Deriving Mass-Spring Systems from Dynamic NURBS Approach

Josildo Pereira da Silva, Antônio Lopes Apolinário Júnior
Department of Science Computer
UFBA–Federal University of Bahia
Salvador,BA,Brazil
josildo091@dcc.ufba.br, apolinario@dcc.ufba.br

Gilson A. Giraldi
LNCC–National Laboratory for Scientific Computing
Petrópolis, RJ, Brazil
gilson@lncc.br

***Abstract*—Mass-Spring Models (MSM) are frequently used for modeling and simulation of deformable objects for computer graphics applications due to their simplicity and computational efficiency. However, the model parameters (stiffness coefficients, damping and masses) are not related to the constitutive laws of elastic material in an obvious way. The computation of mass-spring parameters from a model based on continuum mechanics is a possibility to this problem. More recently, computer aided design (CAD) and finite element (FEM) community realized the need to unify CAD and FEM descriptions which motivates the revision of dynamic NURBS (D-NURBS) approach for modeling deformable objects. In this paper we address the problem of determining stiffness coefficients of the MSM to mimic D-NURBS approach. We validate the methodology for deriving MSM systems by comparing the obtained results with D-NURBS evolution nearby the steady-state configuration.**

***Keywords*-Physically based Modeling, Mass-Spring Model, Dynamic NURBS.**

I. INTRODUCTION

In the last decades, a wide variety of physically based models has been developed by the computer graphics community to address the challenge of simulating natural elements and deformable materials [1]. This paper is focused on the simulation of deformable objects for virtual environments. The mechanical behavior of elastic materials can be simulated by continuum elasticity models that describes how the objects deform under applied forces. In this case, constitutive laws are used for the computation of the symmetric internal stress tensor σ, and a conservation law gives the final (Partial Differential Equation) PDE that governs the dynamic of the material [1]. Continuous systems have infinite degrees of freedom which difficult its description for both the geometric and dynamic aspects. In mathematical terms, we are dealing with infinite basis functions, maybe uncountable. One possibility to simplify the problem is to consider finite dimensional representation with enough flexibility in order to represent the solution with the desired precision. In the context of mechanical systems the Finite Element Method (FEM) is the traditional way to perform this task.

However, as pointed out in [2], NURBS framework can be also considered. That is why geometric modeling and FEM community realized the need to unify CAD and FEM descriptions (Isogeometric Analysis [2]) which motivates a revision of D-NURBS concepts as proposed in [3]. Other possibility for elastic objects simulation is to apply discrete models, based on mass-spring models (MSM). In this case, the object geometry is represented by a mesh and its nodes are treated like mass points while each edge acts like a spring connecting two adjacent nodes [4].

MSM models are simple to implement and can be faster then the continuous ones, and so, more suitable for real time applications [4]. However, the main limitation of MSM models is the difficulty of designing them to represent the mechanical behavior of deformable bodies with enough accuracy [5]. In this article, a preliminary study of the general problem of determining parameters of the MSM, addressing specifically the problem of calculating the stiffness coefficients of the springs of the model. Particularly, we use D-NURBS as reference model and follow [6] to compute the appropriate stiffness coefficients of the springs by comparing the stiffness matrices generated by both the D-NURBS and the linearize MSM approaches.

The paper is organized as follows. In section II we show the key points of the D-NURBS and MSM formulations. In section III we describe our proposal to derive MSM systems from dynamic NURBS approach. The paper ends with computational experiments and perspectives, presented in sections IV and V, respectively.

II. D-NURBS AND MSM APPROACHES

The idea behind D-NURBS is to submit an initial NURBS curve to a Newtonian dynamics generated by an external potential, internal (elastic) and dissipation forces. Therefore, a natural way to parameterize the evolution of the curve along the time t is:

$$\mathbf{c}(u,t) = \frac{\sum_{i=0}^{n} \mathbf{p}_i(t) w_i(t) B_{i,k}(u)}{\sum_{i=0}^{n} w_i(t) B_{i,k}(u)}, \ u \in [0,1], \quad (1)$$

where $B_{i,k}$ are B-spline functions of order k, $\mathbf{p}_i(t)$ are the control points and $w_i(t)$ the weights which become generalized coordinates of the system and can be concatenated as $\mathbf{p}(t) = \left[(\mathbf{p}_0^T, w_0) \quad (\mathbf{p}_1^T, w_1) \quad \ldots \quad (\mathbf{p}_n^T, w_n) \right]^T \in \Re^{4(n+1)}$ [7].

