DYNAMIC ANIMATION OF SPLINE LIKE OBJECTS

Yannick Rémion, Jean-Michel Nourrit, Didier Gillard

Laboratoire d'Etudes et de Recherches Informatiques,
Université de Reims Champagne-Ardenne,
rue des crayères, BP 1035, 51687 Reims cedex 2,
France
{yannick.remion, jm.nourrit, didier.gillard}@univ-reims.fr http://www.univ-reims.fr/leri

ABSTRACT

This paper presents an accurate and efficient method for dynamic animation of one-dimensional objects modelled as successions of spline segments. At each time step the object shape conforms to its splines definitions, thus insuring that every property implied by the nature of the chosen splines is verified. This fact is achieved by the animation of the control points of the splines. However, these control points are not considered as weighty points but moreover as the degrees of freedom of the continuous object. The chosen dynamic equations (Lagrangian formalism) reflect this modelling idea and yield to an accurate and very efficient linear system. Suitable handling of forces and constraints, and numerical resolution are discussed in this scheme too.

Keywords: Simulation, Dynamic animation , Lagrangian equations, spline animation.

INTRODUCTION

Three dimensional parametric curves are important topics in Computer Graphics and Computer Aided Geometric Design (see [Watt 89] or [Farin 90] for example). Different natures of parametric curves have been proposed (for instance Bezier, B-spline, Catmull-Rom,…), differing mainly according to the geometrical properties they infer to the modelled object.

Nevertheless these models almost always consist in successions of curves (which we will call "spline segments") defined as parametric functional combinations of a set of control points.

When dynamic animation of such an object is required, it would be smart to compute this animation in a way insuring that the wanted geometrical properties are kept over time. One simple method then consists in animating the control points of the parametric curves.

This can be achieved with dynamic animation of weighty control points. We believe that the concentration of the object mass on a discrete set of 3D points is unrealistic and even more so if one remembers that some of these control points could not lie on the object.

Hence, we propose another method somehow similar in its assumptions to the D-NURBS proposed by D. Terzopoulos and H. Qin in [Terzopoulos 94] and [Qin 96]. This method is more accurate since it considers the control points as the degrees of freedom of the non discrete (say "continuous") object.

This paper follows a classical mechanics methodology. Starting with a defined mechanical system, we model kinematics and mass repartition. Then we choose Lagrangian formalism as the most suited dynamic equations and reshape this system in computationally efficient formulas. After which forces and constraints incidence in these equations are briefly and generally discussed, and a short presentation of classical numerical resolution methods follows.

Some results then show the validity of the method on examples.

SPLINE SEGMENTS KINEMATICS

The first step in mechanical modelling of an identified system concerns kinematics, and thus expressions of the positions and velocities of the material points.

The objects we aim to animate are continuous 3D curves defined as successions of *ns* spline segments collectively controlled by a set of *n* mobile control points $q_i(t)$.

The spline segments are defined as position functions of a parametric position *s* and the time *t* :

$$P_j : \; [0,1]\times I\!R \;\rightarrow\; I\!R^3$$
$$(s,t) \;\mapsto P_j(s,t) = \sum_{i\in[0,n[} b_i^j(s)q_i(t)$$

where $b_i^j(s)$ is the combination function of the control point q_i used by spline segment number *j*.

Hence an object material point is referenced by its spline segment number *j* and its parametric position *s* on this segment, see Fig.1.

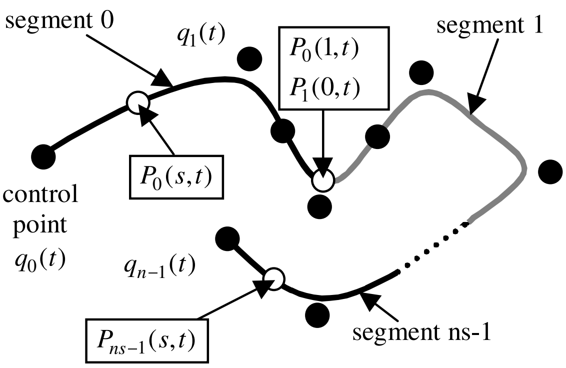


Figure 1

Therefore, the kinematics of the object are defined by the position functions P_j and their first time derivatives expressing the velocities of the material points :

$$\dot{P}_j(s,t) = \sum_{i\in[0,n[} b_i^j(s)\dot{q}_i(t)$$

As one can see, the kinematics of the whole object are defined by the finite set of the *n* 3D functions $q_i(t)$. These functions can be seen as the *3n* degrees of freedom of our continuous mechanical system.

