International Journal of Computational Vision and Biomechanics
Vol.1 No. 2 (July-December, 2015)

MUK PUBLICATIONS
Open Access Publisher

Dynamic Parametric Surface Deformation using Finite Elements based on B-splines

Manuel González-Hidalgo[1] & Arnau Mir, Gabriel Nicolau[2]

[1]*Computer Graphics and Vision Group, Maths and Computer Science Department, University of the Baleric Islands, Spain*

[2]*Mathematics and Computer Science Department, University of the Baleric Islands, Spain*

The study of the motion of deformable objects is one of the most important topics in computer animation, computer modelling, and so on. This paper presents a dynamic evolution model in order to deform parametric surfaces with linear complexity. The deformation model is based on an associated energy to one surface, that it check the shape of it. The associated variational formulation to the problem of minimize the energy functional is solved using the finite element method based on B-splines. The spatial discretization where these finites elements are defined and computed shows as a reduced number of control points is deformed instead of all the surface points, obtaining an efficient numerical scheme.

Keywords: *B-splines, Finite Elements, Variational Formulation, Dynamic Deformation Model*

1. INTRODUCTION

The deformation models include a large number of applications, and they have been used in fields as edge detection, computer animation, geometric modelling, and so on. In this work, a deformation model will be introduced that uses B-splines as finite elements. This theory was introduced by Höllig in [Höllig 2003]. In fact, we have used a variational formulation similar to the used one in [Cohen 1992] changing, among other things, the selected finite elements.

The most used finite elements for surfaces are triangles, squares, among others; with them, it is easy to make a mosaic that it fills all space, but this technique needs long computation time since we must use some big data structures to solve our problem. This data structure are bound by the quantity of surface points that we must to take to obtain a good approximation of the surface.

In this work[1], we have applied the finite elements method based on B-splines to solve numerically a partial differential equation problem. The advantage to use B-splines finite elements is that it combines the computational advantage of B-splines and standard mesh-based elements. Thus, we will obtain a data structure smaller than the obtained one using the usual finite elements.

This work is organized as follows. In section 2, we define the uniform B-splines used to introduce finite elements. First, we introduce one dimensional B-splines: we display the recurrence relation they verify and how to compute their derivatives. Next, we define the multivariate B-splines. The section finishes with the definition of a B-spline parametric surface. Section 3 is devoted to the model of surface deformation where we have applied the finite elements methods using B-splines. Once we have introduced the partial differential equation verified by the surface, we have solved it numerically in section 4. In this section, we have raised the variational formulation to follow with its spatial and temporal discretization where the finite elements B-splines are defined. We have shown that a subset of the control points are deformed instead of all the surface points. The model evolution has also been introduced. In the next section, several numerical computed deformations are displayed using the evolution model with different forces. In section 6, we have shown the efficiency of our method computing its computational cost. In section 7, we have studied how the deformation of the surface changes under the influence of first order parameters. Finally, some conclusions are exposed.

[1]This work is an extended version of (González-Hidalgo, Mir, and Nicolau 2006) presented in ComplMAGE'06 Conference

2. SPLINES

The B-splines are piecewise polynomial functions. It has been verified, with others approximations functions technics [Piegl 1997] that the polynomials provide a good local approximation for smooth functions. However, if we use large intervals, accuracy of the approximation can be very low, the exactitude of the approach could be very low and local changes have global influence. Therefore, it is natural to use piecewise polynomials, defined on a fine partition of the function domain. We have choosed B-splines as piecewise polynomial approximation because of its local support. This property reduces the computational cost of the model.

2.1 Splines Functions

Uniform B-splines can be defined in several ways [de Boor 1978] [Farin 1997] [Piegl 1997] [Höllig 2003]. In this work we have taken the definition given by Höllig in [Höllig 2003], which we describe next.

Definition 1. [Höllig 2003] An uniform B-spline of degree n, b^n, is defined by

$$b^n(x) = \int_{x-1}^{x} b^{n-1}(t)dt$$

starting with $b^0(x) = \begin{cases} 1, & x \in [0,1), \\ 0, & otherwise \end{cases}$

The previous definition is not adapted for numerical evaluations. In order to be able to evaluate the B-splines in a simple form and fast computationally, we can use the recurrence equation. This equation was given by De Boor [de Boor 1978] and Cox [Cox 1972], and it is a linear combination of smaller degree B-splines.

$$b^n(x) = \frac{x}{n} b^{n-1}(x) + \frac{(n+1-x)}{n} b^{n-1}(x-1) \quad (1)$$

In order to construct the finite elements bases, we will use a scaled and translated uniform B-spline. They are defined by transforming the standart uniform B-spline, b^n, to the grid $h\mathbb{Z} = \{..., -2h, h, 0, h, 2h,...\}$, where h is the scaled step.

Definition 2: The transformation for $h > 0$ and $k \in \mathbb{Z}$ is $b^n_{k,h}(x) = b^n\left(\frac{x}{h} - k\right)$. The support of this function is $[k, k + n+1)h$

In order to build a variational formulation of a differential equation problem, we will need the derivatives of the finite elements. From the definition 1 we obtain that the first order derivative of degree n B-spline is given by

$$\frac{d}{dx} b^n(x) = b^{n-1}(x) - b^{n-1}(x-1)$$

with $b^n(0) = 0$ [Höllig 2003]. If we apply the transformation given in definition 2.2, the first order derivative of the transformed B-spline is given by

$$\frac{d}{dx} b^n_{k,h}(x) = h^{-1}(b^{n-1}_{k,h}(x) - b^{n-1}_{k+1,h}(x)) \quad (2)$$

[Höllig 2003]. We also need the derivatives of any order. These ones are given by a linear combination of lower degree B-splines. The differentiation formula can be expressed in a compact form as follows.

Theorem 2.1: The m^{th} derivative of a degree n transformed B-spline following the definition 2.2 is given by the recurrence relation

$$\frac{d^m}{dx^m} b^n_{k,h}(x) = h^{-m} \sum_{i=0}^{m} (-1)^i \binom{m}{i} b^{n-m}_{k+i,h}(x) \quad (3)$$

Obviously, this equation has sense if $m \le n$ since in others cases the derivative is 0.

2.2 Multivariate B-Splines

There is no unique generalization of one dimensional B-splines. These generalizations differs in the underlying partition for the polynomial segments [de Boor, Höllig and Riemenschneider 1993] [Piegl 1997], [Höllig 2003].

A possibility is to form products of one dimensional B-splines, as described in the following construction. The N-variate B-spline of degree $\mathbf{n} = (n_1,...,n_N)$, of index $k = (k_1,...,k_N)$ and the space discretization $\mathbf{h} = (h_1,..., h_N)$ is defined as

$$B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(\mathbf{x}) = \prod_{i=1}^{N} b^{n_i}_{k_i,h_i}(x_i). \quad (4)$$

The support of this function is $\prod_{i=1}^{N} [k_i, k_i + n_i + 1)h_i$.

Applying basic properties of differential calculus and applying theorem 2.1, a compact expression for any partial derivative of multivariate B-spline can be obtained. Using theorem 2.1, the derivatives can be evaluated and they are obtained with a smaller computational cost, since less recurrences are applied.

In the next figure, the graph of several bidimensional B-splines are shown.

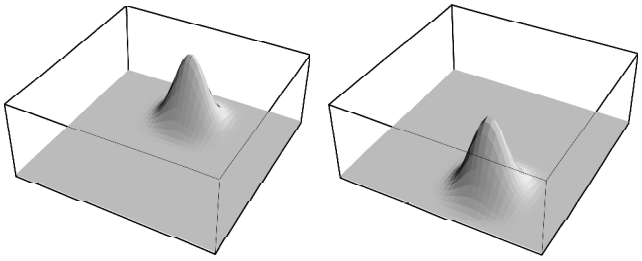


Figure 1: Bicubic B-splines with scaled step h = 1/6. Left: using a translation k = (4, 9). Right: using a translation k = (9, 0)

2.3 Parametric Surfaces with B-Splines

A parametric surface is defined as $S : \Omega \subset IR^2 \to IR^3$, $(u, v) \to S(u, v) = (x(u, v), y(u, v), z(u, v))$ with the necessary degree of differentiability [do Carmo 1976], where Ω is a bounded bidimensional subset. This surface will be B-spline if we can put it as a linear combination of bidimensional B-splines. That is,

$$S(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} P_{\mathbf{k}} B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(\mathbf{x}) \quad (5)$$

where $\mathbf{n} \in IN^2$, and $\mathbf{h} \in \mathbb{R}^2$ with positive coordinates. The coefficients $P_{\mathbf{k}} \in IR^3$ are called *control points* and they are the elements that determine the B-spline surface.