As time goes on, the system changes its configuration due to internal and external forces. Therefore, the evolution of the system can be seem as a continuous path, or curve, $\mathbf{p}(t)$ in the configuration space, parameterized through the time t. The Hamilton's Principle gives a methodology to write the evolution equation of the system in terms of the generalized coordinates and time t. It states that for a mechanical systems with kinetic energy $T = T\left(\dot{\mathbf{p}}\right)$, where $\dot{\mathbf{p}} = d\mathbf{p}/dt$, with all force fields derivable from a generalized scalar potential $V = U\left(\mathbf{p}\right) + F\left(\mathbf{p}{,}\dot{\mathbf{p}}\right)$, the time evolution satisfies the associated Euler-Lagrange equations, which renders the following D-NURBS evolution equation [7]:

$$M\ddot{\mathbf{p}} + D\dot{\mathbf{p}} + K_D\mathbf{p} = -\frac{\partial E_{ext}}{\partial \mathbf{p}} - I\dot{\mathbf{p}}, \qquad (2)$$

where matrices M, D, K_D and I are computed by equations:

$$M = \int_u \mu J^T J du; \quad D = \int_u \gamma J^T J du; \qquad (3)$$

$$K_D = \int_u \left(\alpha J_u^T J_u + \beta J_{uu}^T J_{uu}\right) du; \quad I = \int \mu J^T \dot{J} du; \quad (4)$$

and $J \in \Re^{3\times 4(n+1)}$ is the associated Jacobian, defined in [3].

In the MSM system the mesh nodes work as masses and the edges define the linear springs with damping. So, given a particle i with mass and position vector $\mathbf{x}_i$, the force system is composed by the elastic ($\mathbf{f}^i_{elastic}$), gravitational ($\mathbf{f}^i_{grav}$) and damping ($\mathbf{f}^i_{damp}$) forces, defined respectively, by [4]:

$$\mathbf{f}^i_{elastic} = \sum_{j\in V} k_{ij}\left(l_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|\right)\frac{(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|}, \qquad (5)$$

where V is the set of nodes linked to $\mathbf{x}_i$, k_{ij} is the stiffness of the spring linking the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ and l_{ij} the spring rest length;

$$\mathbf{f}^i_{grav} = m_i\mathbf{g}; \quad \mathbf{f}^i_{damp} = \gamma_i\dot{\mathbf{x}}_i \qquad (6)$$

with $\mathbf{g}$ being the gravity field and γ_i is the damping factor.

Following Newton's Laws, we get the following evolution equation:

$$m_i\ddot{\mathbf{x}}_i = \mathbf{f}^i_{elastic} + \mathbf{f}^i_{damp} + \mathbf{f}^i_{grav} \qquad (7)$$

Both equations (2) and (7) need initial conditions composed by the initial configuration and velocity to assure existence and uniqueness of the solution. These conditions will be also the input to numerical methods based on finite difference method (FDM) in time and Gauss quadrature for computing the integrals (see section 4 of [3]) to get the matrices M, D, K_D and I in expressions (3) and (4).

III. Proposed Method

The aim of the proposed method is to compute the MSM parameters such that it behaves like the D-NURBS model for small deformations nearby the steady-state configuration. We will focus on deformable surface models but the methodology can be straightforward extend to $3D$. The topology of the MSM is pictured on Figure 1 which shows how the particles are connected by structural (edges) and shear (diagonals) springs to account for tension.

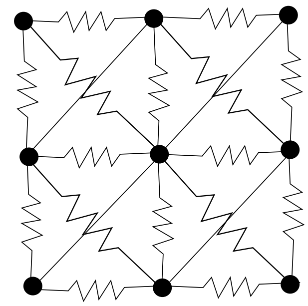


Figure 1. MSM 2D Lattice.

