Control and Cybernetics

vol. **23** (1994) No. 3

Shape optimization of shells

by

Denise Chenais

Laboratoire de Mathématiques
CNRS–URA 168
Université de Nice
Parc Valrose
06108 Nice 2
France

We consider a nonshallow shell working in linear elastic conditions, subject to a given load. Its response is the solution of an elliptic partial differential system of equations. Our aim is to change the shape of the shell so that it behave as well as possible with respect to a given criterion which depends on the shell through the solution of the p.d.e.. This is an optimal control problem governed by a partial differential system of equations. The shape of the shell is given by the shape of its midsurface on the one hand, and by its thickness on the other hand. We will discuss essentially *the midsurface shape optimization* and make just a few comments about the thickness.

We work out this problem using classical optimization techniques, based on gradient methods. Our main subject is to differerentiate the criterion which has to be minimized; this requires differentiation of the solution of the p.d.e. with respect to the shape. We give a mathematical proof of this differentiability and deduce an analytical expression of the differential of the functional which has to be minimized, when this one depends on the state in a differentiable manner. We also give some results using nonsmooth optimization techniques allowing, for instance, optimization of the "max type" functionals.

After having given mathematical proofs, we show how to approximate the differential in order to get *numerical approximations* of local optima. We give numerical results concerning two examples: arches and axisymmetric shells.

1. Shell equations

1.1. The geometry

A shell Ω_e is a 3–dimensional solid body which has a small thickness e around a given midsuface ω. This midsurface is given as the image of a given bounded open set $\hat{\omega} = \{\hat{m} = (\xi_1, \xi_2)\} \subset \mathbf{R}^2$ by a mapping φ, which is supposed to be of class $W^{3,\infty}$ (it has bounded third order derivatives in the sense of distributions; in other words, it is twice continuously differentiable, and its second order partial derivatives are lipschitz continuous — see Brezis, 1983). It is also assumed that ω is *regular*, which means that for all $m \in \omega$, the two vectors:

$$a_\alpha(m) = \frac{\partial \varphi}{\partial \xi_\alpha}(\hat{m}) \qquad \alpha = 1, 2$$

which are tangent to ω at the point m are linearly independent. They generate the tangent plane to ω at the point m.

(From now on, as usual in shell theory, Greek indices will take the values 1 and 2. We will also use the Einstein summation convention.)

The vector:

$$n(m) = \frac{a_1(m) \times a_2(m)}{||a_1(m) \times a_2(m)||}$$

is the unit normal vector to the surface ω at the point m. For a given small real parameter e, the shell Ω_e is the following set:

$$\Omega_e = \{M = m + x_3 n(m); m = \varphi(\hat{m}), \hat{m} \in \hat{\omega}; x_3 \in]-e, +e[\}.$$

It is a subset of the 3-dimensional euclidian space that we denote by $\mathcal{E}^3$.

As a matter of fact, a point M in Ω_e is identified by the pair (m, x_3) such that:

$$M = m + x_3 n(m).$$

This requires that e be small enough compared to the curvature of ω so that such a pair be unique.

In the sequel, the following geometrical quantities will be used:

(For anybody who is not familiar with tensor calculus, all the tensors we describe can be simply regarded as 2×2 matrices with the upper or left index as the line index and the lower or right one as the column one).

- a^1, a^2 denote the dual basis of a_1, a_2. They are defined by:
  $$< a^\alpha, a_\beta >= \delta^\alpha_\beta,$$
  ($< .,. >$ denotes the inner product, a^α is a covariant vector, a_α is contravariant).
- $a_{\alpha\beta} =< a_\alpha, a_\beta >, \quad a^{\alpha\beta} =< a^\alpha, a^\beta >$.
- The Christoffel symbols of the surface ω are the following numbers:
  $$\Gamma^\alpha_{\beta\lambda} =< a^\alpha, a_{\beta\lambda} >$$
  where:
  $$a_{\beta\lambda} = \frac{\partial^2 \varphi}{\partial \xi_\beta \partial \xi_\lambda}.$$

- Let Γ^3 and Λ be the two following tensors:
  $$\Gamma^3_{\alpha\beta} = < n, a_{\alpha\beta} > \qquad \Lambda_{\alpha\beta} = < a^\alpha, a^\beta >,$$
  the curvature tensor $B = (b^\alpha_\beta)$ is defined by:
  $$B = \Lambda\Gamma^3$$
  or in components:
  $$b^\alpha_\beta = \Lambda^\alpha_\mu \Gamma^3_{\mu\beta}.$$

1.2. The mechanical equations

This shell is now submitted to a load. This creates a displacement field on Ω_e. In shell theories assumptions are made, like the Kirchhoff-Love ones for instance, which make the displacement at a point M to be completely determined by the displacement of $m \in \omega$, where $M = m + x_3 n(m)$. Therefore the real unknown displacement field is the displacement field of the *midsurface* that we denote by $\mathbf{u}$. In the following, we will use the Budianski-Sanders model. It is very close to the Koiter one. Anyway, most of the techniques we develop here are directely adaptable to several other models.

We give now the equations of the Budianski–Sanders model. At a point m $\in \omega$ we consider the local basis made of the three vectors $a_1(m), a_2(m), n(m)$. The vector field $\mathbf{u}$ can be decomposed as:

$$\mathbf{u}(m) = u^1(m)a_1(m) + u^2(m)a_2(m) + u^3(m)n(m),$$

so that the unknown is now the triplet $u = (u^1, u^2, u^3)$ of functions defined on ω. The membrane strain tensor is given by:

$$\gamma(u) = \gamma^\alpha_\beta(u)\; a_\alpha \otimes a^\beta \tag{1}$$

with:

$$\gamma^\alpha_\beta(u) = \frac{1}{2}[u^\alpha|_\beta + u_\beta|^\alpha] - b^\alpha_\beta u^3$$
$$u^\alpha|_\beta = u^\alpha_{,\beta} + \Gamma^\alpha_{\beta\lambda} u^\lambda$$
$$u_\beta|^\alpha = a^{\alpha\lambda} a_{\beta\nu} u^\nu|_\lambda$$

$\Gamma^\alpha_{\beta\lambda}$ being the Christoffel symbols of the surface
b^α_β being the curvature tensor
($u^\alpha_{,\beta}$ denotes $\frac{\partial u^\alpha}{\partial \xi_\beta}$).

The bending strain tensor is given by:

$$\rho(u) = \rho^\alpha_\beta(u)\; a_\alpha \otimes a^\beta \tag{2}$$

with:

$$\begin{aligned} \rho^\alpha_\beta &= \frac{1}{2}[\theta^\alpha|_\beta + \theta_\beta|^\alpha] \\ \theta^\alpha &= -b^\alpha_\lambda u^\lambda - a^{\alpha\lambda} u^3_{,\lambda} \\ \theta^\alpha|_\beta &= \theta^\alpha_{,\beta} + \Gamma^\alpha_{\beta\lambda}\theta^\lambda \\ \theta_\beta|^\alpha &= a^{\alpha\lambda} a_{\beta\nu} \theta^\nu|_\lambda \end{aligned}$$

($\theta = \theta^\alpha a_\alpha$ is the rotation of the normal induced by the displacement).

The membrane energy in the displacement field $\mathbf{u}$ is:

$$a_m^\omega(u,u) = \frac{E}{1-\nu^2}\int_\omega \{(1-\nu)tr[\gamma(u)\gamma(u)] + \nu tr\gamma(u)tr\gamma(u)\}(m)\, dm$$

where $tr\gamma = \gamma_1^1 + \gamma_2^2$ is the trace of the tensor γ.

The bending energy is:

$$a_b^\omega(u,u) = \frac{E}{1-\nu^2}\int_\omega \{(1-\nu)tr[\rho(u)\rho(u)] + \nu tr\rho(u)tr\rho(u)\}(m)dm.$$

Let $L^\omega(v)$ be the virtual work of the external loads in the virtual displacement v. The displacement field $u_\omega = (u_\omega^1, u_\omega^2, u_\omega^3)$ in the shell for this loading is the unique solution of the following equation:

$$u_\omega \in \mathbf{V}_\omega \qquad a^\omega(u_\omega, v) = L^\omega(v) \qquad \forall v \in \mathbf{V}_\omega \tag{3}$$

where:

$$\mathbf{V}_\omega = [H_0^1(\omega)]^2 \times H_0^2(\omega)$$

and:

$$a^\omega(u,v) = 2ea_m^\omega(u,v) + 2\frac{e^3}{3}a_b^\omega(u,v) \tag{4}$$

It is proved in Destuynder (1985) that a^ω is bilinear continuous symmetric coercive, so that this equation has a unique solution. It is nothing but the virtual work principle. Notice that we have given a displacement formulation.

The membrane and bending stresses $n(u)$ and $m(u)$ are given by:

$$\begin{aligned} n(u) &= \frac{E}{1-\nu^2}\{(1-\nu)\gamma(u) + \nu tr\gamma(u)Id\} \\ m(u) &= \frac{E}{1-\nu^2}\{(1-\nu)\rho(u) + \nu tr\rho(u)Id\}. \end{aligned}$$

(Id denotes the identity of the tangent plane).

2. Optimal control results in an abstract setting

We will see in the next sections that the shape optimization of shells can be regarded as a classical optimal control problem governed by a partial differential equation. So we recall usual results.

2.1. The general setting

- $\mathbf{V}$ is a Hilbert space, $\mathbf{\Lambda}$ is a Banach space and Φ is an open subset of $\mathbf{\Lambda}$,
- $a : \Phi \times \mathbf{V} \times \mathbf{V} \longrightarrow \mathbf{R}$ is for each $\varphi \in \Phi$ a bilinear continuous, symmetric coercive functional on $\mathbf{V} \times \mathbf{V}$
- $L : \Phi \times \mathbf{V} \longrightarrow \mathbf{R}$ is for each $\varphi \in \Phi$ a linear continuous functional on $\mathbf{V}$,
- $J : \Phi \times \mathbf{V} \longrightarrow \mathbf{R}$ is a continuously differentiable given functional.

By the Lax–Milgram theorem, for each $\varphi \in \Phi$ the equation:

$$u_\varphi \in \mathbf{V}, \qquad a(\varphi; u_\varphi, v) = L(\varphi; v) \qquad \forall v \in \mathbf{V} \tag{5}$$

has one and only one solution. In the problems we consider, this is the weak form of a partial differential equation.

The problem is the following:

$$Min \ \{j(\varphi);\ \varphi \in \Phi\},$$

where $j(\varphi)$ is defined by:

$$j(\varphi) = J(\varphi, u_\varphi).$$

This is a classical optimal control problem governed by an elliptic partial differential equation.

We do not discuss here the existence of an optimal control. As for the uniqueness, for the problems we are concerned by, usually nothing is known, because we have no information about the convexity of the functional j. We give a way to build descent algorithms, which should converge to at least local optima, and hopefully to the global optimum.

We plan to use gradient type methods, so that the main subject is to differentiate j with respect to φ. We do it differentiating first the continuous problem. After this, we see how to compute a numerical approximation of this differential in order to use it in a gradient method in finite dimension.