In order to be able to work with finite elements, we will need bases with a finite number of elements. The parametric surfaces we will use must have a bounded domain. Consequently, all $P_{\mathbf{k}}$ will be zero except a finite number of them. In order to find this set of control points, we must find the relevant B-splines. These ones fulfill $Sup(B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}) \bigcap \Omega \neq \emptyset$.

The relevant B-splines of our surface are determined by the spatial discretization, since the B-splines support depends on them. This problem will be addressed in the section of numerical resolution (section 4).

3. DEFORMATION MODEL. MINIMAL SURFACES

The deformation model is based on an associated energy to one surface, that it checks the shape of it.

The energy function is a non convex function with a local minimum. The goal is to achieve this minimum using an evolution model. This mininum depends on the initial surface and the used evolution model.

The associated energy functional, $E : F(S) \to IR$, $S \mapsto$ E(S), is defined as

$$E(S) = \int_\Omega \omega_{10}\left|\frac{\partial S}{\partial u}\right|^2 + \omega_{01}\left|\frac{\partial S}{\partial v}\right|^2 + \omega_{11}\left|\frac{\partial S}{\partial u \partial v}\right|^2 + \omega_{20}\left|\frac{\partial^2 S}{\partial u^2}\right|^2 + \omega_{02}\left|\frac{\partial^2 S}{\partial v^2}\right|^2 + \mathcal{P}(S(u,v))dudv$$

[Terzopoulos 1986], [Cohen 1992], [Montagnat, Delingette and Ayache 2001], where $\mathcal{P}$ is a potential of the forces that works on the surface. Using the equations of Euler-Lagrange, it can be proved [Cohen 1992] that an energy local minimum must satisfy:

$$-\omega_{10}\frac{\partial^2 S}{\partial u^2} - \omega_{01}\frac{\partial^2 S}{\partial v^2} + 2\omega_{11}\frac{\partial^4 S}{\partial u^2 \partial v^2} + \omega_{20}\frac{\partial^4 S}{\partial u^4} + \omega_{02}\frac{\partial^4 S}{\partial v^4} = -\nabla\mathcal{P}(S(u,v)) + \text{boundary conditions} \quad (6)$$

The surface domain is $\Omega = [0, 1]^2$ and the boundary conditions are: $S(u, 0) = (u, 0, 0)$, $S(u, 1) = (u, 1, 0)$, $S(0, v) = (0, v, 0)$, $S(1, v) = (1, v, 0)$.

4. NUMERICAL RESOLUTION

4.1 Variational Formulation

With the purpose of establishing the variational formulation of the boundary value problem done by (6), we recall the definition of a Sobolev Space of order two $H^2(\Omega)$,

$$H^2(\Omega) = \{S \in L^2(\Omega) : \frac{\partial^{(\alpha_1+\alpha_2)} S}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} \in L^2(\Omega), 0 \le \alpha_1 + \alpha_2 \le 2, \alpha_1, \alpha_2 \in IN\}.$$

We will consider the set of functions $(H^2(\Omega))^3$ satisfying the previous boundary conditions. We will denote this set by $\mathcal{H}$.

The weak formulation of the equation (6) is:

$$\int_\Omega \left(\omega_{10}\frac{\partial S}{\partial u}\frac{\partial T}{\partial u} + \omega_{01}\frac{\partial S}{\partial v}\frac{\partial T}{\partial v} + 2\omega_{11}\frac{\partial^2 S}{\partial u \partial v}\frac{\partial^2 T}{\partial u \partial v} + \omega_{20}\frac{\partial^2 S}{\partial u^2}\frac{\partial^2 T}{\partial u^2} + \omega_{02}\frac{\partial^2 S}{\partial v^2}\frac{\partial^2 T}{\partial v^2} \right) dudv = -\int_\Omega \nabla P(S) T dudv$$

where the functions S, T belongs to $\mathcal{H}$ and u, v are the spatial variables.

It can be proved [Cohen 1992] [Raviart 1992] that solving equation (6) is equivalent to find an element $S \in \mathcal{H}$, such that $a(S, T) = L(T)$ for all $T \in \mathcal{H}$, where $a(\cdot,\cdot)$ is a bilinear form defined as

$$a(S,T) = \int_\Omega \left(\omega_{10}\frac{\partial S}{\partial u}\frac{\partial T}{\partial u} + \omega_{01}\frac{\partial S}{\partial v}\frac{\partial T}{\partial v} + 2\omega_{11}\frac{\partial^2 S}{\partial u \partial v}\frac{\partial^2 T}{\partial u \partial v} + \omega_{20}\frac{\partial^2 S}{\partial u^2}\frac{\partial^2 T}{\partial u^2} + \omega_{02}\frac{\partial^2 S}{\partial v^2}\frac{\partial^2 T}{\partial v^2} \right) dudv \quad (7)$$

and L(.) is the following linear form

$$L(T) = -\int_\Omega \nabla\mathcal{P}(S) T dudv.$$

4.2 Discretization

We want to find a function $S \in \mathcal{H}$ such that

$$a(S, T) = L(T), \forall T \in \mathcal{H}. \quad (8)$$

In order to do this, the surface domain will be discretized. But, first of all, we have to find a set of functions of finite dimension. The B-splines defined in section 2 will be the finite elements that we will use as the base of our function set. The problem is to find the relevant B-splines; that is, the B-splines satisfying $Sup(B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}) \bigcap \Omega \neq \emptyset$, and from this, the set of index $\mathbf{k}$ of the B-splines satisfying the boundary conditions. Therefore, we want to evolve $N_1 \times N_2$ control points of the B-spline surface S. To do it, we need $N_1 \times N_2$ bidimensional B-splines such that its support will be in Ω.

The solution $S \in \mathcal{H}$ is a B-spline surface of degree $n = (n_x, n_y)$. The surface domain is discretized by $h_1 ZZ \times h_2 ZZ$ where $h_1 = \frac{1}{N_1 + n_x - 1}$ and $h_2 = \frac{1}{N_2 + n_y - 1}$. This spatial discretization will fix the control points that are not zero. The index $\mathbf{k} = (k_1, k_2)$ belongs to the set $\{-n_x, ..., N_1 + n_x - 1\} \times \{-n_y, ..., N_2 + n_y - 1\}$. So, the B-spline surface will come determined by the relevant B-splines, and they are specified by the following equation

$$S(u,v) = \sum_{k_1=-n_x}^{N_1+n_x-1} \sum_{k_2=-n_y}^{N_2+n_y-1} P_{(k_1,k_2)} B^{\mathbf{n}}_{(k_1,k_2)\mathbf{h}}(u,v) \quad (9)$$

In addition, deforming only the corresponding $N_1 \times N_2$ control points of the B-spline surface (9), we made sure that the boundary conditions are satisfied.

The B-spline bases are determined by the following set of finite elements of finite dimension: $V^{\mathbf{n}}_{\mathbf{h}} = < \{(B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v), 0, 0) : \mathbf{k} \in \mathcal{I}\} \cup \{(0, B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v), 0) : k \in \mathcal{I}\} \cup \{(0, 0, B^{\mathbf{n}}_{\mathbf{k},\mathbf{h}}(u,v)) : \mathbf{k} \in \mathcal{I}\} >$ where $\mathcal{I} = \{0, .., N_1 - 1\} \times \{0, .., N_2 - 1\}$. Thus, taking into account the boundary conditions, the control points $P_{\mathbf{k}}$ associated to B-splines belonging to the set $V^{\mathbf{n}}_{\mathbf{h}}$ are the unique ones that are computed using the equations (10) and (11) (see below).