In this model, the stiffness coefficients of the springs are k_{edge}(structural) and k_{diag}(shear). The steady-state configuration of the D-NURBS model can be obtained by solving the equation:

$$K_D\mathbf{p} = -\frac{\partial E_{ext}}{\partial \mathbf{p}} \qquad (8)$$

whose solution is the steady-state solution of D-NURBS equation (2). It is clear that the stiffness matrix K_D takes a central role in expression (2) in the sense that it gives the elastic forces that balance the external forces in the steady-state configuration.

Therefore, a simple way to derive a MSM model that mimics D-NURBS nearby the steady-state solution is to find the MSM parameters that generate a linearized MSM model that approximates the D-NURBS one. In order to perform this task we must linearize the MSM governing equation by using the first order approximation for the force of the spring that connects nodes i and j around the rest nodal position $\left(\mathbf{x}_i^0, \mathbf{x}_j^0\right)$:

$$\mathbf{f}\left(\mathbf{x}_i, \mathbf{x}_j\right) \approx \mathbf{f}_{(i,j)} + \ldots \left[\frac{\partial \mathbf{f}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i}; \frac{\partial \mathbf{f}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j}\right]\left[\begin{array}{c}\left(\mathbf{x}_i - \mathbf{x}_i^0\right)\\ \left(\mathbf{x}_j - \mathbf{x}_j^0\right)\end{array}\right]. \qquad (9)$$

The first term of expression (9) can be discarded because $\mathbf{f}_{(i,j)} = \mathbf{f}\left(\mathbf{x}_i^0, \mathbf{x}_j^0\right) = -\mathbf{f}\left(\mathbf{x}_j^0, \mathbf{x}_i^0\right)$, and so, they will cancel each other when computing the resultant force. The second and third terms can be obtained by computing the derivatives of expression (5) respect to $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The process of assembling the linearized equations for the springs in the mesh of Figure 1 gives a symmetric stiffness matrix K_S. The linearized MSM model has the governing equation:

$$M_S\ddot{\mathbf{x}} + D_S\dot{\mathbf{x}} + K_S\left(\mathbf{x} - \mathbf{x}^0\right) = \mathbf{F}\left(\mathbf{x}\right) \qquad (10)$$

Before proceeding, we must observe that MSM is formulated in Euclidian spatial coordinates while D-NURBS uses generalized coordinates. Therefore, we would make both formulations compatible before establishing relationships. In order to perform this task suppose that the MSM mesh has N nodes that belong to the D-NURBS surface at time $t = 0$. So, there exist a parameter value $u = u_i$ such that

$$\left[\begin{array}{c} x_{i1}\\ x_{i2}\\ x_{i3}\end{array}\right] = J\left(u_i, \mathbf{p}(0), 0\right).\mathbf{p}(0) \qquad (11)$$

for $i = 0, 1, ..., N$. Therefore, by supposing that the initial configuration is the steady-state one and that we have small displacements nearby this configuration we can write the MSM steady-state equation ($K_S \left(\mathbf{x} - \mathbf{x}^0\right) = \mathbf{F}\left(\mathbf{x}\right)$) as:

$$K_S \begin{bmatrix} J\left(u_0, \mathbf{p}(0), 0\right).\left(\mathbf{p}(t_1) - \mathbf{p}(0)\right) \\ J\left(u_1, \mathbf{p}(0), 0\right).\left(\mathbf{p}(t_1) - \mathbf{p}(0)\right) \\ J\left(u_2, \mathbf{p}(0), 0\right).\left(\mathbf{p}(t_1) - \mathbf{p}(0)\right) \\ . \\ . \\ J\left(u_n, \mathbf{p}(0), 0\right).\left(\mathbf{p}(t_1) - \mathbf{p}(0)\right) \end{bmatrix} = \mathbf{F}\left(x\right). \quad (12)$$

where t_1 is a generic time instant. So, we can finally write:

$$\Omega^T \cdot K_S \cdot \Omega \cdot \left(\mathbf{p}(t_1) - \mathbf{p}(0)\right) = \Omega^T \cdot \mathbf{F}\left(x\right) \quad (13)$$

where:

$$\Omega = \begin{bmatrix} J\left(u_0, \mathbf{p}(0), 0\right) \\ J\left(u_1, \mathbf{p}(0), 0\right) \\ J\left(u_2, \mathbf{p}(0), 0\right) \\ . \\ . \\ J\left(u_n, \mathbf{p}(0), 0\right) \end{bmatrix}. \quad (14)$$