REMARK 2.1 *The following results also hold if the bilinear functional $a(\varphi; .,.)$ is not coercive. It is sufficient that it define an isomorphism of $\mathbf{V}$ (see Lods, 1992).*

REMARK 2.2 *We are also able to give results concerning nondifferentiable functionals j, on the condition that the nondifferentiability fits in the general setting of nonsmooth optimization as studied for instance by Zowe(1985), Lemarechal (1980).*

2.2. Differentiability of the continuous problem

The results we give here concern only the case of continuously differentiable functionals J. We have the following theorems.

THEOREM 2.1 (see Rousselet, 1982) *Let φ be given in $\Phi \in \mathbf{\Lambda}$. We suppose that the bilinear functional $a(\varphi; ., .)$ and the linear functional $L(\varphi; .)$ are Fréchet differentiable. This means that there exist $\frac{\partial a}{\partial \varphi}(\varphi; u, v)$ and $\frac{\partial L}{\partial \varphi}(\varphi; v)$ linear continuous functionals defined on $\mathbf{\Lambda}$ such that for all $\psi \in \mathbf{\Lambda}$:*

- *H1a)* $|\frac{\partial a}{\partial \varphi}(\varphi; u, v).\psi| \leq C(\varphi)||\psi||_{\mathbf{\Lambda}}\ ||u||_{\mathbf{V}}\ ||v||_{\mathbf{V}}$
- *H2a)* $|\delta^2_\psi a(\varphi; u, v)| \leq \epsilon(\psi)||\psi||_{\mathbf{\Lambda}}\ ||u||_{\mathbf{V}}\ ||v||_{\mathbf{V}}$
- *H1L)* $|\frac{\partial L}{\partial \varphi}(\varphi; v).\psi| \leq C(\varphi)||\psi||_{\mathbf{\Lambda}}\ ||v||_{\mathbf{V}}$
- *H2L)* $|\delta^2_\psi L(\varphi; v)| \leq \epsilon(\psi)||\psi||_{\mathbf{\Lambda}}\ ||v||_{\mathbf{V}}$

where:

$$\delta^2_\psi a(\varphi; u, v) = a(\varphi + \psi; u, v) - a(\varphi; u, v) - \frac{\partial a}{\partial \varphi}(\varphi; u, v).\psi$$

and similarly for $\delta^2_\psi L(\varphi; v)$.

Then the mapping $\varphi \mapsto u_\varphi : \Phi \subset \mathbf{\Lambda} \longrightarrow \mathbf{V}$ is Fréchet differentiable.

REMARK 2.3 *If the bilinear form $a(\varphi; ., .)$ and the linear form $L(\varphi; .)$ were* continuously differentiable *with respect to φ, this would just be a consequence of the implicit functions theorem. As a matter of fact even when these functionals are continuously differentiable, it is often easier to prove that they are differentiable using H1a), H2a), H1L), H2L). So it can be more convenient to use this theorem than the implicit functions theorem.*

THEOREM 2.2 (see Chenais, 1987, for instance) *If, moreover $J(\varphi; v)$ is continuously differentiable, then j is Fréchet differentiable with respect to φ, and its differential can be explicitly written using the adjoint state p_φ defined by:*

$$a(\varphi; p_\varphi, w) = \frac{\partial J}{\partial \varphi}(\varphi; u_\varphi).w \qquad \forall w \in \mathbf{V} \tag{6}$$

as follows:

$$\frac{dj}{d\varphi}(\varphi).\psi = \frac{\partial J}{\partial \varphi}(\varphi; u_\varphi).\psi - \frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi + \frac{\partial L}{\partial \varphi}(\varphi; p_\varphi).\psi \tag{7}$$

3. Application to the shell equations; differentiability of the continuous problem

At a first glance, the shell equations do not fit in the abstract setting described in section 2. Actually, in the shell state equation, the space $\mathbf{V}_\omega$ depends on the design variable ω. This is of course not allowed in the general setting as the aim is to differentiate with respect to ω. As a matter of fact, this is the standard main question in shape optimization.

In the shell equations that we are studying, ω is the image of a given bounded open set $\hat{\omega}$ of $\mathbf{R}^2$ by the mapping φ. So it is natural to get rid of this difficulty by taking φ as the design variable instead of ω. This means that we fix $\hat{\omega}$ and search φ in order to get ω. This gives of course the usual restriction of this kind of methods: ω is searched in a set of surfaces which are manifolds having the same topology as $\hat{\omega}$. One main hypothesis is necessary for the use of this kind of method : the change of variable, which is here φ, induces an isomorphism between $\mathbf{V}_\omega = [H_0^1(\omega)]^2 \times H_0^2(\omega)$ and $\mathbf{V} = [H_0^1(\hat{\omega})]^2 \times H_0^2(\hat{\omega})$. This is clearly fulfilled here because $\varphi \in \mathbf{W}^{3,\infty}(\hat{\omega})$ and it is regular. So, we do this change of variable in the state equation of shells (eq.(3)).

3.1. Shell equations written on $\hat{\omega}$

We go back to the shell equations described in section 1 and make the change of variables $m = \varphi(\hat{m})$, where $\hat{m} = (\xi_1, \xi_2) \in \hat{\omega}$ which is fixed. As a matter of fact, several expressions were already implicitely written in terms of $\hat{m}$. We get the following equation:

$$u_\varphi \in \mathbf{V} \qquad a(\varphi; u_\varphi, v) = L(\varphi; v) \qquad \forall v \in \mathbf{V} \tag{8}$$

with:

- $\mathbf{V} = [H_0^1(\hat{\omega})]^2 \times H_0^2(\hat{\omega})$
- $a(\varphi; u, v) = 2ea_m(\varphi; u, v) + 2\frac{e^3}{3}a_b(\varphi; u, v)$
- $a_m(\varphi; u, v) = \frac{E}{1-\nu^2}\int_{\hat{\omega}}\{(1-\nu)tr[\gamma(\varphi; u)\gamma(\varphi; v)]$ $\qquad + \nu tr\gamma(\varphi; u)tr\gamma(\varphi; v)\}(\hat{m})[g(\hat{m})]^{1/2}\, d\hat{m}$
- $a_b(\varphi; u, v) = \frac{E}{1-\nu^2}\int_{\hat{\omega}}\{(1-\nu)tr[\rho(\varphi; u)\rho(\varphi; v)]$ $\qquad + \nu tr\rho(\varphi; u)tr\rho(\varphi; v)\}(\hat{m})\ [g(\hat{m})]^{1/2}\, d\hat{m}$
- $g(\hat{m}) =< a_1(\hat{m}), a_1(\hat{m}) >< a_2(\hat{m}), a_2(\hat{m}) > - < a_1(\hat{m}), a_2(\hat{m}) >^2$
- $\gamma(\varphi; u)$ and $\rho(\varphi; u)$ are given in section 1, equations(1) and (2). We have used the abusive notation $u(\hat{m}) = u(m)$, and as $u(\hat{m})$ and φ are decoupled, it is possible to write γ and ρ as functions of the two independent variables φ and u.

The functional a is now defined on $\Phi \times \mathbf{V} \times \mathbf{V}$ where Φ is an open subset of $W^{3,\infty}(\hat{\omega})$. It is bilinear symmetric continuous coercive on $V \times V$, so we are in the setting of section 2.

3.2. The continuous optimal control problem; differentiability

We consider now a functional $J_\omega(v)$ which is assumed to be defined on each $\omega = \varphi(\hat{\omega})$, for $\varphi \in \Phi$, and for $v \in \mathbf{V}_\omega$. We also make the change of variable $m = \varphi(\hat{m})$ to get another functional that we still abusively denote $J(\varphi; v)$ defined on $\Phi \times \mathbf{V}$. We suppose that it is of class $\mathcal{C}^1$. For example, we consider:

$$J = \int_\omega ||\mathbf{v}(m)||^2 dm$$

which gives:

$$J(\varphi; v) = \int_{\hat{\omega}} \{||v^1(\hat{m})a_1(\hat{m}) + v^2(\hat{m})a_2(\hat{m})||^2 + [v^3(\hat{m})]^2\}\ [g(\hat{m})]^{1/2}\ d\hat{m}.$$

We have to minimize $j(\varphi) = J(\varphi, u_\varphi)$ where u_φ is the solution of equation (8), for $\varphi \in \Phi$. We are now exactly in the context described in section 2. We hope to use a descent method in order to make the functional decrease. We would like to use the theorems 2.1 and 2.2. We have to check the differentiability conditions on a and L. Concerning L, this is a hypothesis on the type of loading we can deal with. About a, this hypothesis has to be checked. This is done extensively in Chenais (1987). We give here just the basic ideas. First, as $a(\varphi; u, v)$ is an integral, in order to differentiate it suffice to differentiate the function which is integrated from $\Phi \in W^{3,\infty}(\hat{\omega})$ into $L^1(\hat{\omega})$. Then looking at this function, it is easy to see that it is sufficient that both γ and ρ be differentiable with values in $[L^2(\hat{\omega})]^4$. These are linear in u with coefficients made of derivatives of φ at the order at most 3. It is not difficult to see that each of these coefficients is differentiable with respect to φ with values in $L^\infty(\hat{\omega})$. With all these basic ingredients, one can prove that the differentiability conditions required to use the theorem 2.1 are fulfilled.

Now, we know that if the loading is regular enough and if the functional which has to be minimized is continuously differentiable, then $j(\varphi)$ is Fréchet differentiable, and its differential is given by the formula (7). An extensive computation of the term $\frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi$ would be an awful work. So, before facing this question, we are going to discuss what is needed for from a practical point of view, i.e. for numerical computations.

REMARK 3.1 *As previously announced, we are also able to deal with nondifferentiable functionals J, as long as they fit in the so-called nonsmooth optimization techniques. It is the case, for instance, of functions of the type :*

$$J(u) = Max\{u(x); x \in \hat{\omega}\}.$$

One problem arises in general shell problems: the first components of the displacement a priori belong to the space $H_0^1(\hat{\omega})$. In two dimensions, these functions are not necessarily continuous, so such a functional is not defined. But in the case of particular geometries, like cylinders or axisymmetric shells, the dimension reduces to one, then the components of the displacement are continuous. In Habbal (1990) the specific problem of minimizing the maximum displacements in an arch (invariant cylinder in one direction) is studied. The hypothesis required in the non-smooth optimization techniques (existence of subdifferentials and use of subgradients, see Zowe, 1985; Lemarechal, 1980) are proved to be fulfilled, and numerical results are given. For the general study, we refer to Habbal (1990). In this article, we will just show some numerical results which have been obtained.

4. Numerical treatment

We need now to face the problem of choice of discretizations so that we can get an approximation of the functional we want to minimize and of its gradient, in order to give them as input parameters in an optimization procedure.

Let us look at the formula (7) giving the differential of j. From the three terms which are concerned, the first and the third ones depend on the chosen functional and on the chosen load. The middle one, corresponding to the energy depends only on the kind of structure which is concerned, and on its modelization. As for us, we are now concerned by the Budianski–Sanders model for shells.

This mid term is usually from far away the heaviest one. In this paper, we will only discuss the treatment of this term. The same technique has to be used for the other ones which will be much easier to handle. We recall that this mid term is:

$$\frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi$$

where φ is the actual shape of the shell, u_φ and p_φ are the direct and adjoint states for this shape, ψ is the direction in which the differentiation is performed. This function ψ lies in the vector space in which φ lies. We can also notice that this expression is given by an integral over $\hat{\omega}$.