Using equations (8) and (9) we can obtain three linear systems, one for each coordinate:

$$AP_i = L_i, \; i = 1, 2, 3 \quad (10)$$

where A is a square matrix and their elements are:

$$a((B^n_{k,h}, 0, 0), (B^n_{j,h}, 0, 0))_{(k,j) \in I \times I},$$

P_i is a vector of component i of each control point and L_i is a vector with components $L_1 = L((B^n_{k,h}, 0, 0))_{k \in \mathcal{I}}$, $L_2 = L((0, B^n_{k,h}, 0))_{k \in \mathcal{I}}$, $L_3 = L((0, 0, B^n_{k,h}))_{k \in I}$.

The static problem has been introduced. Next, we will construct the evolution model.

4.2.1 Dynamic Evolution Model

The classical dynamical model of evolution has been applied [Cohen 1992], [Qin 1997], [Montagnat, Delingette and Ayache 2001], [González, Mascaro, Mir, Palmer and Perales 2001], [Mascaró 2002] [Mascaro, Mir and Perales 2002]. In our dynamic model, the surface depends on time. So, we have $S(u, v, t)$. Nevertheless, this dependency only affects to the control points which is an advantage since in each iteration we do not have to calculate all the surface.

Therefore, we only must calculate the new control points. Thus, our dynamic model of evolution comes determined by the equations

$$M\frac{d^2P_i}{dt^2}+C\frac{dP_i}{dt}+AP_i=L_i, i=1,2,3. \quad (11)$$

where M and C are the mass and damping matrices respectively and they are diagonal matrices.

The dynamic system (11) has been discretized in time using the standard finite-difference approximation operators. In a first place a scheme based on central finite-differences has been used, where the difference operators are done by

$$\frac{\partial P}{\partial t}\approx\frac{P(u,v,t+\Delta t)-P(u,v,t-\Delta t)}{2\Delta t},$$
$$\frac{\partial P}{\partial t^2}\approx\frac{P(u,v,t+\Delta t)-2P(u,v,t)+P(u,v,t-\Delta t)}{\Delta t^2}.$$

Placing these operators in the evolution equation (11) we obtain

$$M\frac{P_i(t+\Delta t)-2P_i(t)+P_i(t-\Delta t)}{\Delta t^2}+C\frac{P_i(t+\Delta t)-P_i(t-\Delta t)}{2\Delta t}$$
$$+AP_i(t)=L_i,\ i=1,2,3. \quad (12)$$

So, to solve equation (12) we have to solve the following three linear system of equations for $P_i(t+\Delta t)$, associated to an explicit integration procedure for the ordinary differential equation:

$$G\cdot P_i(t+\Delta t)=H_i,\ i=1,2,3 \quad (13)$$

where $G=2M+\Delta t\,C$ and $H_i=2\Delta t\,L_i-(2\Delta t^2A-4M)(P_i(t)-(2M-\Delta t\,C)P_i(t-\Delta t)$. The previous system is easy to solve because matrix G is diagonal. Then, the original nonlinear differential equation (11) has been reduced to a sequence of diagonal linear algebraic systems done by (13). And, taking in account the set of finite elements that determines our B-spline bases, this algebraic system of equations is solved only for a reduced number of control points.

The used numerical scheme in the dynamic model depends on two previous iterations $P^{t-\Delta t}$ and P^t, taking the same P^0 and P^1, where $P^0=S(u,v,0)$.

5. EXAMPLES

This section shows several examples of deformations obtained applying our dynamical model. All the experiments displayed in this section has been made using $\omega_{10}=\omega_{01}=0.1$ and $\omega_{11}=\omega_{20}=\omega_{02}=0.01$ and taking a temporal step $t=0.1$.

In Figure 2, we show several iterations obtained using the dynamic model with bicubic B-splines, a positive force in the direction (0, 1, 0) and $N_1\times N_2=25$, The force is applied only in one point of the surface.

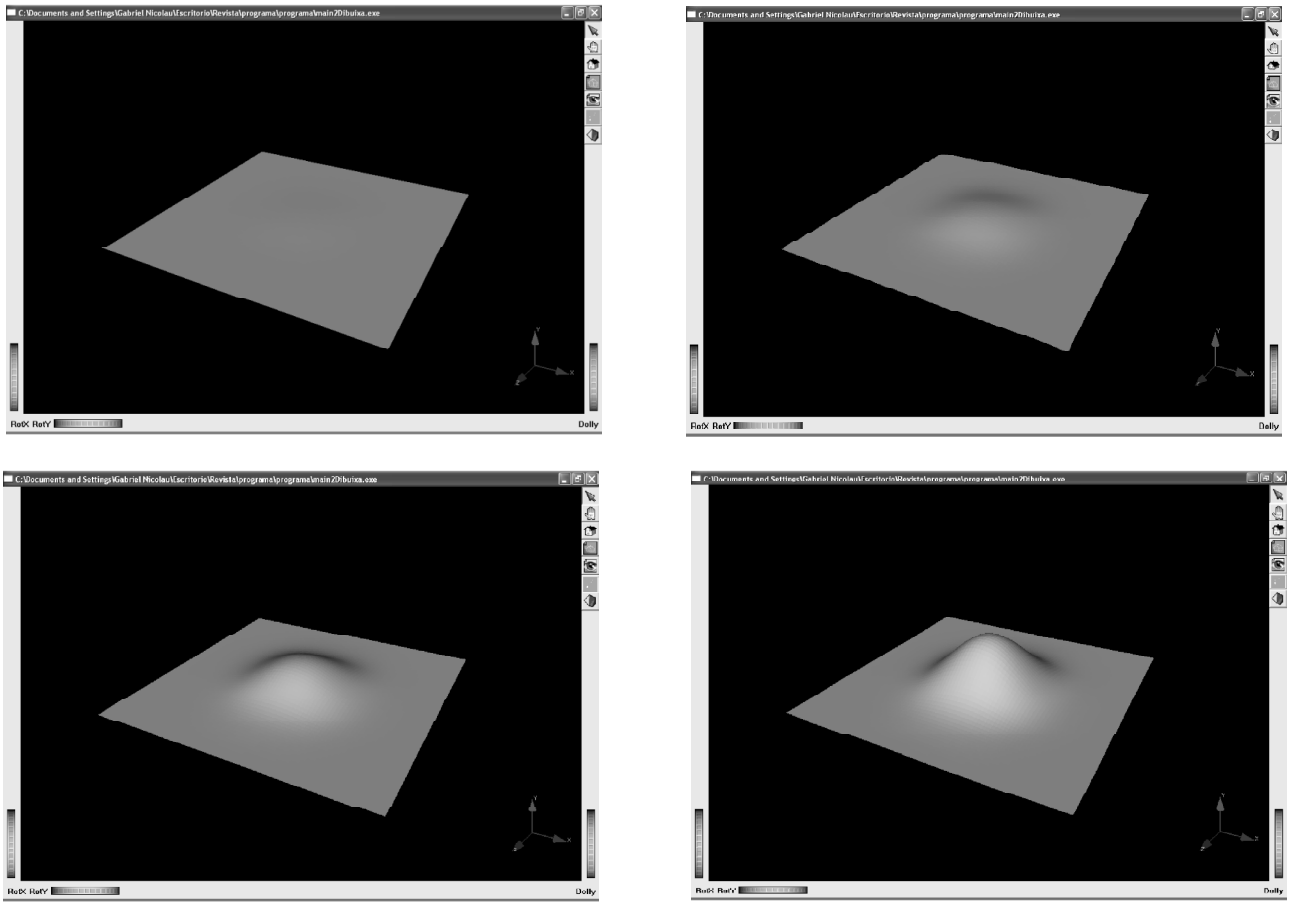


Figure 2: Dinamic simulation of a plane deformation using a positive constant force with direction (0, 1, 0)