We shall observe that $\Omega \in \Re^{3(n+1)\times 4(n+1)}$ and $K_S \in \Re^{3(n+1)\times 3(n+1)}$, and so, the above matrix operations are well defined. Therefore, in order to get a MSM that simulates the D-NURBS nearby the steady-state configuration we shall solve the problem

$$\min_{K_S} \left\| \Omega^T \cdot K_S \cdot \Omega - K_D \right\|_2^2. \quad (15)$$

The solution of the problem above gives the stiffness coefficients of the springs. Next, we must find the mass m_i associated to each node $\mathbf{x}_i$ and the damping fator γ_i of the MSM. In the actual implementation, we set to null the damping factor. To get the mass, we compute the eigenvalues of the mass matrix M and then test each one to find the best value for all the masses. Specifically, we sort the non-null eigenvalues in decreasing order, choose the smallest one λ_{min} and simulate the MSM obtained by setting $m_i = \lambda_{min}$. Then, we compare both the MSM and D-NURBS systems and stop if their close enough, that means the difference in their behaviour is minimum. Otherwise, we take the next eigenvalue and repeat the process.

The D-NURBS equation (2) without damping is simulated with initial conditions:

$$\mathbf{p}\left(\mathbf{0}\right) = \mathbf{p_0}, \quad (16)$$

$$\dot{\mathbf{p}}(\mathbf{0}) = \delta\dot{\mathbf{p}}, \quad (17)$$

using the FDM described in [3], where $\delta\dot{\mathbf{p}}$ is a small velocity in generalized coordinates. Once defined the initial velocity in D-NURBS coordinates space, its counterpart in Euclidian coordinates can be obtained by using the fact that (see expression (39) of [3]):

$$\begin{bmatrix} \dot{x}_{i1}^0 \\ \dot{x}_{i2}^0 \\ \dot{x}_{i3}^0 \end{bmatrix} = J\left(u_i, \mathbf{p}(0), 0\right).\dot{\mathbf{p}}(0), \ i = 0, 1, 2, \ldots, n \quad (18)$$

So, the initial velocity for MSM is $\dot{\mathbf{x}}\left(0\right) = \delta\mathbf{v}$, where $\delta\mathbf{v} = \mathbf{\Omega}\dot{\mathbf{p}}(\mathbf{0})$ and $\dot{\mathbf{p}}(0)$ is the initial velocity for the D-NURBS. The initial configuration of the MSM is obtained by equation (11) and the MSM simulation is performed using the FDM described in [8].

IV. Experiments

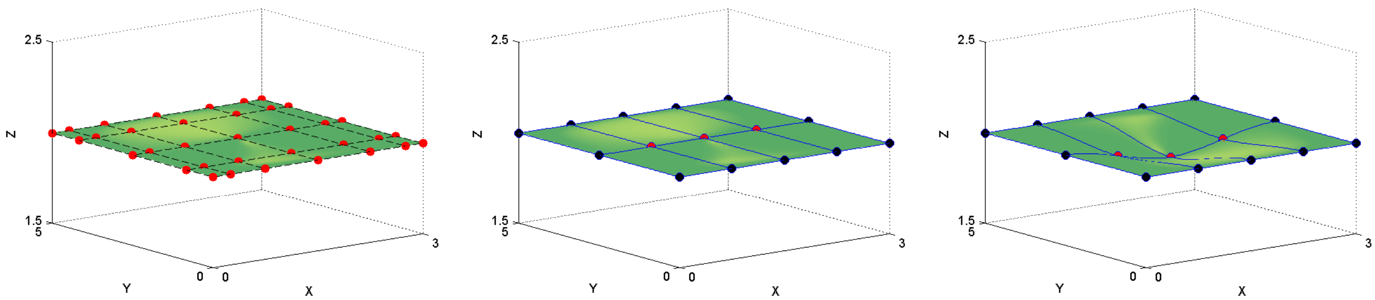
We have developed an experimental environment based on the D-NURBS approach with constraints. In our setting we consider the case of an elastic surface with dimension $5 \times 3m^2$ with negligible transverse section fixed at boundary, as shown in Figure 2a.