From these general remarks, we see that we need discretizations for the three following subjects:

- the space of the design variable, which is here $W^{3,\infty}(\hat{\omega})$,
- the computation of u_φ and p_φ,
- the numerical computation of the integral.

These three questions will be treated in the three following subsections.

4.1. Discretization of the design variable space

Let us first briefly recall the principle of descent methods using gradients. The algorithm builds a sequence φ_n of shapes using basically the following idea:

if φ is known, it is improved by an increment ψ. We have:

$$j(\varphi + \psi) = j(\varphi) + \frac{dj}{d\varphi}(\varphi).\psi + o(||\psi||).$$

The usual basic idea is the following:

- compute $\frac{dj}{d\varphi}(\varphi)$
- choose $\psi = \frac{dj}{d\varphi}(\varphi)$.

We remember that here $\varphi \in W^{3,\infty}(\hat{\omega})$, which is an infinite dimensional Banach space, and not a Hilbert space. So $\frac{dj}{d\varphi}(\varphi)$ belongs to its dual which is not isomorphic to itself, and ψ which has to be taken in $W^{3,\infty}(\hat{\omega})$ cannot be taken equal to $\frac{dj}{d\varphi}(\varphi)$.

We will restrict ourselves to the search of an optimal φ in a *finite dimensional subvector space* of $W^{3,\infty}(\hat{\omega})$ that we will denote Λ_H. One standard family of such subvector spaces is a family of $\mathcal{C}^2$ spline functions, built on a discretization of $\hat{\omega}$. In the sequel, the nodes used to build such a vector space are called the master nodes. The discretization we are describing here is called the geometrical approximation.

Let $\{s_i; i = 1, ..., n\}$ be a basis of Λ_H. We suppose that we know $\varphi \in \Lambda_H$. We look for an increment $\psi \in \Lambda_H$ such that $j(\varphi + \psi)$ be as small as possible. Such a ψ is of the form:

$$\psi = \sum_1^n \alpha_i s_i$$

so that we search in fact the vector $\alpha = (\alpha_i; i = 1, ..., n)$ of real numbers. We have now:

$$j(\varphi + \psi) = j(\varphi) + \sum_1^n \alpha_i \frac{dj}{d\varphi}(\varphi).s_i + o(||\alpha||),$$

and $\frac{dj}{d\varphi}(\varphi).s_i$ is for each i a real number. The best choice for α is:

$$\alpha_i = \frac{dj}{d\varphi}(\varphi).s_i \qquad \textit{for } i = 1, ..., n.$$

The vector $\{\frac{dj}{d\varphi}(\varphi).s_i; i = 1, ..., n\}$ is the gradient of the functional j in the finite dimensional vector space Λ_H. This is what needs to be numerically computed.

As we explained before, we are going to concentrate on the term:

$$\frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).s_i.$$

4.2. State equations discretizations

We suppose here that $\varphi \in \Phi$ is given. It is not necessarily in the space Λ_H. The discretization described here is a priori independent from the design variable one.

We need to know a numerical approximation of the two functions u_φ and p_φ. The first one is the solution of the equation (8):

$$u_\varphi \in \mathbf{V}, \qquad a(\varphi; u_\varphi, v) = L(\varphi; v) \qquad \forall v \in \mathbf{V}$$

where $a(\varphi; u, v)$ is the energy functional associated to the shell equation, L is the energy of the external forces.

Such an equation, except for very exceptional cases cannot be explicitely solved. One has to use an approximation procedure. Notice that up to here, any kind of approximation could be used: finite elements, spectral methods or anything else. The only question we have to focus on is the following:

for given φ and ψ, which precision of approximation do we need for u_φ and p_φ in order to have a suitable approximation of $\frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi$?

This requires a look at the expression of $a(\varphi; u, v)$ which is given in section 3, equation (8), and in section 1, equations (3), (1) and (2). The expression of $\frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi$ is obtained from equation (8) by differentiation with respect to φ for fixed values of u and v, which are afterwards replaced by u_φ and p_φ.

As this concerns the partial differential of a with respect to φ, it is easy to realize that the accuracy of approximation needed is the same which is needed to get a good approximation of $a(\varphi; u_\varphi, p_\varphi)$. The full expression is given by an integral in which:

- u^1_φ and its first order derivatives, u^2_φ, its first and second order derivatives interfere,
- p^1_φ and its first order derivatives, p^2_φ, its first and second order derivatives interfere.

Moreover they interfere in products of two terms depending linearly in u (or p) with coefficients which are L^∞, because they involve φ and its derivatives up to the order 3, and in the differential ψ and its derivatives up to the order 3. As these two functions have been chosen in $W^{3,\infty}$, all coefficients are in L^∞. Therefore:

it is sufficient to use an approximation procedure for u_φ and p_φ which converges in $\mathbf{V} = [H^1_0(\hat\omega)]^2 \times H^2_0(\hat\omega)$.

REMARK 4.1 *In both shape and state equations approximations, we have here investigated only conforming methods. The space Λ_H has been chosen as a subspace of Λ, and we have considered approximations of u_φ and p_φ which converge in the space $\mathbf{V}$. This is to simplify the exposition. The approximations do not need to be conforming. A reasonable condition would be: φ, $\frac{\partial \varphi}{\partial \xi_1}$, $\frac{\partial \varphi}{\partial \xi_2}$, and all successive derivatives up to the order 3 must be approximated by functions $\varphi_{H,0}$, $\varphi_{H,1}$, $\varphi_{H,2}$,etc...which each converge in $L^\infty(\hat\omega)$ to the corresponding derivative. The same thing can be done for the displacement space but in this space the convergence has to work in $L^2(\hat\omega)$. An extensive description of this subject can be found in Lods (1992).*

Let us now look at the equation defining the adjoint state p_φ. It is the following (see equation (6)):

$$p_\varphi \in \mathbf{V}, \qquad a(\varphi; u_\varphi, w) = \frac{\partial J}{\partial v}(\varphi, u_\varphi).w \qquad \forall w \in \mathbf{V}.$$

The function u_φ which is in the right hand side is known only approximately. The left hand side of this equation is the same as the direct state equation one. It is of course convenient (and cheap) to use the same numerical procedure to solve both direct and adjoint state equations. The precision required on u_φ in order to get the good precision on p_φ depends of course on J. For instance, if:

$$J(\varphi, v) \quad = \quad \int_\omega ||\mathbf{v}(m)||^2 dm$$

$$= \int_{\hat{\omega}} \{ \| v^1(\hat{m})\ a_1(\hat{m}) + v^2(\hat{m})\ a_2(\hat{m}) \|^2 + [v^3(\hat{m})]^2 \}\ [g(\hat{m})]^{1/2}\ d\hat{m}.$$

it is sufficient to use for both a scheme which converges in $\mathbf{V}$. More generally, for any criterion which is lipschitz continuous with respect to $v \in \mathbf{V}$ it is the same, but for max criteria, one has to be careful. In one dimension, H_0^1 is continuously imbedded in $\mathcal{C}(\hat{\omega})$, but not in two dimensions. So, it would be necessary to use more precise approximation methods.

4.3. Computation of the discretized continuum gradient; software structure

Let us sum up the results obtained in this section about the numerical approximation of the differential of j.

- We have chosen a subvector space Λ_H of $W^{3,\infty}(\hat{\omega})$ of finite dimension, spanned by $\{s_i; i = 1, ...n\}$.
- The gradient we need to compute is the vector $\frac{dj}{d\varphi}(\varphi).s_i$ for $i = 1, ..., n$, with (see eq.(7)):
$$\frac{dj}{d\varphi}(\varphi).\psi = \frac{\partial J}{\partial \varphi}(\varphi; u_\varphi).\psi - \frac{\partial a}{\partial \varphi}(\varphi; u_\varphi, p_\varphi).\psi + \frac{\partial L}{\partial \varphi}(\varphi; p_\varphi).\psi.$$

In the preceding subsection, we have seen the basic principles required to get a suitable approximation of u_φ and p_φ. Supposing that we use a finite element procedure, notice that the master nodes used for the shape approximation can be used also in the finite element mesh, but it is strongly recommended to use a finer one for the finite elements. Actually, if the shape starts oscillating in the optimization algorithm, it is clear that several finite element nodes are necessary between the master nodes in order to get a reasonably good approximation of the direct and adjoint states.

Let us denote by h the mesh size for the approximation of the state equations. We get the approximate states $u_{\varphi,h}$ and $p_{\varphi,h}$, from which we hope to compute as an approximation of the gradient the vector:

$$\frac{\partial J}{\partial \varphi}(\varphi; u_{\varphi,h}).s_i - \frac{\partial a}{\partial \varphi}(\varphi; u_{\varphi,h}, p_{\varphi,h}).s_i + \frac{\partial L}{\partial \varphi}(\varphi; p_{\varphi,h}).s_i$$

which has been obtained replacing u_φ and p_φ by $u_{\varphi,h}$ and $p_{\varphi,h}$ in the expression of $\frac{dj}{d\varphi}(\varphi).s_i$.

Now we focus on the computation of the energy term $\frac{\partial a}{\partial \varphi}$. Looking at the formulas for the shell equations in section 1, we see that this is the integral of a heavy function that we denote by $G(\varphi, u, v, \psi)$, for $\varphi := \varphi$, $u := u_{\varphi,h}$, $v := p_{\varphi,h}$, $\psi := s_i$. This function G is obviously not piecewise polynomial, even if φ, u, v, ψ are. So a quadrature formula has to be used, using for instance Gauss points associated to the finite element mesh. We approximate then $\frac{dj}{d\varphi}(\varphi).\psi$ by:

$$[\frac{dj}{d\varphi}(\varphi).\psi]_h = \sum_{k=1}^{n} \lambda_k G(\varphi, u_{\varphi,h}, p_{\varphi,h}, \psi)(\hat{m}_k)$$

for a well fitted set of real coefficients λ_k and of points $(\hat{m}_k) \in \hat{\omega}$.

In a general setting of approximations (not necessarily finite elements), these points have to be compatible with the discontinuities of φ, ψ, $u_{\varphi,h}$ and $p_{\varphi,h}$ such that the quadrature formula be precise enough.

Now we are in a position to look at the structure of the code which has to be written in order to get a numerical approximation of the gradient of j. What we need is a subroutine which has as input parameters the functions φ, u, v and ψ, one point $\hat{m} \in \hat{\omega}$, and as output parametre the value of $G(\varphi, u, v, \psi)(\hat{m})$. At the step l of the optimization procedure, such a subroutine will be run for $\varphi = \varphi_l$, $u = u_{\varphi_l,h}$, $v = p_{\varphi_l,h}$, $\psi = s_i$, for $i = 1, ...n$.