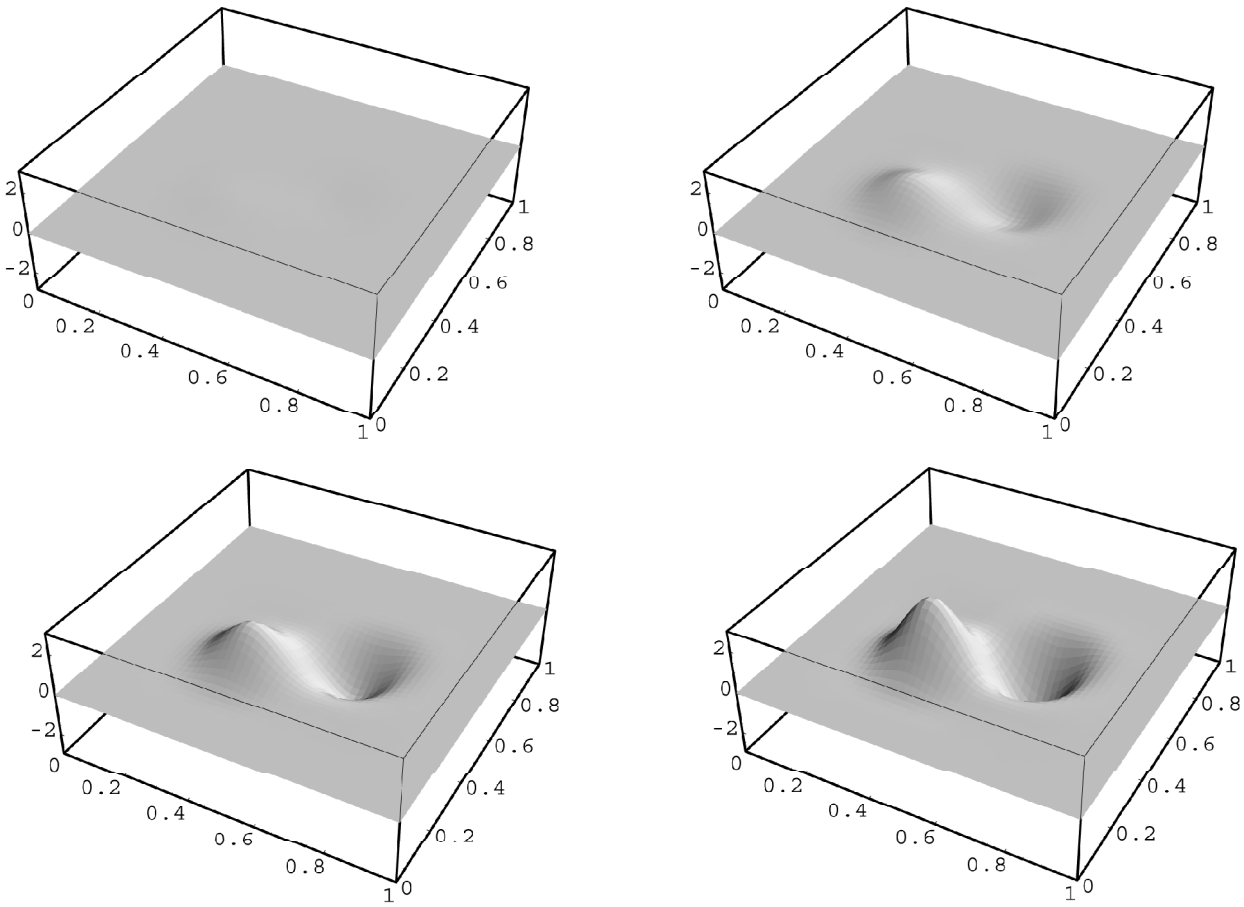


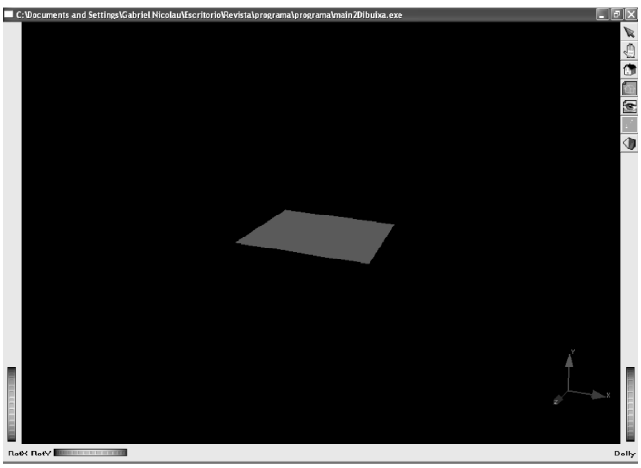
Figure 3: Dinamic simulation of a plane deformation using two constant forces simultaneously in opposite sense

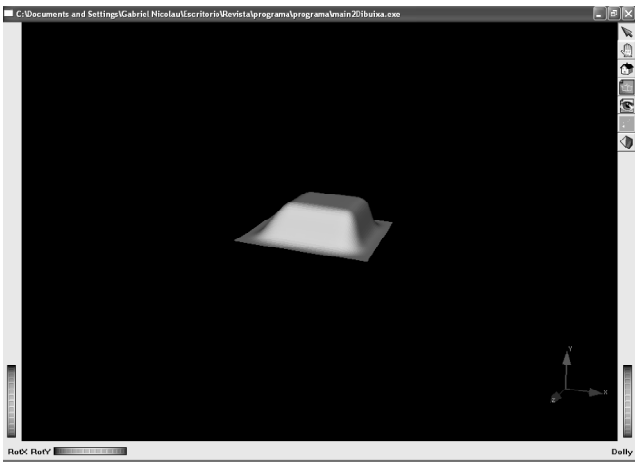
Figure 3 shows the deformation obtained using bicubic B-splines, two forces simultaneously in opposite sense in the direction (0, 0, 1) and $N_1 \times N_2 = 25$. The force is applied in two nearly points of the surface.

Not only we can apply forces in vertical directions. Also, we can apply forces in other directions as we can see in Figure 4, where we have applied over all the surface domain an oblique force in the direction (1, 4, 1), with module 106, taking bicubic B-splines and $N_1 \times N_2 = 49$.

Figure 5 shows several iterations obtained using the vertical force (0, 200, 0) over all the surface, using bicubic B-splines and $N_1 \times N_2 = 36$. We can compare this deformation with the obtained one in Figure 2.

The next figures, Figure 6 and Figure 7, display experiments using sinusoidal forces. In Figure 6 the force is given by $\nabla\mathcal{P}(u, v) = (50 \sin v\pi, 50 \sin u\pi, 200 \cos u\pi \sin v\pi)$, B-splines of degree $\mathbf{n} = (4, 4)$ and $N_1 \times N_2 = 36$. In Figure 7 the force is given by $\nabla\mathcal{P}(u, v) = (-50 \sin u\pi, 0, 200 \cos v\pi \sin u\pi)$, bicubic B-splines are considered and $N_1 \times N_2 = 36$.





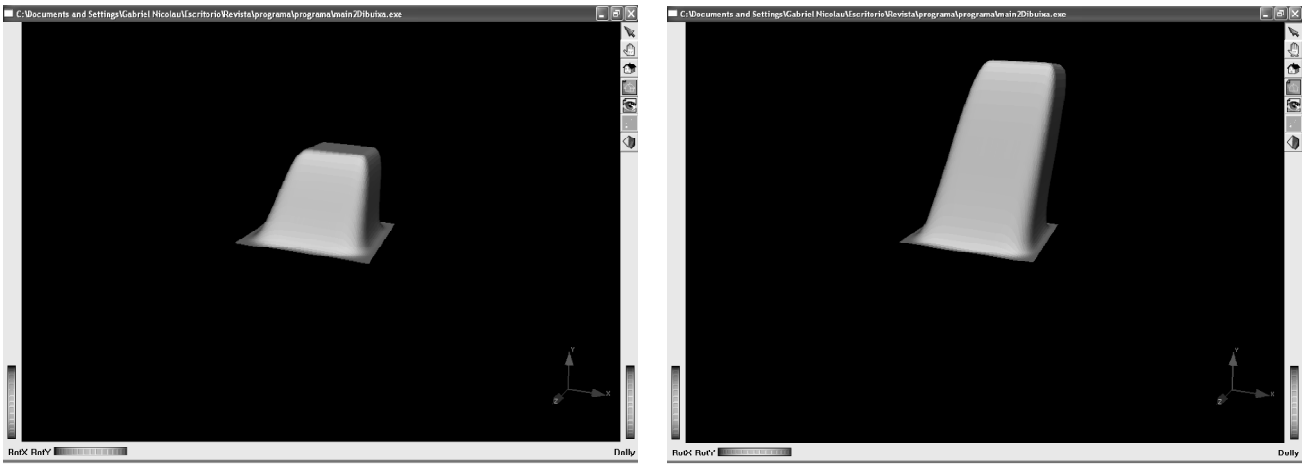


Figure 4: Dynamic simulation of a plane deformation using a force in the direction (1, 4, 1)