The NURBS surface geometry is parameterized by (u, v), with $0 \leq u, v \leq 1$. It is instantiated using an open knot vector $\mathbf{U} = (0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1)$ in the u direction, with B-spline functions of order $p = 4$ (degree $p - 1 = 3$), and knot vector $\mathbf{V} = (0, 0, 0, 0, 0.5, 1, 1, 1, 1)$ in the v direction, with B-spline functions of order $q = 4$ (degree $q - 1 = 3$). Therefore, the $2D$ spline space is given by the tensor product between the $1D$ B-spline bases and has dimension $(n - p) \times (m - q) = (11 - 4) \times (9 - 4) = 7 \times 5$, which means that we have 35 controls points, as pictured on Figure 2a. Each point of the NURBS surface is influenced by 4 control points. Therefore, to set geometric constraints that keep the surface fixed at the boundary, we must let 32 fixed control points. We consider null the external force field and define control points position and weights at $t = 0$ according to the cartesian product $P_x \times P_y \times \{2\}$ where $P_x = (0, 0.25, 0.75, 1.52, 2.5, 2.75, 3)$ and $P_y = (0, 0.83, 2.5, 4.16, 5)$ and $W = (1, 1, \ldots, 1)$. In our actual simulation, we keep the weights constant along the simulation in order to simplify the D-NURBS evolution.

Besides, we set $\dot{\mathbf{p}}(0) = (\dot{p}_{ix} = 0, \dot{p}_{iy} = 0, \dot{p}_{iz} = -0.2, \dot{w}_i = 0;\ i = 0, 1, \ldots, 34)$ to complete the initial conditions. To perform spatial integration we define 4 points in Gauss quadrature and for time integration of equation (2) we consider the finite difference scheme described in [3], with time step $\triangle t = 0.08s$, respectively. Once the external forces are null, the steady-state solution of D-NURBS corresponds to the same configuration shown in Figure 2a. In the case of D-NURBS surfaces, the corresponding matrix K_D of expression (4) has 5 parameters $\alpha_{ii}, \beta_{ij},\ i, j = 1, 2$ [7]. In the actual implementation we set non-null only the bending stiffness coefficients $\beta_{11} = \beta_{22} = 0.01$ and set the mass density $\mu = 25$.

The MSM structural mesh follows the topology presented on Figure 2b. Its time integration follow the finite difference scheme presented in [8] with time step $\triangle t = 0.08s$. We generate the MSM system using the mesh defined by the D-NURBS patches at $t = 0$, that are pictured on Figure 2b. Therefore,

$$\mathbf{x}_{ij}\left(0\right) = J\left(u_i, v_j, \mathbf{p}(0), 0\right).\mathbf{p}(0), \quad (19)$$



(a) Controls points at time $t = 0s$ (b) NURBS patches and MSM structural topology (c) D-NURBS at $t = 4s$

Figure 2. Mesh configuration

with $i = 0, 1, \cdots, 4$; $j = 0, 1, 2$, $u_i \in \mathbf{U}$ and $v_i \in \mathbf{V}$.

Now, we compute the analytical expression for K_S as a function of k_{edge} and k_{diag}(equation (9)). The process of assembling the linearized equations for the springs in the mesh of Figure 2b gives a symmetric 45×45 stiffness matrix K_S. Then, we solve the optimization problem (15) in order to find the best stiffness coefficients. We apply the standard trust-region optimization method [9] to perform this task. In this way, we have derived the MSM from the D-NURBS model.