In order to write this subroutine, we have to look at the shell energy functional which is given in section 1 and 8. The function G has a very complicated expression. Any extensive computation by hand has to be avoided. We suggest to write a modular software, using simple procedures calling previous ones, using the structure of composed functions with which $a(\varphi; u, v)$ and $\frac{\partial a}{\partial \varphi}$ are naturally expressed. This begins with pure geometric functions like:

$$\begin{aligned} a_\alpha(\hat{m}) &= \frac{\partial \varphi}{\partial \xi_\alpha}\,(\hat{m}), \\ [\frac{d\ a_\alpha}{d\ \varphi}.\psi](\hat{m}) &= \frac{\partial \psi}{\partial \xi_\alpha}(\hat{m}). \end{aligned}$$

To these functions, we associate the (Fortran) subroutines which have the functions φ and its derivatives, ψ and its derivatives, and $\hat{m}$ as input parameters, $a_\alpha(\hat{m})$ and $[\frac{d\ a_\alpha}{d\ \varphi}.\psi](\hat{m})$ as output parameters.

Then we need to differentiate $n(m)$, which requires to differentiate $||a_1 \times a_2||$. We notice that:

$$\begin{aligned} ||(a_1 \times a_2)(\hat{m})||^2 &= g(\hat{m}) \\ &= < a_1(\hat{m}), a_1(\hat{m}) >< a_2(\hat{m}), a_2(\hat{m}) > \\ &\quad - < a_1(\hat{m}), a_2(\hat{m}) >^2 \end{aligned}$$

so that:

$$\begin{aligned}[\frac{d}{d\,\varphi}(||a_1 \times a_2||^2).\psi](\hat{m}) &= [2 < \frac{d\,a_1}{d\,\varphi}.\psi, a_1 > +2 < \frac{d\,a_2}{d\,\varphi}.\psi, a_2 > \\ &+ < \frac{d\,a_1}{d\,\varphi}.\psi, a_2 > + < \frac{d\,a_2}{d\,\varphi}.\psi, a_1 >](\hat{m}).\end{aligned}$$

Then, we can use the following formulas, in which we denote $f(\varphi)'$ instead of $\frac{df}{d\varphi}(\varphi).\psi$:

- $||a_1 \times a_2|| = \sqrt{||a_1 \times a_2||^2}$, so that:
$$[||a_1 \times a_2||]' = \frac{[||a_1 \times a_2||^2]'}{2\sqrt{||a_1 \times a_2||^2}}.$$

Using the same approach, one can write a list of subroutines teaching the computer how to compute the value of the geometrical quantities associated to the suface ω at the point $\varphi(\hat{m})$. These give the coefficients of $\gamma(\varphi; u)$ and $\rho(\varphi; u)$. It is then easy to write a subroutine computing $\gamma(\varphi; u)(\hat{m})$ and $\rho(\varphi; u)(\hat{m})$ for given values of the input parameters φ and its derivatives, ψ and its derivatives, u and its derivatives, and $\hat{m}$. Then the function $G(\varphi, \psi, u, v)(\hat{m})$ can be written down.

We do not give here the extensive list of subroutines which are used in the case of a general shell. It can be found in Chenais (1987). We limit ourselves in this paper to the case of an arch that we treat in the following section.

5. Numerical treatment of arch equations

5.1. Arch equations

An arch is a shell which is invariant in one direction, say y. Both the geometry and the loading are, so the elastic displacement is also, invariant. It is sufficient to study the problem in the plane xOz. The midsurface of the shell reduces to a curve in this plane. We limit ourselves to the case of a curve which has an equation $z = \phi(x)$ defined on $[0, l]$, and such that $\Phi(0) = \Phi(l) = 0$. In this case it is convenient to decompose the displacement vector $\mathbf{u}$ in the local basis made of the *unit* tangent and normal vectors, say $\mathbf{t}(x)$ and $\mathbf{n}(x)$. So:

$$\mathbf{u}(x) = u^1(x)\mathbf{t}(x) + u^3(x)\mathbf{n}(x).$$

Notice that very often arch equations are written parametrizing the midcurve with its curvilinear abscissa. We do not do this because we are going to move this curve, and we need the parameter to be in a fixed real interval. As a matter of fact, using the curvilinear abscissa would correspond to the use of ω in the full shell equation, the use of $x \in [0, l]$, with a fixed value for l corresponding to $\hat{\omega}$.

In this plane case, all two–dimensional tensors fields reduce to one dimension. They are just real functions of x.

Let us give the equation defining $u = (u^1, u^3)$. In the sequel, we will denote $f^\bullet$ for $\frac{df}{dx}(x)$.

5.1.1. The geometrical quantities

- $\Lambda = W^{3,\infty}(]0, l[)$
- $S(\Phi) = \sqrt{1 + \Phi^{\bullet 2}}$ (which we often denote by S)
- $\mathbf{t} = \frac{1}{S}(1, \Phi^\bullet)$, $\qquad \mathbf{n} = \frac{1}{S}(-\Phi^\bullet, 1)$
- $\frac{1}{R} = -\frac{\Phi^{\bullet\bullet}}{S^3}$

5.1.2. The kinematical quantities

- $\mathbf{V} = H_0^1(]0, l[) \times H_0^2(]0, l[)$ (corresponding to a clamped arch)
- For any $\Phi \in \Lambda$ and any $u \in \mathbf{V}$:
$$\gamma(\Phi; u) = \frac{1}{S}u^{1\bullet} + \frac{u^3}{R}, \;\; \theta(\Phi; u) = \frac{u^1}{R} - \frac{1}{S}u^{3\bullet}, \;\; \rho(\Phi; u) = \frac{1}{S}[\theta(\Phi; u)]^\bullet$$
- For any $\Phi \in \Lambda$, any u and $v \in \mathbf{V}$:
$$a_m(\Phi; u, v) = \tfrac{Eb}{1-\nu^2} \int_0^l [\gamma(\Phi; u)\gamma(\Phi; v) \; S(\Phi)](x)dx$$
$$a_b(\Phi; u, v) = \tfrac{Eb}{1-\nu^2} \int_0^l [\rho(\Phi; u)\rho(\Phi; v) \; S(\Phi)](x)dx$$
$$a(\Phi; u, v) = 2ea_m(\Phi; u, v) + 2\tfrac{e^3}{3}a_b(\Phi; u, v)$$
where E is the Young modulus of the material, ν the Poisson ratio, e the thickness of the arch, b its depth.

Like in the previous section, we denote:

$$\frac{\partial a}{\partial \Phi}(\Phi; u, v).\psi = \int_0^l G(\Phi, u, v, \psi)(x) \; dx$$

5.1.3. The loading

We give here as an example the explicit formula for the selfweight load:

$$L(\Phi; v) = -\int_0^l \rho e(\Phi^\bullet S v^1 + v^3)(x)dx.$$

5.2. The optimal control problem

In order to illustrate the problem, we treat the example of the minimization of the functional $j(\Phi) = J(\Phi, u_\Phi)$ with:

$$J(\Phi, v) = \int_0^l ||\mathbf{v}(x)||^2 \; S(\Phi)(x) \; dx = \int_0^l [(v^1)^{\;2} + (v^3)^{\;2}](x) \; S(\Phi)(x) \; dx.$$

As we have seen in the previous section, we need to write a subroutine which has the functions Φ, ψ, u, v, their derivatives, and x as input parameters, and $G(\Phi, u, v, \psi)(x)$ as output parameter, where:

$$\frac{\partial a}{\partial \Phi}(\Phi; u, v).\psi = \int_0^l G(\Phi, u, v, \psi)(x)\ dx.$$

We do this with a list of simple subroutines computing the sequence of derivatives with respect to Φ which are needed in G. Looking at the functional $a(\Phi; u, v)$, we see that five geometrical quantities are concerned:

$$S(\Phi) = \sqrt{1 + \Phi^{\bullet 2}}, \quad \frac{1}{S}\ (\Phi), \quad [\frac{1}{S}\ (\Phi)]^{\bullet}, \quad \frac{1}{R}\ (\Phi) = -\frac{\Phi^{\bullet\bullet}}{S^3}, \quad [\frac{1}{R}(\Phi)]^{\bullet}$$

which need to be differentiated with respect to Φ. The value of $G(\Phi, u, v, \psi)(x)$ can be computed by the following list of subroutines (we use here f' for $\frac{df}{d\Phi}(\Phi).\psi$):

	input		output
1:	Φ, x	$\longrightarrow$	$S(\Phi)(x) = \sqrt{1 + \Phi^{\bullet 2}(x)}$
2:	Φ, x	$\longrightarrow$	$\frac{1}{S}(\Phi)(x) = \frac{1}{S(\Phi)(x)}$ (call subroutine 1)
3:	Φ, x	$\longrightarrow$	$\frac{1}{R}\ (\Phi)(x) = -\frac{\Phi^{\bullet\bullet}}{S^3}(x)$ (call subroutine 2)
4:	Φ, x	$\longrightarrow$	$[\frac{1}{S}]^{\bullet}\ (\Phi)(x) = -(\frac{\Phi^{\bullet}\Phi^{\bullet\bullet}}{S^3})(x)$
5:	Φ, x	$\longrightarrow$	$[\frac{1}{R}]^{\bullet}\ (\Phi)(x) = [-(\frac{1}{S})^3\Phi^{\bullet\bullet\bullet}$ $-3(\frac{1}{S})^2(\frac{1}{S})^{\bullet}\Phi^{\bullet\bullet}](x)$
6:	Φ, u, x	$\longrightarrow$	$\gamma(\Phi; u)(x) = [\frac{1}{S}(\Phi)\ u^{1\bullet} + \frac{1}{R}(\Phi)\ u^3](x)$
7:	Φ, u, x	$\longrightarrow$	$\theta(\Phi; u)(x) = [\frac{1}{R}(\Phi)\ u^1 - \frac{1}{S}(\Phi)\ u^{3\bullet}](x)$
8:	Φ, u, x	$\longrightarrow$	$\rho(\Phi; u)(x) = \frac{1}{S}\ [\theta(\Phi; u)]^{\bullet}(x)$
9:	Φ, ψ, x	$\longrightarrow$	$S'(x) = \frac{\Phi'\psi'}{S}(x)$
10:	Φ, ψ, x	$\longrightarrow$	$(\frac{1}{S})'(x) = [-(\frac{1}{S})^2 S'](x)$
11:	Φ, ψ, x	$\longrightarrow$	$(\frac{1}{R})'(x) = [-(\frac{1}{S})^3\psi^{\bullet\bullet} - 3(\frac{1}{S})^2(\frac{1}{S})'\Phi^{\bullet\bullet}](x)$
12:	Φ, ψ, x	$\longrightarrow$	$[(\frac{1}{S})^{\bullet}]'(x) = [-(\frac{1}{S})^3\psi^{\bullet}\Phi^{\bullet\bullet} - (\frac{1}{S})^3\Phi^{\bullet}\psi^{\bullet\bullet}$ $-3(\frac{1}{S})^2(\frac{1}{S})'\varphi^{\bullet}\varphi^{\bullet\bullet}](x)$

$$\begin{array}{lll}
13: & \Phi,\psi,x \longrightarrow & [(\frac{1}{R})^{\bullet}]'(x) = [-(\frac{1}{S})^3\psi^{\bullet\bullet\bullet} - 3(\frac{1}{S})^2(\frac{1}{S})'\Phi^{\bullet\bullet\bullet} \\
 & & -6(\frac{1}{S})(\frac{1}{S})'(\frac{1}{S})^{\bullet}\Phi^{\bullet\bullet} - 3(\frac{1}{S})^2\{(\frac{1}{S})^{\bullet}\}'\Phi^{\bullet\bullet} \\
 & & -3(\frac{1}{S})^2(\frac{1}{S})^{\bullet}\psi^{\bullet\bullet}](x)
\end{array}$$