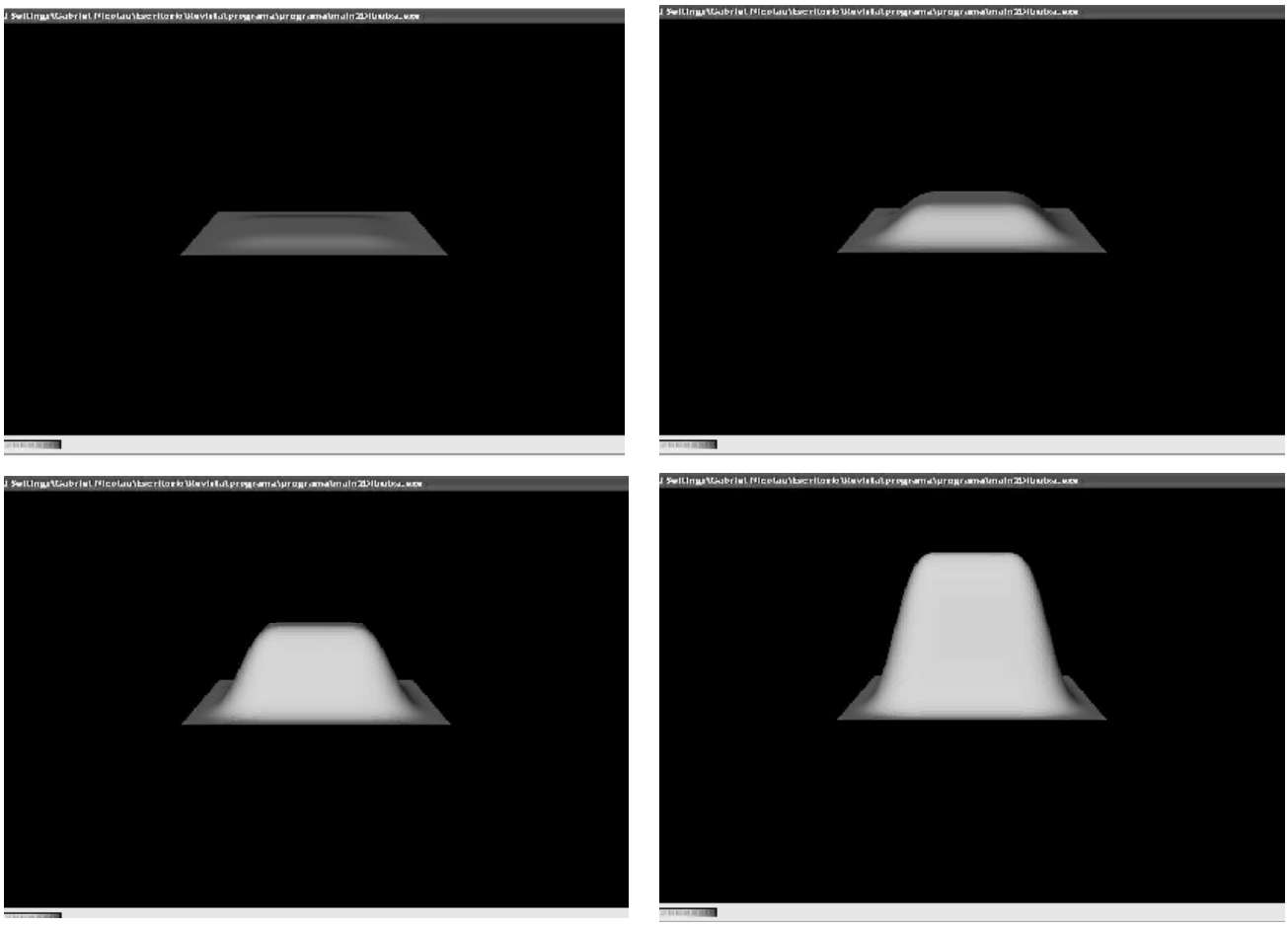


Figure 5: Several iterations of a plane deformation using the vertical force (0, 200, 0) over all the surface.

6. COMPUTATIONAL COST OF THE MODEL

We now consider the problem of study the computational cost of our model behavior, done by equation (13). More precisely, a simulation where $N_1 \times N_2$ control points are evolved, is considered. Let $\mathbf{n} = (n_1, n_2)$ be the B-spline degree.

First of all, we have to compute the computational cost of the vectors and matrices which are necessary to solve the ordinary differential equation numerically.

Initially, the computational cost of the bilinear matrix $(a(S, T))$ is $O((N_1 \times N_2)^2)$, but, taking into account that

(a) the support of the product of B-splines $B^n_{k,h}$ and $B^n_{j,h}$ or their derivatives only depends on the degree of the chosen B-splines $\mathbf{n} = (n_1, n_2)$,

(b) only $O((2n_1 + 1) \times (2n_2 + 1))$ values of each of the rows of matrix (a(S,T)) are non zero, and

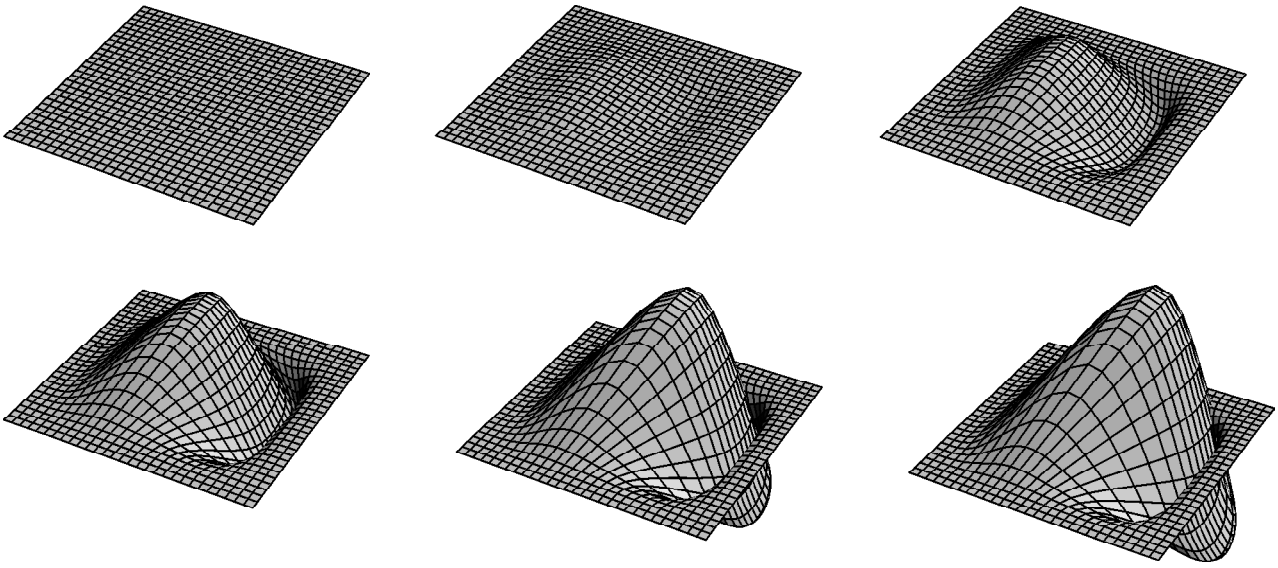


Figure 6: Dinamic simulation of a plane deformation using a sinusoidal force. Details in the text.

