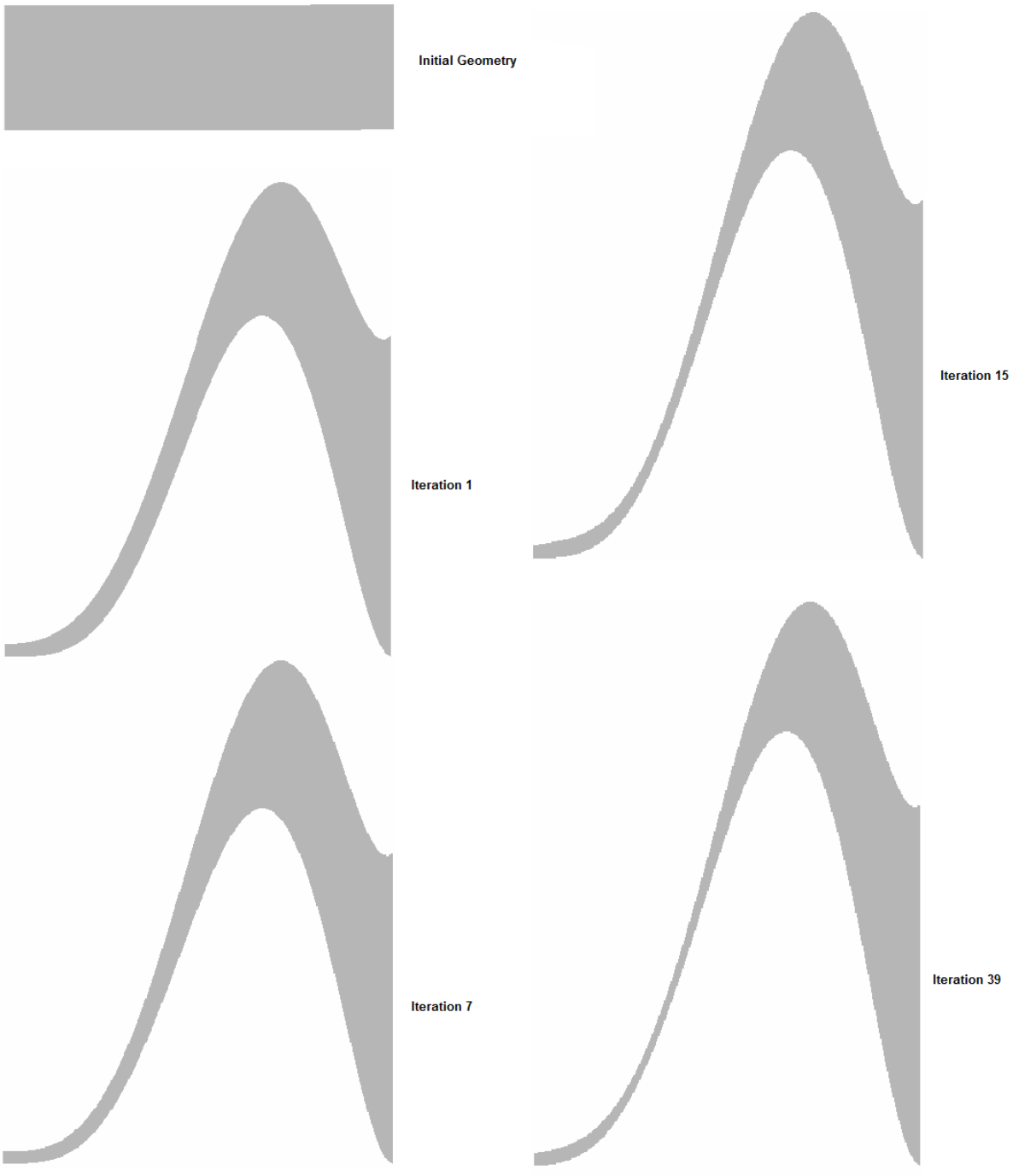

Figure 13: Sequence of shapes corresponding to different number of iterations for example 2.

relative objective function as a function of the iterations is presented in Figure 15. In this figure the final geometry is also presented. The initial and final coordinates of the 16 optimization variables are shown in Table 3.