$$\begin{array}{lll}
14: & \Phi,\psi,u,x \longrightarrow & \gamma'(u)(x) = [(\frac{1}{S})'\ u^{1\bullet} + (\frac{1}{R})'\ u^3](x) \\
 & & \text{(where we have denoted } \gamma'(u)(x) \\
 & & \text{for } [\frac{\partial}{\partial\Phi}\gamma(\Phi;u).\psi](x))
\end{array}$$

$$\begin{array}{lll}
15: & \Phi,\psi,u,x \longrightarrow & \theta'(u)(x) = [(\frac{1}{R})'\ u^1 - (\frac{1}{S})'\ u^{3\bullet}](x) \\
 & & \text{(where we have denoted } \theta'(u)(x) \\
 & & \text{for } [\frac{\partial}{\partial\Phi}\theta(\Phi;u).\psi](x))
\end{array}$$

$$\begin{array}{lll}
16: & \Phi,\psi,u,x \longrightarrow & \rho'(u)(x) = [(\frac{1}{S})'\ \theta(u) - \frac{1}{S}\ \theta'(u)](x) \\
 & & \text{(where we have denoted } \rho'(u)(x) \\
 & & \text{for } [\frac{\partial}{\partial\Phi}\rho(\Phi;u).\psi](x))
\end{array}$$

$$\begin{array}{lll}
17: & \Phi,\psi,u,v,x \longrightarrow & G(\Phi,u,v,\psi) = \\
 & & \frac{E}{1-\nu^2}[\gamma'(u)\gamma(v) + \gamma(u)\gamma'(v)](x)S(x) \\
 & & +\frac{E}{1-\nu^2}[\rho'(u)\rho(v) + \rho(u)\rho'(v)](x) \\
 & & +\frac{E}{1-\nu^2}[\gamma(u)\gamma(v) + \rho(u)\rho(v)](x)S'(x).
\end{array}$$

This list of subroutines builds the energy term in the gradient of j. In order to give the complete gradient for one example, we give the formulas which are needed in the gradient of j for the examples cited in this section: the loading is the selfweight:

$$L(\Phi;v) = -\int_0^l \rho e(\Phi^{\bullet} S v^1 + v^3)(x)dx,$$

and the functional we minimize is:

$$J(\Phi,v) = \int_0^l ||\mathbf{v}(x)||^2\ S(\Phi)(x)\ dx = \int_0^l [(v^1)\ ^2 + (v^3)\ ^2](x)\ S(\Phi)(x)\ dx.$$

We have:

$$\frac{\partial J}{\partial\Phi}(\Phi;v).\psi = \int_0^l [(v^1)\ ^2 + (v^3)\ ^2](x)\ [S(\Phi)]'(x)\ dx$$

and:

$$\frac{\partial L}{\partial\Phi}(\Phi;v).\psi = -\int_0^l \rho e\{[\psi^{\bullet} S\ + \Phi^{\bullet} S']v^1\}(x)\ dx.$$

Remembering that, as we have seen in theorem 2.2,

$$\frac{dj}{d\varphi}(\varphi).\psi = \frac{\partial J}{\partial\varphi}(\varphi; u_\varphi).\psi - \frac{\partial a}{\partial\varphi}(\varphi; u_\varphi, p_\varphi).\psi + \frac{\partial}{\partial\varphi}(\varphi; p_\varphi).\psi, \tag{9}$$

and proceeding as described before (subsection 4.3), we write:

$$\frac{dj}{d\varphi}(\varphi).\psi = \int_0^l F(\Phi, u_\Phi, p_\Phi, \psi)(x)\; dx.$$

The function F is the sum of the function G described below and of a similar function, say H, corresponding to the two other terms in $\frac{dj}{d\varphi}(\varphi).\psi$, which can be numerically computed by the following procedures:

	input		output
18:	Φ, ψ, v, x	$\longrightarrow$	$H_1(\Phi, v, \psi)(x) = \rho e\{[\psi^\bullet S \; + \Phi^\bullet S']v^1\}(x)$
19:	Φ, ψ, v, x	$\longrightarrow$	$H_2(\Phi, v, \psi)(x) = [(v^1)\,^2 + (v^3)\,^2](x)\; [S(\Phi)]'(x)$
20:	Φ, ψ, v, x	$\longrightarrow$	$H(\Phi, v, \psi)(x) = H_1(\Phi, v, \psi)(x) + H_2(\Phi, v, \psi)(x)$
21:	Φ, ψ, u, v, x	$\longrightarrow$	$F(\Phi, u, v, \psi)(x) \; =$ $G(\Phi, u, v, \psi)(x) \; + \; H(\Phi, v, \psi)(x)$

Now, the computation of the gradient of $j(\Phi)$ is straightforward. It is done in the following steps:

1. Choose a finite dimensional subspace of $W^{3,\infty}(]0, l[)$, and a basis $\{s_i; i = 1, ..., n\}$,
2. choose an approximation procedure solving the arch state equation, use it to get $u_{\Phi,h}$ and $p_{\Phi,h}$ and their derivatives,
3. choose an integration scheme which is compatible with points 1 and 2 (for instance take the Gauss points which likely have been used in point 2); this gives two lists $\{x_k\}$ and $\{\lambda_k\}$,
4. using the previous list of subroutines, compute:
$$[\frac{dj}{d\varphi}(\varphi).s_i]_h \; = \; \sum_{k=1}^{k=K} \lambda_k F(\Phi, u_{\Phi,h}, p_{\Phi,h}, s_i)(x_k).$$
These are the components of the discretized gradient of $j(\Phi)$.

6. Numerical results

In this section we give numerical results for the optimization of the shape first of an arch, then of an axisymmetric shell. We only optimize the shape of the midsurface, we do not change the thickness.

6.1. Case of an arch

The numerical experiments we show here have been performed by A.Habbal. They are published in details in Habbal (1990).

6.1.1. The continuous mechanical and optimization problem

The arch model has been described in the previous section. The equations are given in section 5.1. The numerical parameters are chosen as follows (in S.I. unities):

- $E = 10^6$ (Young modulus)
- $e = 0.50$ (half thickness of the arch)
- $l = 1$ (x belongs to $[0, 1]$)
- the Poisson ratio is approximated by 0.

The loading is given by a density $G(x)$ (on the horizontal axis). The expression of its work in the virtual displacement v is:

$$L(\Phi; v) = \int_0^l \{-G[\frac{1}{S}v^1 + \frac{\Phi^\bullet}{S}v^3]\}(x)\,dx.$$

Notice that for the self-weight the density is constant per constant ds, differential element of curvilinear abscissa (for a constant thickness). A constant density G per constant dx is commonly called the snow-loading.

Two examples are treated:

$$\begin{aligned} G_1(x) &= -10 \qquad \forall x \in [0, 1], \\ G_2(x) &= \begin{cases} -10 & \forall x \in [\,0,\ 0.20\,[\,\cup\,]\,0.80, 1\,], \\ 0 & \forall x \in [\,0.20,\ 0.80\,]. \end{cases} \end{aligned}$$

The functionals to be minimized of two types are chosen :

- $J_1(\Phi, v) = \int_\omega ||\mathbf{v}(s)||^2\,ds$
  $= \int_0^1 ||\mathbf{v}(x)||^2\,S(\Phi)(x)\,dx$
  $= \int_0^1 [(v^1)^{\ 2} + (v^3)^{\ 2}](x)\,S(\Phi)(x)\,dx,$
- $J_2(\Phi, v) = Max_{x\in[0,\ 1]}||\mathbf{v}(x)||^2 = Max_{x\in[0,\ 1]}[(v^1)^{\ 2} + (v^3)^{\ 2}](x).$

Notice that the second one is not Fréchet differentiable, but it fits in the setting of the nonsmooth optimization techniques. We do not give any description of these techniques here. They can be found in Zowe (1985) and Lémarechal (1980). Their application to the arch case is given in Habbal (1990).

6.1.2. The discretizations

The design variable space $W^{3,\infty}(]0,1[)$ is approximated by a finite dimensional subspace. Spline spaces have been chosen. Cubic splines would a priori be sufficient, they belong to the space $W^{3,\infty}(]0,1[)$. Actually, fifth degree splines have been chosen to insure that the solution of the state equation is more regular than $\mathrm{H}_0^1(]0,1[) \times H_0^2(]0,1[)$ because the convergence of the finite elements used to solve this equation is only insured if the solution has the regularity $\mathrm{H}^2(]0,1[) \times H^3(]0,1[)$. Cubic splines are not sufficient to insure it.

Nodes are chosen, which are the master nodes, on which the splines are based. The experiments have been done with few master nodes. We had a priori no reason to choose more. Further experiments should be done in order to follow the influence of their number on the optimization procedure. The following five nodes have been chosen:

$$X_0 = 0, \quad X_1 = 0.20, \quad X_2 = 0.50, \quad X_3 = 0.80, \quad X_4 = 1.$$

The end nodes do not move in the optimization procedure, so that there are three discrete design variables.

The state equation is discretized with a finite element procedure. The procedure which has been chosen consists in the approximation of the arch by straight beam elements. The angle between the beams is assumed to be the same before and after loading. This scheme is completely analysed in Bernadou, Ducatel (1983). It is proved in this article that this procedure converges in $H_0^1(]0,1[) \times H_0^2(]0,1[)$ at the order 1 when the max length h of the beams goes to zero provided that the solution of the continuous problem has the regularity $H^2(]0,1[) \times H^3(]0,1[)$. The choice of fifth degree splines insures this regularity, if the loading is in $L^\infty(]0,1[)$ (see Habbal, 1990). This is also why the experiments on concentrated loads have not been done with pointwise loads. The choice has been to make the experiments in secure conditions. As the conditions which are used are only sufficient, it would be of interest to try to violate them and see what happens.

The number of finite element nodes between the master nodes has to be sufficient so that the analysis results be trustable, even if the master nodes start oscillating along the optimization procedure.

In the following results 40 to 80 finite element nodes have been chosen. Close to the optimum, 80 are used.

REMARK 6.1 *It is the moment to make a remark about what would happen if we would couple the optimization of the midcurve and the optimization of the thickness. From a differentiability point of view, it is trivial. The differential with respect to the thickness is straightforward, as well as its discretization, because it can be chosen piecewise constant. Yet, numerical difficulties are expected. First, experiments are described in Cheng, Olhoff (1982) on plates showing that if a* *lower bound is not prescribed to the thickness, there seems to be no existence of a minimum. Something like a composite or homogenized material could appear. There is another difficulty: the usual finite element procedures computing arches (and likely shells) lock when the thickness approaches zero as soon as there is a nonzero curvature, and the thickness for which the locking appears depends on the shape of the midcurve (see Habbal, 1990; Habbal, Chenais, 1992; Chenais, Zerner, 1994). So, in order to couple these two optimizations, one has to be extremely careful about the meaning of the finite element results. As a matter of fact, finite element procedures are now known which converge* uniformly with respect to the thickness *(see Kikuchi, 1982; Arnold, 1981; Reddy, 1988; Chenais, Paumier, 1994). Those should certainly be used even if they are more expensive.*

The approximation of integrals is done with 2 Gauss points per finite element.