WEIGHTY SPLINE MODELLING

The second step in the mechanical modelling of the system concerns mass repartition over the object.

In the particular case of spline like objects we consider that thickness is negligible and so, model mass repartition as *ns* mass density functions (one per spline segment) :

$$\rho_j : \; [0,1]\times I\!R \;\rightarrow\; I\!R$$
$$(s,t) \;\mapsto \rho_j(s,t)$$

These mass density functions should not be considered as giving mass density per unit length but rather per unit parametric variation of *s*. Hence the mass of a small part *ds* around point *s* in segment number *j* is $\rho_j(s,t)\,ds$.

As spline segment lengths can evolve in time, this implies that mass density per unit length can evolve too, even if $\rho_j(s,t)$ is actually invariant in time (this time invariance simply meaning that matter is not gliding along the curve and is tied to its parametric position *s*).

DYNAMIC EQUATIONS

Kinematics and mass repartition being modelled, we now have to choose the dynamic equations most suited to the nature of the system.

Since the mechanical system is continuous and non rigid with a finite number of degrees of freedom (*3n*), analytic mechanics seem best suited than Euler or Newton/Euler formalisms dealing respectively with weighty points and solids. We therefore choose to use Lagrangian equations :

$$\forall\, i \in [0,n[,\; \alpha \in \{x,y,z\}$$
$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i^\alpha}K - \frac{\partial}{\partial q_i^\alpha}K = Q_i^\alpha - \frac{\partial}{\partial q_i^\alpha}E$$

where (see [Germain 86]) :

- $K(q,\dot{q},t)$ is the kinetic energy function,
- $q_i^\alpha(t)$ are the degrees of freedom,
- E is the potential energy of the external forces deriving from a potential
- Q_i^α are the power rating of the other external forces in the virtual movement instilled by q_i^α.

The kinetic function can be expressed as :

$$K(q,\dot{q},t) = \tfrac{1}{2} \sum_{j\in[0,ns[} \int_0^1 \rho_j(s,t)\, [\, \sum_{i\in[0,n[} b_i^j(s)\dot{q}_i(t)\,]^2 ds$$

Then, the left terms of the equations become :

$$\frac{\partial}{\partial q_i^\alpha} K = 0$$

and

$$\frac{\partial}{\partial \dot{q}_i^\alpha} K = \sum_{j\in[0,ns[} \int_0^1 \rho_j(s,t)\, [\, \sum_{l\in[0,n[} b_l^j(s)\dot{q}_l^\alpha(t)\,] b_i^j(s)\, ds$$
$$= \sum_{j\in[0,ns[} \sum_{l\in[0,n[} [\int_0^1 \rho_j(s,t)\; b_l^j(s) b_i^j(s)\, ds]\, \dot{q}_l^\alpha(t)$$

Under the supplementary hypothesis that mass density functions are invariant in time (i.e. matter is not gliding along the curves), this last term can be simplified as :

$$\frac{\partial}{\partial \dot{q}_i^\alpha} K = \sum_{l\in[0,n[} M_{il}\, \dot{q}_l^\alpha(t)$$

with $M_{il} = \sum_{j\in[0,ns[} \int_0^1 \rho_j(s)\; b_l^j(s) b_i^j(s)\, ds$ constant over time.

Lagrangian equations then yield to the following linear equations system of the unknowns $\ddot{q}_i(t)$:

$$M\, \ddot{q}^\alpha = W^\alpha \qquad \alpha \in \{x, y, z\}$$

where M is the n square symmetric matrix filled with elements M_{il} and the right term combines all forces incidences :

$$W_i^\alpha = Q_i^\alpha - \frac{\partial}{\partial q_i^\alpha} E\,.$$

Some important facts should be noted concerning this equations system :

it is formally accurate (i.e. it implies no methodological numerical error such as those brought by a purely numerical resolution of Lagrangian equations) and nevertheless, requires no symbolic computation.

it is composed of 3 independent systems, one for each scene axis (x,y,z) .

these 3 systems use the same matrix M .

this matrix is constant over time and can be built and inverted once at the beginning of the animation process.

FORCES

In order to change the kinematics of the object, forces have to be exerted on it and these forces should be handled as suited in the chosen dynamic equations.

In Lagrangian equations, forces are introduced in the system either by their contribution to the potential energy E or by their power ratings in the virtual movements instilled by the degrees of freedom.

Let's show these two cases upon two classical force fields : gravity and viscosity.