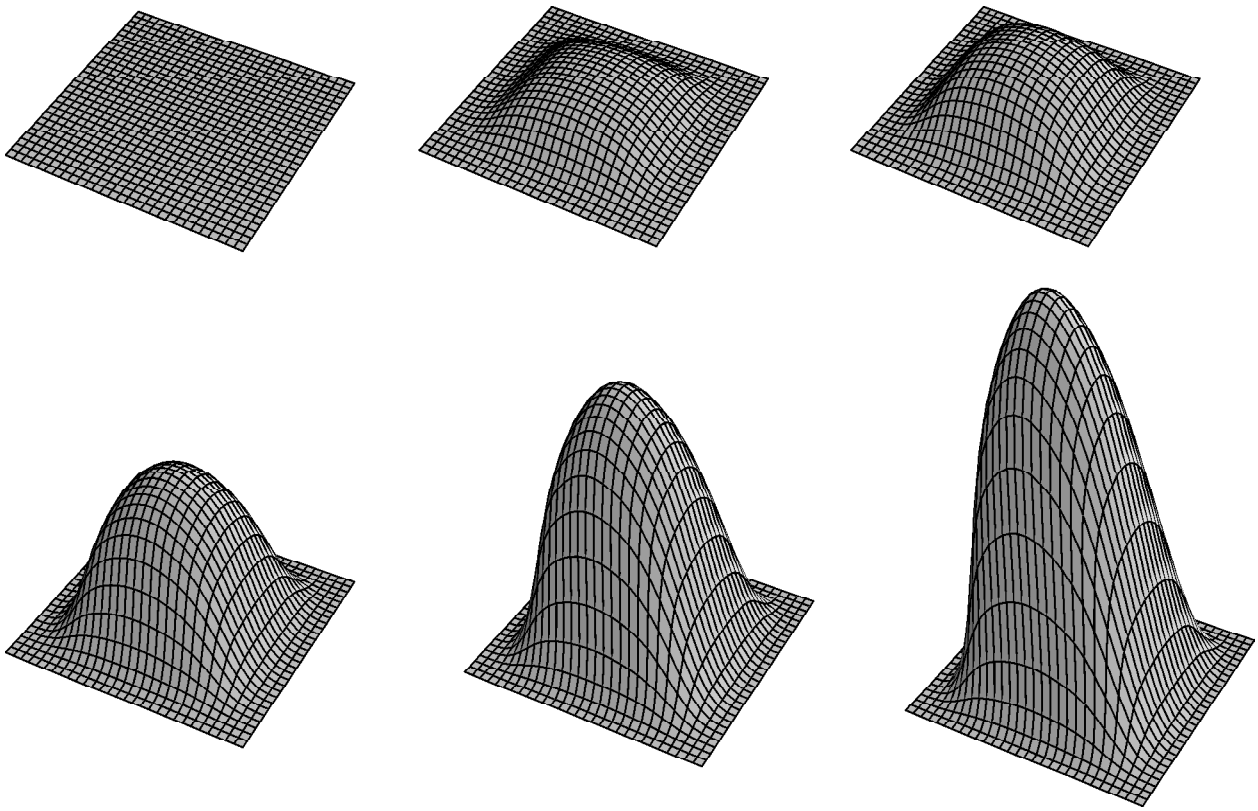


Figure 7: Deformations obtained using other sinusoidal force different from the applied one in the figure 6. Details in the text.

(c) the rows of this matrix are simply a shift of the values of the first row,

it is easy to check that the computational cost of the matrix associated to the bilinear form is only $O((2n_1 + 1) \times (2n_2 + 1))$. So, the computational cost of the bilinear matrix $(a(S, T))$ is constant since only depends on the degree of the *B*-splines.

This matrix is computed only one time at the beginning of the simulation process.

The mass and damping matrices are diagonal and constant for all the control points; so, its associated computational cost is constant.

Taking into account the support of *B*-splines $B^n_{k,h}$, the order of the computational cost of the vector L_i with components

$$\int_\Omega \nabla \mathcal{P}(S(u,v))\, B^n_{k,h}(u,v) du dv,$$

is $O(N_1 \times N_2)$.

Now we are going to analize the computational cost, in time, of the simulation part of our linear algebraic system (see equation (13)),

$$G \cdot P_i(t + \Delta t) = F_i - B_i - D_i,\ i = 1, 2, 3, \qquad (14)$$

where $G = 2M + \Delta tC$, $F_i = 2\Delta t\ L_i$, $B_i = (2\Delta t^2 A - 4\mathrm{M})\ P_i(t)$ and $D_i = (2M - \Delta tC)\ P_i(t - \Delta t)$. In general, the product of a vector by a matrix requieres $O((N_1 \times N_2)^2)$ time computing. But, to compute B_i, we need only $O(N_1 \times N_2)$ time computing, because M is diagonal and each row of the matrix A has only $O((2n_1 + 1) \times (2n_2 + 1))$ (constant) elements non zero. On the other hand, the computational cost of the matrices D_i and F_i is $O(N_1 \times N_2)$ because we only make the product of diagonal matrices by vectors.

Finally, to solve the three diagonal linear algebric system (14), we only require $O(N_1 \times N_2)$ computational cost since G is diagonal and the calculus of its inverse G^{-1} and the product of G^{-1} by $F_i - B_i - D_i$ are $O(N_1 \times N_2)$.

Then, we can conclude that the explicit procedure to integrate through time the ordinary differential equation represented by equation (13) need only a $O(N_1 \times N_2)$ time computing. That is, we obtain a linear complexity depending on the number of control points to deform.

7. THE INFLUENCE OF FIRST ORDER PARAMETERS

In this section, we have studied how the simulations of the model can change if the value of the parameters ω_{01} and ω_{10} are changed.

To do this, we have made several deformations changing the value of the parameter ω_{01} first, and ω_{10} later. The other parameters and the applied force remain constant throughout the simulation.

Let P^0 be the vector of control points of the initial surface; that is, the vector of control points of the surface before the deformation. The number of components of P^0 is $3 \times N_1 \times N_2$, three spatial components for each control point. So, if $P_1^0, ...P_{N_1 \times N_2}^0$ are the initial control points, P^0 can be written as:

$$P^0 = (P_{1,1}^0, P_{1,2}^0, P_{1,3}^0, ..., P_{N_1 \times N_2,1}^0, P_{N_1 \times N_2,2}^0, P_{N_1 \times N_2,3}^0)^\top$$

Using the same notation, we define P^f the vector of control points of the deformed surface; that is, the vector of control points of the surface after the deformation. So, P^f can be written as:

$$P^f = (P_{1,1}^f, P_{1,2}^f, P_{1,3}^f, ..., P_{N_1 \times N_2,1}^f, P_{N_1 \times N_2,2}^f, P_{N_1 \times N_2,3}^f)^\top.$$

The influence of the parameters has been measured using the following norm:

$$\| P^f - P^0 \| = \sqrt{\sum_{i=1}^{N_1 \times N_2} \sum_{j=1}^{3} \left(P_{i,j}^f - P_{i,j}^0 \right)^2}. \qquad (15)$$

7.1 Study for the Parameter ω_{10}

With the purpose of study the influence of parameter ω_{10}, we have made several simulations with differents values of them. The values of the other parameters have been $\omega_{01} = 0.1$, $\omega_{11} = 0.01$, $\omega_{20} = 0.01$, $\omega_{02} = 0.01$. The applied force in the simulations is zero over all surface except in a small central square whose value is 100.

In figure 8, we have shown a graphic where horizontal axis is the values of the parameter w_{10} and the vertical axis is the quadratic norm described above.

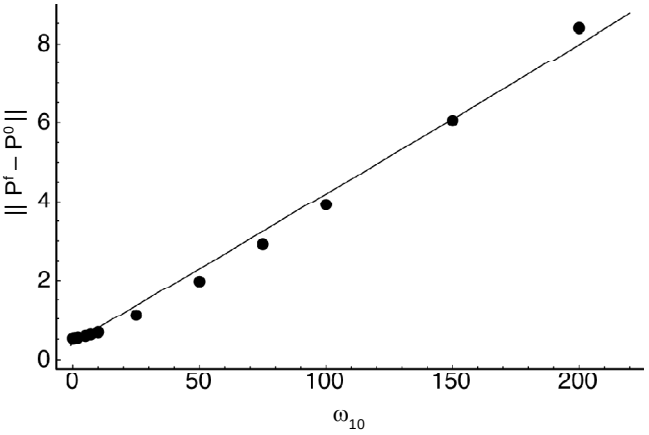


Figure 8: In this graphic are represented the norms (15) of the simulations with the values of w_{10} = 0; 0.0001; 0.001; 0.01; 0.1; 0.3; 0.5, 1; 2; 5; 7; 10; 25; 50; 75; 100; 150 and 200. Also, it displays the linear approximation of the values.