Different initial shapes have been chosen in the optimization procedure. As no uniqueness and convergence result is known because of the lack of convexity, this is a way to validate the results. The three following initial shapes give the same optimum:

1. $\Phi_1^0 = x(1-x)$ (parabola),
2. Φ_2^0 is a slightly curved beam, (for a straight beam the gradient is equal to 0),
3. Φ_3^0 is a nonsymmetric spline curve.

The optimizer which has been used is the code M1FC1 which is a nonsmooth optimizer, written by Claude Lémarechal at INRIA.

6.1.3. The arch results

We show the results in figures. As we explained before, three initial shapes have been chosen for each experiment. We show only one case. The other ones give the same result with less than one per cent error.

The figures show the following results:

- – Figure 1.(a) shows the initial and final shape for the uniform snow–loading (density G_1) and the max functional (functional J_2).

  – Figure 1.(b) shows the evolution of the cost functional in this situation. It is noticeable that it decreases very rapidly at the beginning. Then it decreases slowly although the change in shape can clearly be seen. With the other initial shapes, the difference is very slight. After one, may be two iterations, the algorithm works within the same range of values.
- – Figure 2.(a) shows the initial and final shape for the uniform snow–loading (density G_1) and the integral functional (functional J_1).

  – Figure 2.(b) shows the evolution of the cost functional in this situation. The same remarks can be done as before.

- - Figure 3.(a) shows the initial and final shape for the concentrated loading (density G_2) and the *max* functional (functional J_2).
  - Figure 3.(b) shows the evolution of the cost functional in this situation.
- - Figure 4.(a) shows the initial and final shape for the concentrated loading (density G_2) and the integral functional (functional J_1).
  - Figure 4.(b) shows the evolution of the cost functional in this situation.
- Figure 5 shows the norm of the displacement field on both the optimal shapes for the concentrated loading. It is quite clear that one is better from the *max* point of vue, the other for the integral criterion. We do not show the same picture for the uniform loading because no difference can be seen. From theoretical considerations this was clearly expected. The displacement must be poorly smooth if we want to see a difference. It is theoretically known that if the working conditions are smooth, the solution is also smooth .

6.2. An example of axisymmetric shell

This example has been performed by Sabine Moriano (see Moriano, 1988). The subject is the optimization of the shape of cooling towers for electric plans. It has been realized for the French Electricity Company. The static structural behaviour of the tower has only been considered. It would be interesting to consider vibration and buckling conditions, and also to couple the structural equations with heat equations because these towers are chimneys which have to cool down steam. A classical shape is shown in Fig.6.

6.2.1. The continuous mechanical and optimization problem

The general shell equations are described in section 1. In the case we are treating here, the mapping φ depending on two variables defines an axisymmetric surface. This surface is represented in a classical system of axes x, y, z, in polar coordinates. So

$$\omega = \{m = \varphi(z, \theta);\ z \in (0, L),\ \theta \in (0, 2\pi)\}$$

with:

$$\varphi(z, \theta) = \begin{pmatrix} r(z)\cos\theta \\ r(z)\sin\theta \\ z \end{pmatrix}.$$

So the real unkown is the function $r(z)$ of one variable, defined on $(0, L)$. The shell equations written in terms of this unknown function $r(z)$ can be found in Moriano (1988). It can be seen from these equations that $r(z)$ needs to be in the space $W^{3,\infty}(]0, L[)$.

The geometrical and mechanical parameters are the following:

• $L = 200\ m$	(height)
• $e = 0.70\ m$	(thickness)
• $E = 0.38\ 10^{11}$	(Young modulus)
• $\nu = 0.2$	(Poisson ratio).

The loading is the self–weight with a fixed constant density of material. The tower is clamped at its basis.

The functionals to be minimized are the two following:

- $J_1(r; v)$ is the work of external forces:
  $J_1(r, v) = L(r; v)$.
- $J_2(r; v) = \int_\omega ||\mathbf{v}||^2\ (m)\ dm$.

Constraints are required on the design variable which are:

$$45m \leq r(z) \leq 80m.$$

6.2.2. The discretizations

The design variable $r(z)$ is in the space $W^{3,\infty}(]0, L[)$. As in the arch case, it is approximated by splines. 7 master nodes are chosen on the z axis. They are equidistant, one is equal to 0, another one is equal to 200. All 7 nodes are allowed to move in the optimization procedure.

In the state equation, a Fourier series decomposition is used to deal with the variable θ. It is well-known that the different harmonics are decoupled (this is true for the state equation but not for the optimization problem). Each harmonic gives a one dimensional linear variational problem, posed on the space $H^1(]0, L[) \times H^2(]0, L[)$. These problems are solved with a finite element procedure. The code, which is called NOLICO, has been written by the French Electric Company. It is based on the following standard scheme: there is no approximation of the geometry, the equation of the midcurve is given to the computer. Then the displacements are approximated by $(P^1,\ P^3)$ elements.

The approximations of integrals are taken like in the finite element code. Two Gauss points are taken in each finite element.

Different initial shapes are taken, for the same reason than in the arch experiments. Three linear shapes are chosen, namely:

1. $r_0(0) =\ 65\ m,$	$r_0(L) =\ 65\ m,$
2. $r_0(0) =\ 80\ m,$	$r_0(L) =\ 50\ m,$
3. $r_0(0) =\ 50\ m,$	$r_0(L) =\ 80\ m.$

6.2.3. The cooling towers results

As in the arch case, we show the results in figures. The three different initial shapes give the same results, so we show only one.

- Figure 6 shows the general aspect of a cooling tower with its parametrization.
- Figure 7 shows the initial and final shape for the functional J_1 (work of external forces).
- Figure 8 shows the evolution of the value of the criterion along the iterations.
- Figure 9 shows the initial and final shape for the functional J_2 (integral functional).
- Figure 10 shows the evolution of the value of the criterion along the iterations.

We notice that in both examples the constraints become active at the upper part of the tower. As a matter of fact, from a strictly structural point of vue, it seems that the shape would like to close at the top point to make a dome. Of course this would not make a good chimney!

7. Conclusion

We have shown how to work with linear elastic equations of shells in order to optimize the shape of the shell with respect to a given criterion. The technique which has been used relies on descent methods. We have shown how to differentiate the equations and then use discretizations in order to get an approximation of the gradient of the functional to minimize. Two explicit examples have been treated: arches and axisymmetric shells. Let us notice that the aim of this paper is to demonstrate the feasibility of the method. The examples have not been seriously investigated from a mechanical point of vue. This would be interesting to do now that we know that the method looks efficient.

Fig. 1(a) uniform snow loading, cost = max||u(x)||

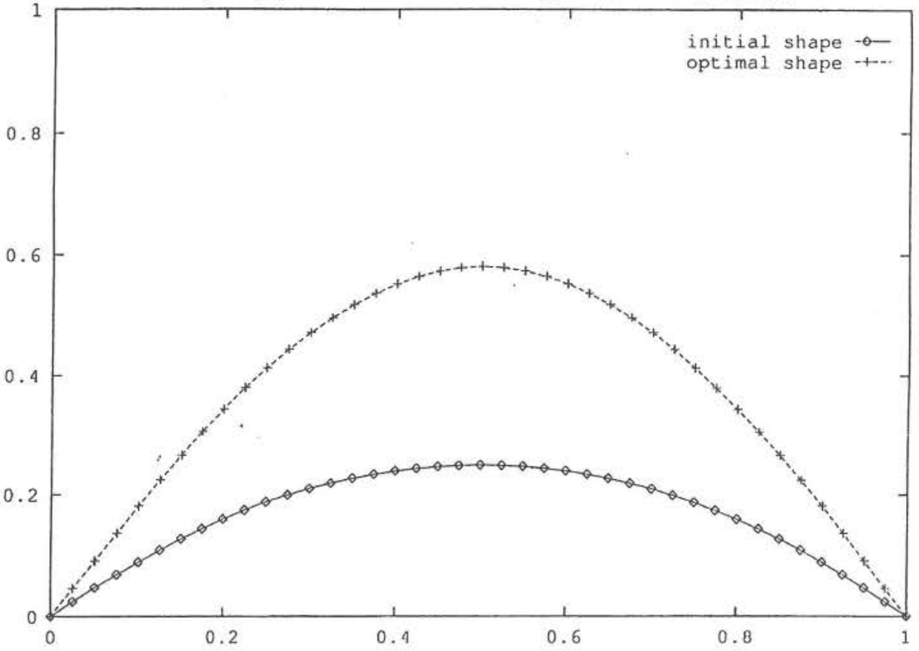
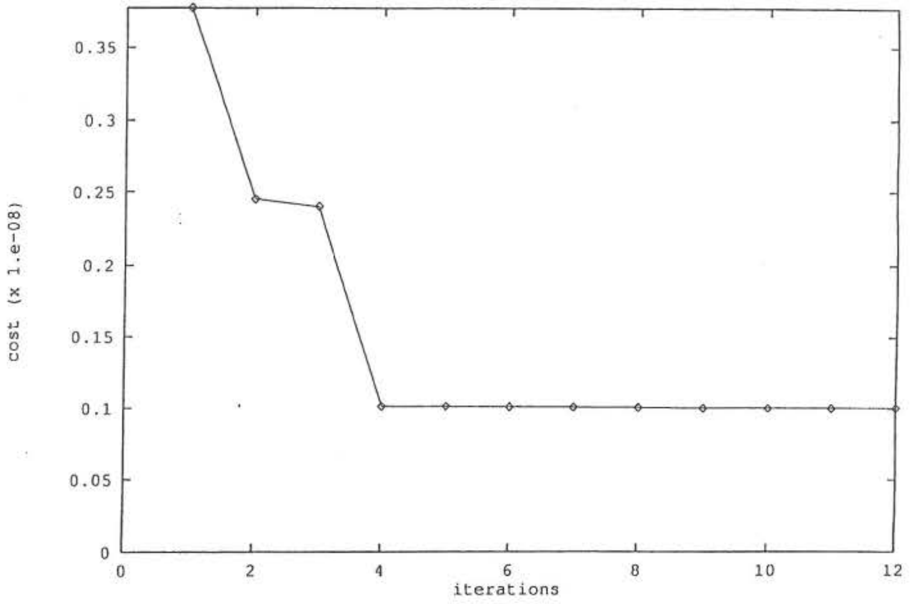


Fig. 1(b)