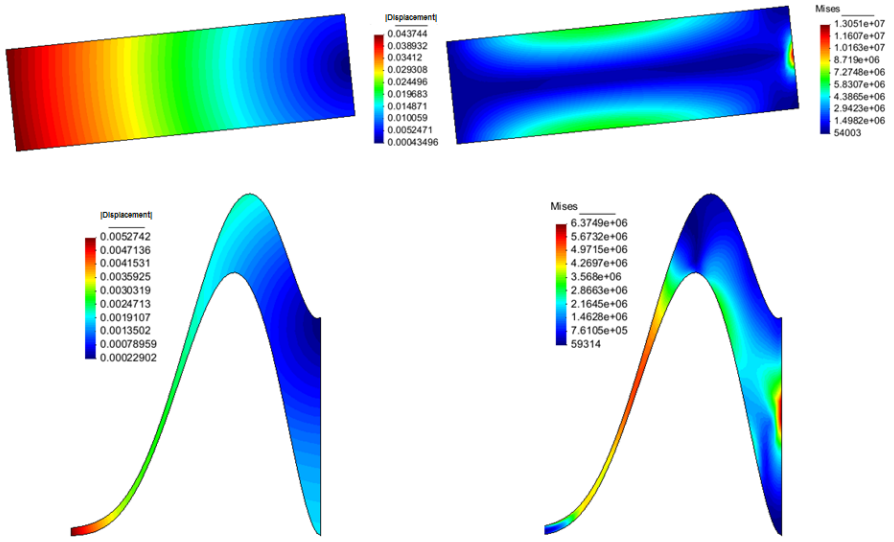

Figure 14: Comparison between displacements and von Mises stresses in the initial and the final optimized geometries for example 2.

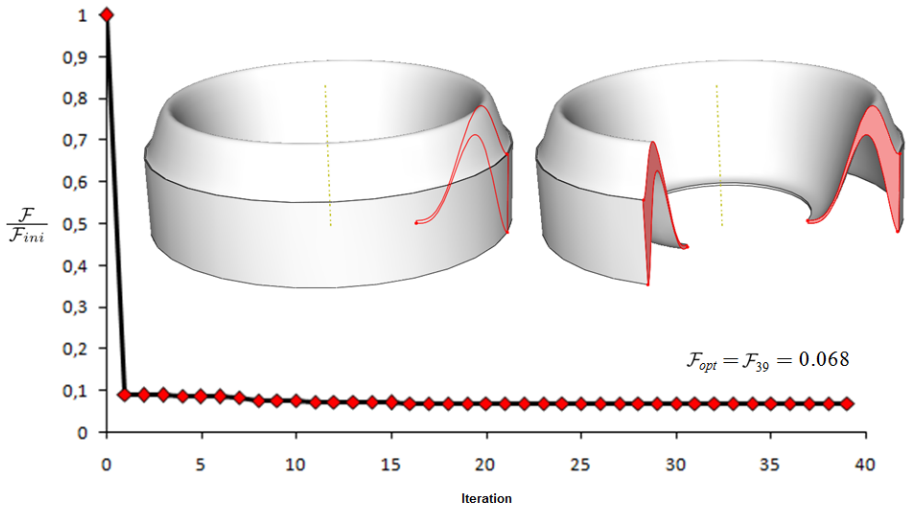

Figure 15: Relative objective function as function of the iteractions for example 2.

In can be seen from Figure 15 that in the first iteration a great reduction of the objective function is obtained and in the next ones reduction of the value of the objective function continue but with a relatively lower intensity. This is a consequence of the shape optimization evolution shown in Figure 13, where initially a double curvature shape is formed with the mass being concentrated at the supports of the structure due to the stress concentration on this area. However, it can be seen that the upper and lower bounds fixed for the optimization variables displacements are reached. This way, these restrictions do not allow curvature to increase and the optimal geometry obtained looks like two inverted parabolas.

In Figure 13 strong differences in thickness can be observed. Near the the support region of the structure, where the optimal shape has the thickest region, it cannot be correctly described with the shell theory approach. Thus, this example shows that a structural approach based on solid is indispensable, when three different behaviors (related to thin, thick shells as well as solids structures) are simultaneously present.

P	x	y_i	y_f
(0,0)	5.00	0.00	-7.50
(1,0)	5.83	0.00	-7.50
(2,0)	6.67	0.00	-7.05
(3,0)	7.50	0.00	-6.01
(4,0)	8.33	0.00	7.35
(5,0)	9.17	0.00	-7.50
(6,0)	10.00	0.00	-7.50
(0,5)	5.00	1.50	-7.35
(1,5)	5.83	1.50	-7.24
(2,5)	6.67	1.50	-6.90
(3,5)	7.50	1.50	-5.86
(4,5)	8.33	1.50	-7.50
(5,5)	9.17	1.50	-3.83
(6,5)	10.00	1.50	-3.13

Table 3: Initial and final coordinates of optimization control points for example 2.