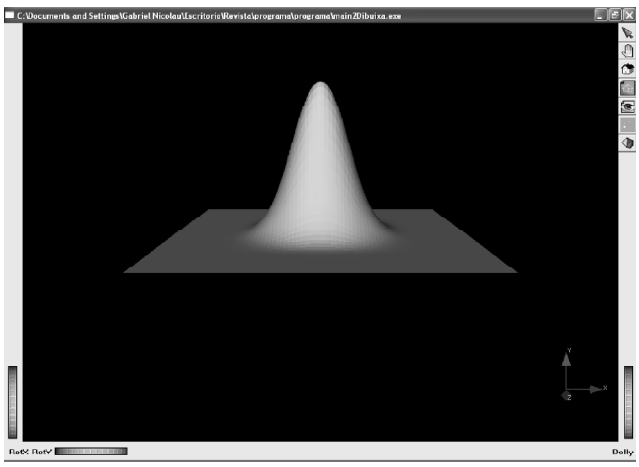
We can see in figure 8, as larger is the parameter ω_{10}, smaller is the resistance to the deformation. The relationship between the increase of the parameter ω_{10} and the decrease of the resistance to the deformation (15) is linear. If one compute the line of regression, one obtains $n = 0.4 + 0.038\ \omega_{10}$ where n is the norm (15) with regression coeffcient 0.9924.

The effects made by this parameter can be seen in figures 9 and 10. The figure 9 shows the effect of this parameter on a deformation made by a vertical force over a central square of surface $0.04u^2$ and module 200. In the figure 10, we can see the effect of this parameter over a deformation made by an oblique and positive force in direction (10, 40, 10), in this case, as we increase the parameter the length defomation is increased, following the direction of the applied force. So, we can conclude that the resistance to the length deformation decreases as the value of parameter ω_{10} increases.

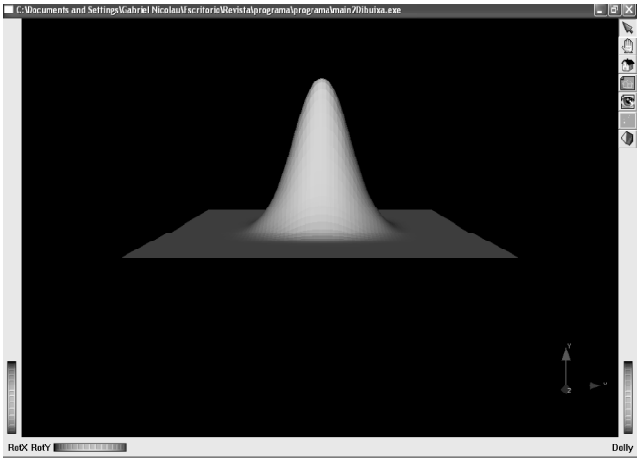
7.2 Study for the Parameter ω_{01}

We have studied the behaviour of parameter ω_{01} using the same outline of the previous one. The values of the other parameters have been $\omega_{10} = 0.1$, $\omega_{11} = 0.01$, $\omega_{20} = 0.01$, $\omega_{02} = 0.01$. The force applied in the deformation is zero over all surface except in a small central square which value is 100.

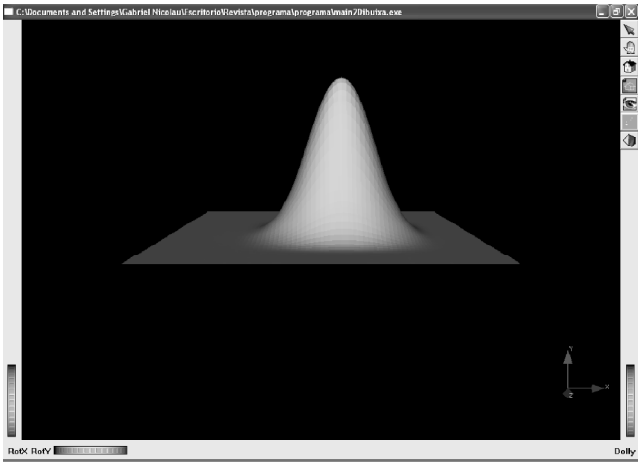
In figure 11, we show a graphic where horizontal axis is the values of the parameter w_{01} and the vertical axis is the quadratic norm described above (see (15)). We can also observe, (see figure 11), that as larger is the value of this parameter, smaller is the resistance to the deformation.



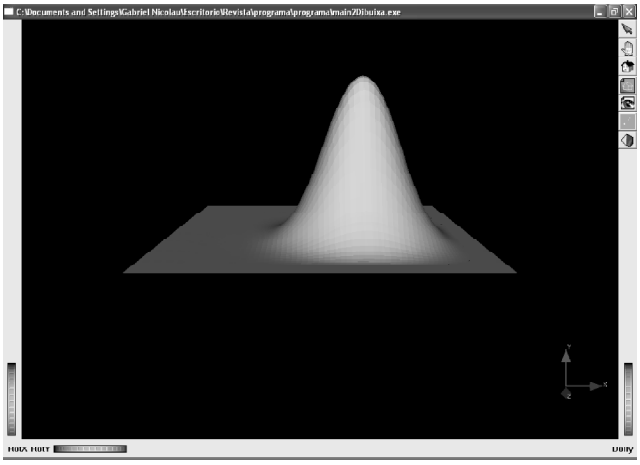
w_{10} = 0.0



w_{10} = 10.0

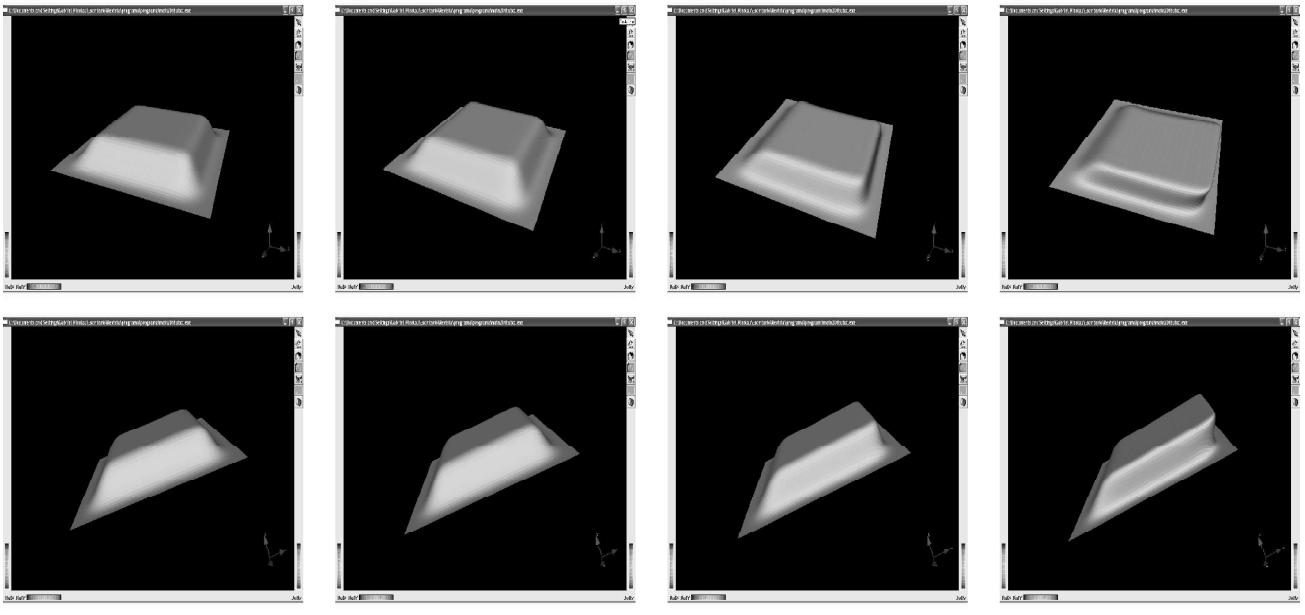


w_{10} = 100.0



w_{10} = 200.0

Figure 9: last iteration of deformation with Δt = 0.1 and t = 1. Vertical Force (0,200,0)



w_{01} = 0.0 w_{01} = 10.0 w_{01} = 100.0 w_{01} = 200.0

Figure 10: two points of view of the last iteration of deformation with Dt = 0.1 and t =1. Oblique force (10, 40, 10)