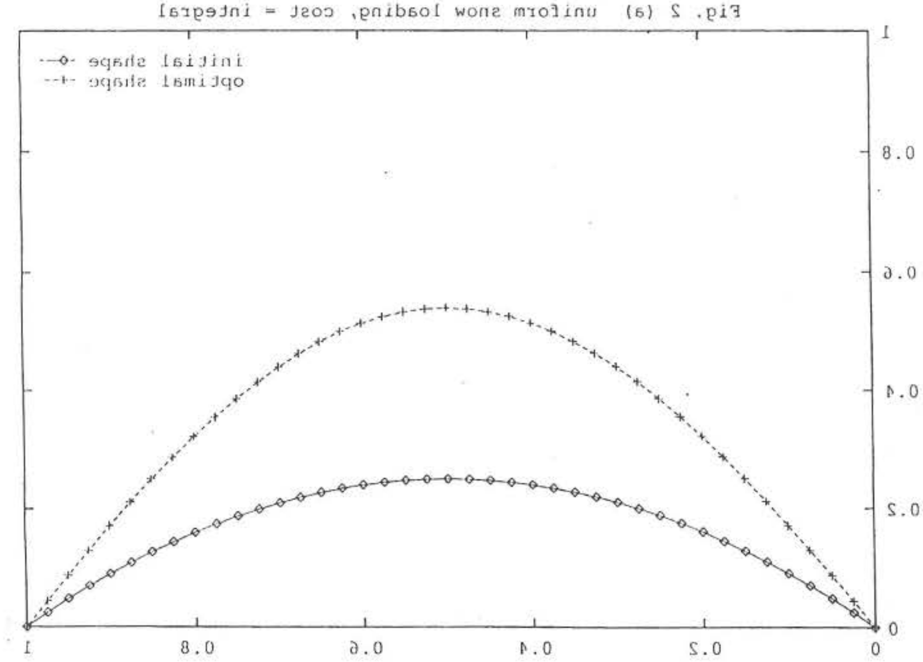


Fig. 2 (a) uniform snow loading, cost = integral
initial shape
optimal shape
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1

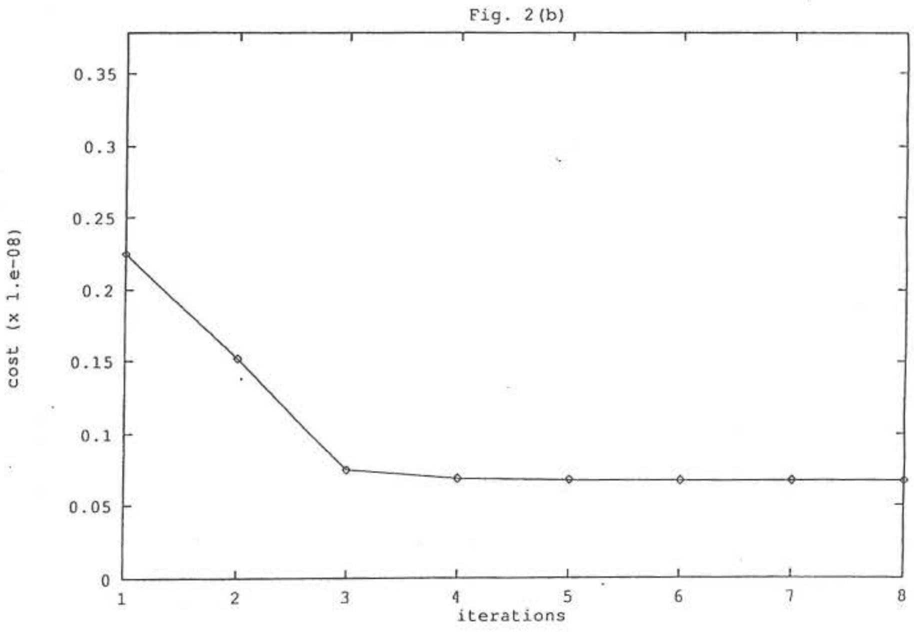


Fig. 2(b)
cost (x 1.e-08)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
1
2
3
4
5
6
7
8
iterations

Fig. 3(a) concentrated loading, cost=max||u(x)||

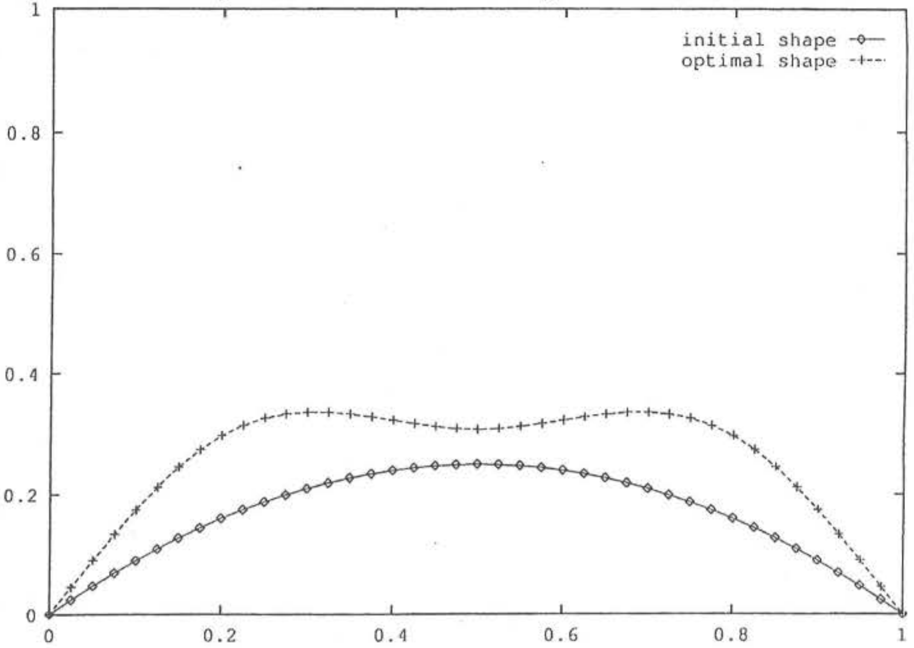


Fig. 3(b)

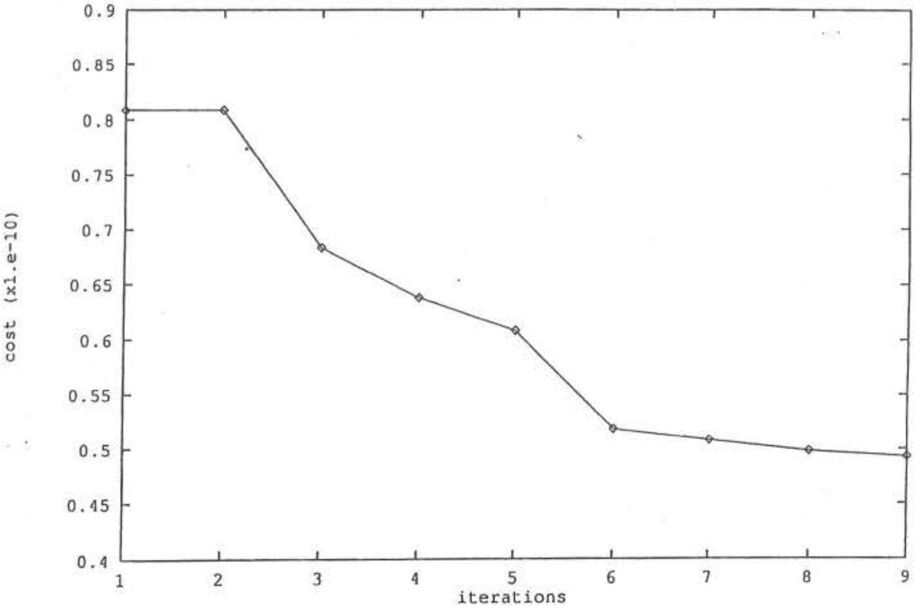


Fig. 4(a) concentrated loading, cost = integral

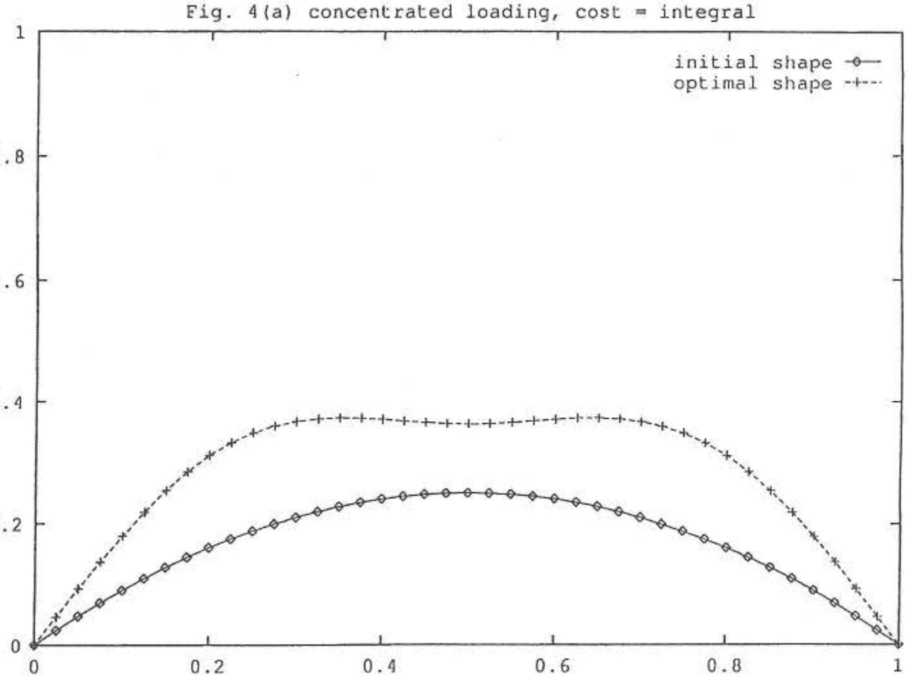


Fig. 4(b)

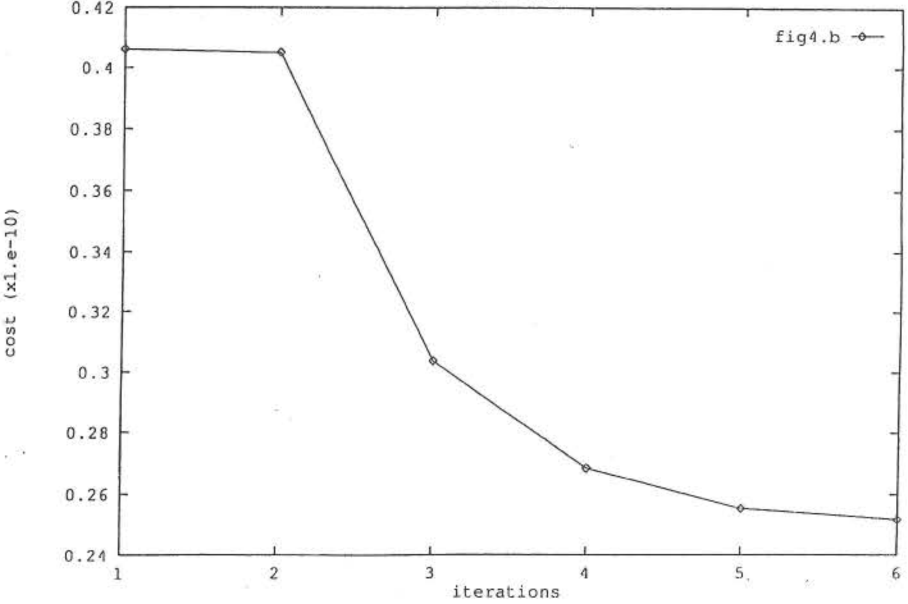
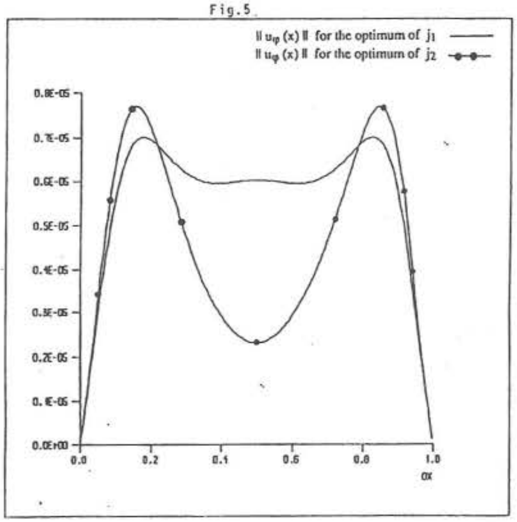