The next step is simulate the two models in order to verify the quality of the result for small perturbations of the steady-state solution. For simplicity, we set null damping for both models. Then, we perform temporal integration of expressions (2) and (7), where the weights w_i were considered constant, so we have simplified the D-NURBS by removing the term $\frac{\partial \mathbf{c}}{\partial w}$ from the Jacobian. Figure 2c shows the D-NURBS at $t = 4s$ using the initial conditions described above.

The initial velocity $\dot{\mathbf{x}}(\mathbf{0})$ of the MSM model is obtained following expression (18) and the initial configuration $\mathbf{x}(0)$ is obtained through equation (19). The best masses value is given by the eigenvalue $\lambda = 0.005$ from the mass matrix M.

Figure 3 shows the time evolution in the z direction of the central node of Figure 2b for D-NURBS and MSM simulations. The obtained MSM parameters are: $m = 0.005$, $k_{edge} = 2.7225$ and $k_{diag} = 0.8661$. The x, y coordinates remain unchanged due to the initial conditions and the fact that the external force is null. We observe a suitable agreement between both models which indicates that the derived MSM mimics well the D-NURBS almost nearby the steady-state position.

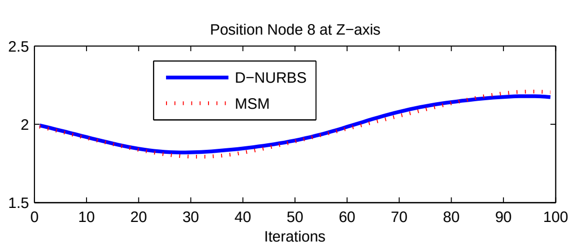


Figure 3. Comparing D-NURBS vs. MSM

V. Conclusions and Perspectives

Despite of the simplicity and computational efficiency of MSM systems the model parameters (stiffness coefficients, damping and masses) are not directly related to the constitutive laws of elastic material which limitates their applications. In this paper we address this problem through a metodology for determining stiffness coefficients of the MSM that mimic D-NURBS approach nearby the steady-state configuration. We validate the proposed methodology by comparing the obtained results with D-NURBS evolution. We observe in this preliminary study a good agreement between both the systems.

Further direction of this research is to adapt the eigenproblem approach presented in [6] which performs the analysis of the spectral properties and principal directions of the stiffness matrices in order to find most suitable parameters for MSM. Besides, damping and mass parameter computation can be incorporated in the methodology following the reference [2]. We can also augment the model by adding interleaving springs for bending [4].

References

[1] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, “Physically based deformable models in computer graphics,” *Computer Graphics Forum*, vol. 25, no. 4, p. 809–836, 2006.

[2] J. Cottrell, T. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[3] J. P. Silva, A. L. Apolinário, and G. A. Giraldi, “A review of dynamic NURBS approach,” University Federal of Bahia, Bahia,Brazil, technical, Mar. 2013. [Online]. Available: http://arxiv.org/abs/1303.6683

[4] P. Volino and N. Thalmann, *Virtual Clothing.: Theory and Practice.* Springer-Verlag GmbH, 2000.

[5] B. A. Lloyd, G. Székely, and M. Harders, “Identification of spring parameters for deformable object simulation,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 1081–1094, Sep. 2007.

[6] G. S. V. Otamendi, “Designing deformable models of soft tissue for virtual surgery planning and simulation using the mass-spring model,” Ph.D. dissertation, University of Navarra, Spain, Jan. 2011.

[7] D. Terzopoulos and H. Qin, “Dynamic NURBS with geometric constraints for interactive sculpting,” *ACM Trans. Graph.*, vol. 13, no. 2, pp. 103–136, Apr. 1994. [Online]. Available: http://doi.acm.org/10.1145/176579.176580

[8] S. Chapra and R. Canale, *Numerical Methods for Engineers*. McGraw-Hill Education, 2009.

[9] S. Wu and F. Wu, “Computation of a trust region step,” *Acta Mathematicae Applicatae Sinica*, vol. 7, no. 4, pp. 354–362, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF02009686