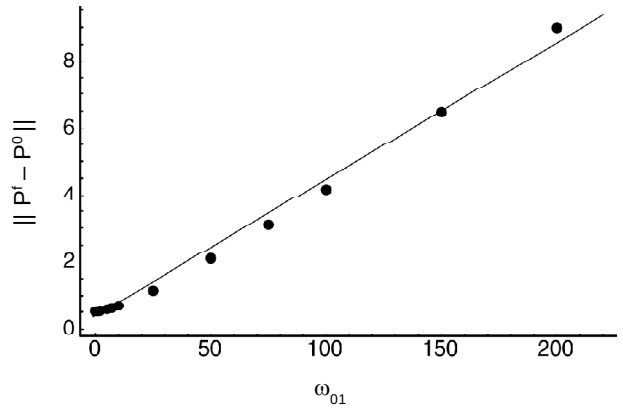
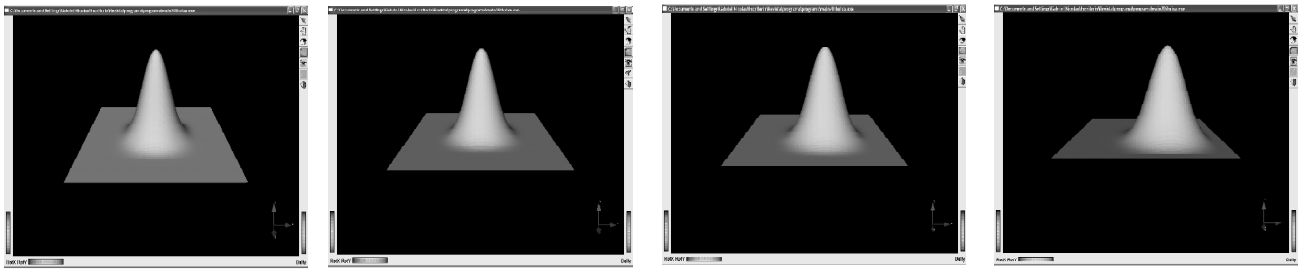


Figure 11: In this graphic are represented the quadratic norms of the simulations with the values of !01 = 0; 0.0001; 0.001; 0.01; 0.1; 0.3; 0.5, 1; 2; 5; 7; 10; 25; 50; 75; 100; 150 and 200. Also, the linear approximation of the values, is displayed.

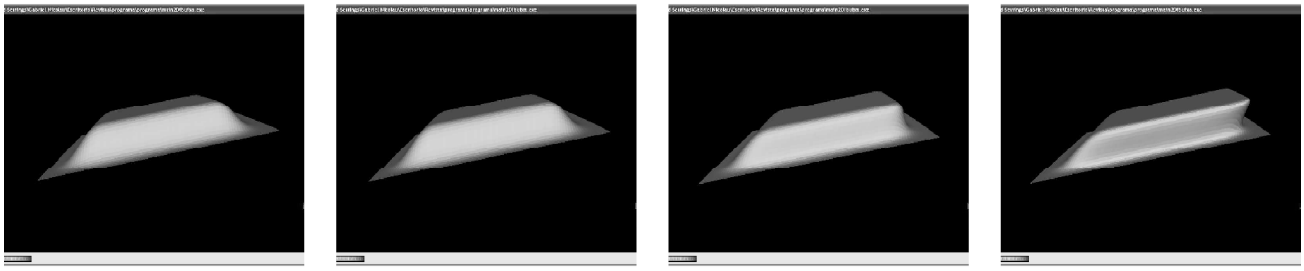
We can conclude that the growth of the first order parameters w_{10} and w_{01} makes a decreasing resistance to the length deformation of the surface. Moreover, if we represent the first order parameters in front of the deformations of the simulations, we obtain a linear curve. The line of regression is, in its case, $n = 0.3924 + 0.0407\omega_{01}$, with regression coefficient 0.9922.

Figure 11 shows as the surface deformation (15) is modified by the influence of parameter w_{01}. The effects made by this parameter can be seen in figures 12 and 13. The figure 12 shows the effect of this parameter on a deformation made by a vertical force over the central square of surface $0.04u^2$ and module 200. In the figure 13, we can see the effect of this parameter over a deformation made by an oblique and positive force in direction (10, 40, 10). We conclude that we observe the same effect as in the parameter w_{10}.



w_{01} = 0.0 **w_{01} = 10.0** **w_{01} = 100.0** **w_{01} = 200.0**

Figure 12: last iteration of deformation with t = 0.1 and t=1. Vertical Force (0,200,0)



w_{01} = 0.0 **w_{01} = 10.0** **w_{01} = 100.0** **w_{01} = 200.0**

Figure 13: last iteration of deformation with t = 0.1 and t=1. Oblique force (10, 40, 10)

8. CONCLUSIONS AND FUTUREWORK

We have developed a variational formulation of a model that allows us the deformation of a surface. Moreover, we have shown how to solve the variational equations numerically in a efficient way using B-splines as finite elements. More explicitly, since only a certain number of control points is evolved, the numerical scheme obtained is very efficient.

Also, we have obtained a low computational cost to solve the equations.

The study of behaviour of first order parameters shows that the growth of these makes a decreasing resistance of the surface.

If we want to obtain a smooth deformation, the external force module must be small or the first order parameters must be smaller than 1.

Now, we are working in the stability of the model and the behaviour of second order parameters w_{20}, w_{11} and w_{02}. Once we have studied all the previous topics of the model, we will work with other functionals in order to improve the deformation of the surface. As a future work, gravity forces, spring viscosity forces and collision forces will be applied. Moreover, we will apply our model to deform second order surfaces as cones, ellipsoids, spheres and so on.

The implementation of this model has been made using C^{++} and Coin3D, a 3D modelling toolkit which simplifies visualization and scene composition tasks.

ACKNOWLEDGEMENTS

This work is supported by the project TIC-2004-07926-E, INEVAI3D, of the Spanish Government. The authors would

like to thank to the department of Mathematics and Computer Science of University of the Balearic Islands.

REFERENCES

[1] Cohen, I. (1992), *Modéles Déformables 2-D et 3-D: Application àla Segmentation d'Images Médicales*. Ph. D. thesis, Universit´e Paris IX, Dauphine.

[2] Cox, M. G. (1972), The numerical evaluation of b-splines. *IMA Journal of Applied Mathematic* 10(2), 134–149.

[3] de Boor, C. (1978), *A practical guide to splines*. New York: Springer Verlag.

[4] de Boor, C., K. Höllig, and S. Riemenschneider (1993). *Box Splines*. New York: Springer Verlag.

[5] do Carmo, M. (1976), *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

[6] Farin, G. (1997), *Curves and Surfaces for Computer-Aided Geometric Desing: A Practical Guide*. Academic Press.

[7] González, M., M. Mascaró, A. Mir, P. Palmer, and F. Perales (2001), Modeling and animating deformable objects. In *Proceedings of IX Spanish Symposium on Pattern Recognition and Image Analysis*, Benicasim, Castellón, Spain, pp. 279–290. AERFAI Society.

[8] Gonźalez-Hidalgo, M., A. Mir, and G. Nicolau (2006), An evolution model of parametric surface deformation using finite elements based on B-splines. In *Proceedings of CompImage'2006 Conference, Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, Coimbra, Portugal.

[8] Höllig, K. (2003), *Finite element methods with B-Splines*. Frontiers in Applied Mathematics. Philadelphya: SIAM.

[9] Mascaró, M. (2002), *Modelo de Simulación de Deformaciones de Objetos Basado en la Teoŕýa de la Elasticidad*. Ph. D. thesis, Universitat de les Illes Balears.

[10] Mascaro, M., A. Mir, and F. Perales (2002), P3DMA: A physical 3D deformable modelling and animations system. *LNCS 2492*, 68-79.

[11] Montagnat, J., H. Delingette, and N. Ayache (2001), A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing* 19(14), 1023-1040.

[12] Piegl, L. & Tiller, W. (1997), *The NURBS book*. Berlin: Springer Verlag.

[13] Qin, H & Terzopoulos, D. (1997), Triangular nurbs and their dynamic generalizations. *Computer Aided Geometric Design* 14, 325–347.

[14] Raviart, P. A. & Thomas, J. M. (1992), *Introduction à l'analyse numérique des équations aux derivées partielles*. Paris: Masson.

[15] Terzopoulos, D. (1986), Regularization of inverse visual problems involving discontinuities. *IEEE PAMI* 8(4), 413-424.