Fig.5


$\| u_\varphi (x) \|$ for the optimum of j_1
$\| u_\varphi (x) \|$ for the optimum of j_2
0.8E-05
0.7E-05
0.6E-05
0.5E-05
0.4E-05
0.3E-05
0.2E-05
0.1E-05
0.0E+00
0.0
0.2
0.4
0.6
0.8
1.0
OX

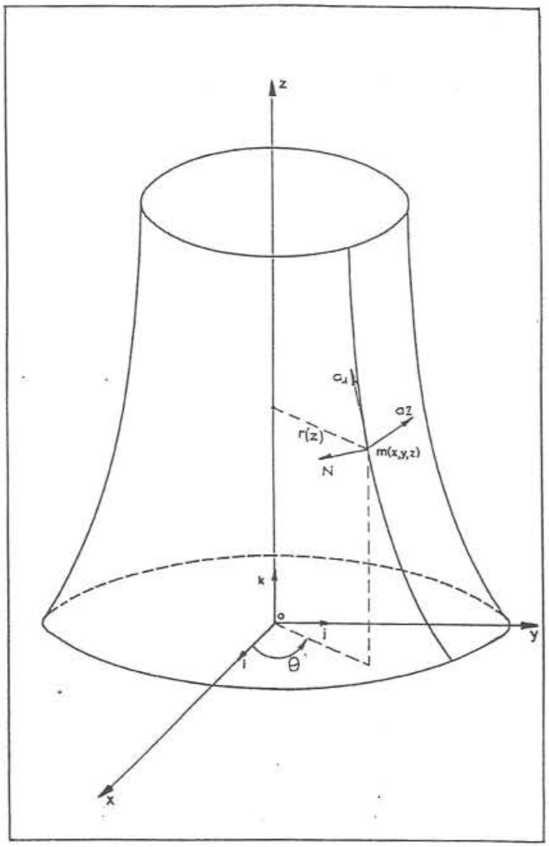

z
a_r
a_z
r(z)
m(x,y,z)
N
k
o
j
y
i
θ
x

fig. 6

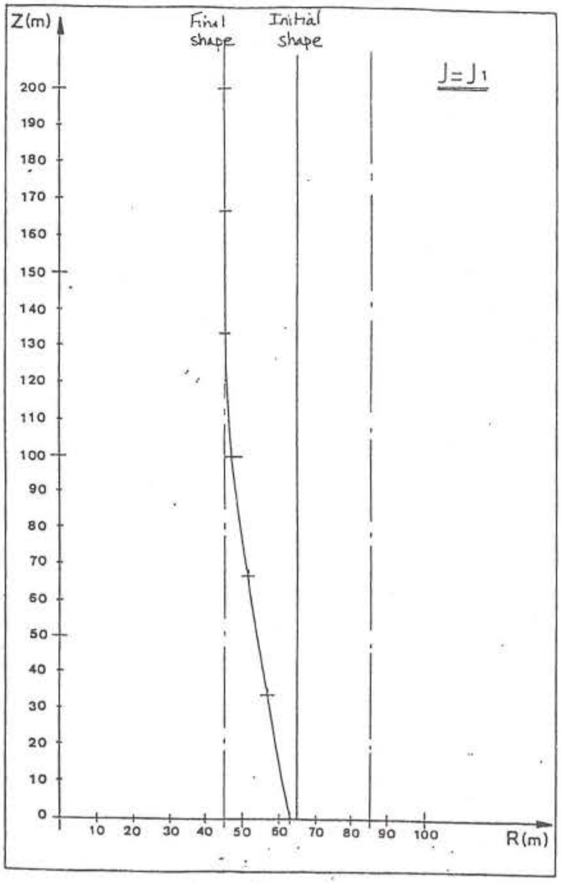

Z(m)
Final shape
Initial shape
J=J1
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90
100
R(m)

fig. 7

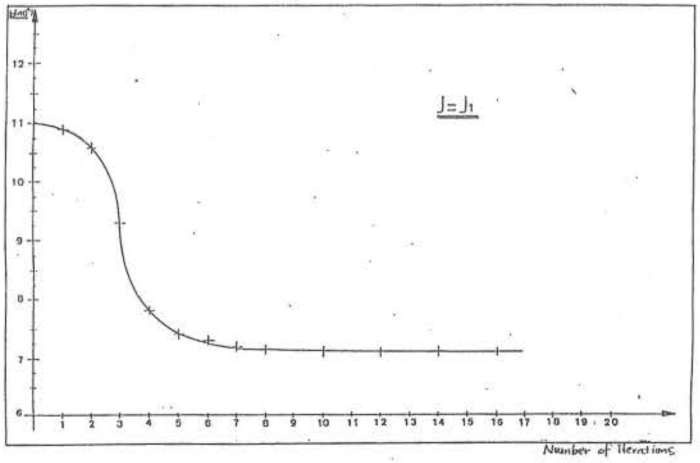

J=J1
12
11
10
9
8
7
6
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Number of Iterations

fig. 8

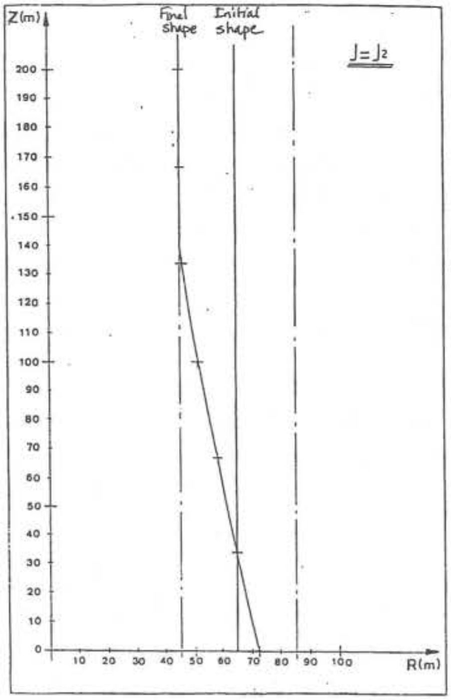

Z(m)
Final shape
Initial shape
J=J2
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90
100
R(m)

fig. 9

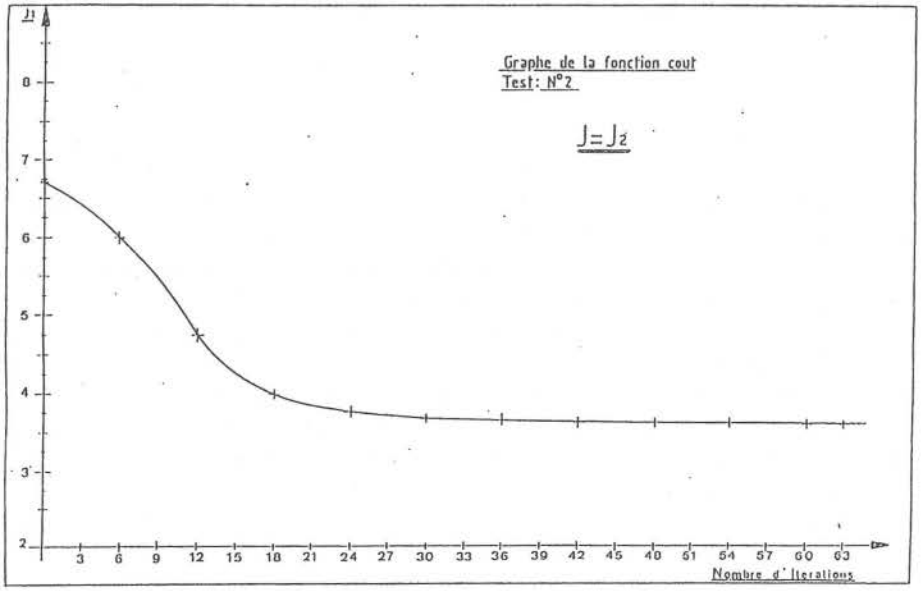

J1
Graphe de la fonction cout
Test: N°2
J=J2
8
7
6
5
4
3
2
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
Nombre d' Iterations

Fig. 10

References

Arnold D.N. (1981) Discretization by finite elements of a model parameter dependent problem, *Num. Math.* **37**, 405–421.

Bernadou M., Ducatel Y.(1983) Approximation of general arch problems by straight beam elements, *Numerische Mathematik* **40**, 1–29.

Brezis H. (1983) *Analyse Fonctionnelle*, Masson.

Chenais D. (1987) Optimal design of midsurface of shells: differentiability proof and sensitivity computation, *Appplied Math. and Optimization* **16**, 93–133.

Chenais D., Paumier J.C. (1994) On the locking phenomenon for a class of elliptic problems, to be published.

Chenais D., Rousselet B., Benedict R. (1988) Design sensitivity for arch structures, *Journal of Opt. Theory and Applications* **58** (2), 225–239.

Chenais D., Zerner M. (1994) Numerical methods for elliptic boundary value problems with singular dependence on a small parameter, necessary conditions, to be published.

Cheng K.T., Olhoff N. (1982) Regularized formulation for optimal design of axisymmetric plates, *Int. Jour. of Solids and Structures* **18**, 153–169.

Ciarlet P.G. (1978) *The Finite Element Method for Elliptic Problems*, North–Holland.

Destuynder Ph.(1985) *Acta Applicandae Mathematicae*, Reidel, Dordrecht, **4**, p. 1563.

Habbal A. (1990) Optimisation non differentiable de forme d'arche, Thèse de l'Université de Nice.

Habbal A., Chenais D. (1992) Deterioration of a finite element method for arch structures when thickness goes to zero, *Numerische Mathematik* **62**, 321–341.

Kikuchi F. (1982) Accuracy of some finite element models for arch problems, *Computer Methods in Applied Mathematics and Engineering* **35**, 315–345.

Lemarechal C. (1980) Non differentiable optimization, *NonLinear Optiimization*, Dixon–Spedicato–Szegö (eds.), Birkauser, Basel, 149–199.

Lods V. (1992) Gradient discret et gradient continu discrétisé en contrôle optimal à paramètres distribués, These de l'Université de Nice (France).

Moriano S. (1988) Optimisation de forme de coques, thesis of the University of Nice (France).

Reedy B.D. (1988) Convergence of mixed finite element approximations for the shallow arch problem, *Num. Math.*, Vol. **53**, 687–699.

Rousselet B. (1982) Note on design differentiability of the static response of elastic structures, *Jour. Struct. Mech.* **10**, 353–358.

Zowe (1985) Non differentiable optimization, *Computational Mathematical Programming*, Springer Verlag, 323–356.