5 CONCLUSION

The model developed to optimize axisymmetric solids showed to be very useful. The optimized shapes found are consistent, having direct physical interpretation and better structural performance. The shape manipulation using a NURBS description promotes an easy form to perform the changes in the geometry while preventing the use of many optimization variables. The methodology applied to move the mesh, avoiding high mesh distortions and control points interpenetration has shown to be a good option instead of the use of remeshing techniques, which are much more expensive computationally and the imposition of this method is coupled with the optimization algorithm, since constraints are imposed to move the mesh. The sensitivity analysis performed by reverse automatic differentiation guarantees not only exact derivative computations but also computational efficiency, and, as previously mentioned, gradient evaluation is one of the main well-known numerical difficulties of optimization problems. Indeed, the use of AD for solid optimization is a little explored area until now. The SQP is a robust numerical optimization approach, as it can be observed once the initial geometry is compared with the final one obtained, both being very distinct in many attributes. The optimization procedure provides here the possibility of a continuous variation of the thickness, which is harder to be obtained with the use of a finite element shell formulation. On the other hand, axisymmetric geometries and loads are a more limited problem. The use of this methodology for plane stress and plane strain cases is straightforward.

6 ACKNOWLEDGMENTS

The authors wish to thank to CAPES and CNPq (Brazilian Research Committees) for their financial support and they thank also to CESUP (UFRGS Supercomputing Center) for its important contributions.

REFERENCES

Aubert P. and Rousselet B. Sensitivity computation and shape optimization for a non-linear arch model with limit-points instabilities. *International Journal For Numerical Methods In*

Engineering, 42:15–48, 1998.

Bletzinger K.U., Firl M., Linhard J., and Wüchner R. Optimal shapes of mechanically motivated surfaces. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):324–333, 2008.

Csonka B. and Kozák I. Shape optimization of axisymmetric structures using a high-order shear deformation theory. *Structural Optimization*, Vol. 9:117–127, 1995.

Espath L., Linn R., and Awruch A. Shape optimization of shell structures based on nurbs description using automatic differentiation. *International Journal for Numerical Methods in Engineering*, Vol. 88:613–636, 2011.

Griewank A. *Some Bounds on the Complexity of Gradients, Jacobians, and Hessian*. Complexity in Nonlinear Optimization, 1993.

Griewank A. A mathematical view of automatic differentiation. *Acta Numerica*, Vol. 22:321–398, 2003.

Griewank A. and Walther A. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Philadelphia, SIAM, 2^{nd} edition, 2008.

INRIA. *Tapenade AD*. http://tapenade.inria.fr:8080/tapenade/index.jsp, 2002.

Isler H. New shape for shells. *Bulletim of the International Association for Shell Structures*, Vol. 8:123–130, 1961.

Khosravi P., Ganesan R., and Sedaghati R. Optimization of thin-walled structures with geometric nonlinearity for maximum critical buckling load using optimality criteria. *Thin-Walled Structures*, 46:1319 – 1328, 2008.

Mota Soares C.A., Barbosa J.I., and Mota Soares C.M. Axisymmetric thin shell structures sizing and shape optimization. *Control Cybernet.*, Vol. 23(3):513–551, 1994. Shape design and optimization.

Nocedal J. and Wright S.J. *Numerical Optimization*. Springer, 2^{nd} edition, 1999.

Piegl L.A. and Tiller W. *The Nurbs Book*. Springer, 2^{nd} edition, 1997.

Ramm E., Bletzinger K., and Reitinger R. Shape optimization of shell structures. *IASS Bulletin of the international association for shelland spatial structures*, Vol. 34(112):103–121, 1993.

Ramm E. and Wall W. Shell structures - a sensitivity interrelation between physics and numerics. *International Journal for Numerical Methods in Engineering*, Vol. 60(1):381–427, 1961.

Reitinger R. and Ramm E. Buckling and imperfection sensitivity in the optimization of shell structures. *Thin-Walled Structures*, Vol. 23(1-4):159–177, 1995.

Rousselet B., Mehrez S., Myslinski A., and Piekarski J. Shell optimization, some remarks. Herskovits, José (ed.), Advances in structural optimization. Dordrecht: Kluwer Academic Publishers. Solid Mech. Appl. 25, 381-412 (1995)., 1995.

Özakça M., Hinton E., and Rao N. Shape optimization of axisymmetric structures with adaptive finite element procedures. *Structural Optimization*, Vol. 5:256–264, 1993.