Gravity

Gravity forces obviously derive from the following potential with z as the vertical axis :

$$E_{grav} = \sum_{j\in[0,ns[} \int_0^1 \rho_j(s,t) \sum_{i\in[0,n[} b_i^j(s) q_i^z(t)\; ds$$

and thus their contributions to the right terms of the equations are :

$$W_{grav}{}_i^x = -\frac{\partial}{\partial q_i^x} E_{grav} = 0$$
$$W_{grav}{}_i^y = -\frac{\partial}{\partial q_i^y} E_{grav} = 0$$
$$W_{grav}{}_i^z = -\frac{\partial}{\partial q_i^z} E_{grav}$$
$$= -\sum_{j\in[0,ns[} \int_0^1 \rho_j(s,t)\; b_i^j(s)\; ds$$

Under the usual hypothesis that mass density functions are invariant in time, these terms are invariant in time too and can be computed once at the beginning of the process.

Viscosity

Viscosity forces do not derive from a potential, hence their power ratings in the virtual movements have to be expressed.

The force exerted upon a small part ds of the curve around the material point at parametric position s on segment number j is :

$$F_j(s,t) = -C\,\dot{P}_j(s,t)\,ds$$
$$= -C \sum_{l\in[0,n[} b_l^j(s)\dot{q}_l(t)\,ds$$

where C is the viscosity coefficient.

The velocity of this material point in the virtual movement instilled by q_i^α is :

$$\hat{\dot{P}}_j^{i,\alpha}(s,t) = \sum_{l\in[0,n[} b_l^j(s)\frac{\partial q_l}{\partial q_i^\alpha} = b_i^j(s)\begin{bmatrix}\delta_{\alpha x}\\ \delta_{\alpha y}\\ \delta_{\alpha z}\end{bmatrix}$$

where $\delta_{\alpha\beta}$ is the Kronecker's symbol

and thus the total contribution of viscosity forces is expressed by :

$$W_{visc}{}_i^\alpha = Q_{visc}{}_i^\alpha = \sum_{j\in[0,ns[} \int_0^1 F_j(s,t)\,\hat{\dot{P}}^{i,\alpha}{}_j(s,t)$$
$$= -C \sum_{j\in[0,ns[}\sum_{l\in[0,n[} \dot{q}_l^\alpha(t)\int_0^1 b_l^j(s)b_i^j(s)\,ds$$
$$= -C \sum_{l\in[0,n[} \dot{q}_l^\alpha(t) \sum_{j\in[0,ns[}\int_0^1 b_l^j(s)b_i^j(s)\,ds$$
$$= \sum_{l\in[0,n[} \dot{q}_l^\alpha(t)\ \Phi_{il}$$

which can be summarised as :

$$W_{visc}{}^\alpha = \Phi\,\dot{q}^\alpha(t)$$

where Φ is a constant n square symmetric matrix filled with :

$$\Phi_{il} = -C \sum_{j\in[0,ns[}\int_0^1 b_l^j(s)b_i^j(s)\,ds\ .$$

CONSTRAINTS

Control of the dynamics is partly achieved by the exertion of external forces, but in many practical cases supplementary constraints are imposed on the movement. These constraints are a smart way of specifying things like "this point should not move", "this point should follow a specific trajectory over time", or many more kinds of movement restrictions.

Constraints are usually handled in Lagrangian formalism by one of the two following methods :

- the penalty method (see [Platt 92] for details), which roughly consists in adding viscous specialised springs aiming to push constraints back to realisation.

This method is computationally efficient since it only adds forces and does not increase the system size, but constraints are only rarely accurately realised during the animation. Moreover, this method can lead to inaccurate equilibrium states as the constraints springs responses required to balance the other forces imply that constraints are not realised.

- Lagrange multipliers, an accurate constraints handling tool provided by the Lagrangian formalism [Germain 86]. Using this tool, the equations classically become :

$$\forall\ i\in[0,n[,\ \alpha\in\{x,y,z\}$$
$$\frac{d}{dt}\frac{\partial}{\partial\dot{q}_i^\alpha}K - \frac{\partial}{\partial q_i^\alpha}K = W_i^\alpha + \sum_{c\in[0,nc[} L_{ci}^\alpha\,\lambda_c$$

$$\forall\ c\in[0,nc[$$
$$\sum_{\alpha\in\{x,y,z\}}\sum_{l\in[o,n[} L_{cl}^\alpha\,\ddot{q}_l^\alpha = D_c(q,\dot{q})$$

where the second equation set expresses the nc constraints upon the unknowns, and λ_c is the Lagrange multiplier of the constraint number c, this value is related to the reaction force needed to keep the constraint realised.

In this scheme, constraints are accurately realised but the system size is increased by the number of scalar constraints nc.

In order to choose a constraint handling method, let's express our linear system with the more accurate solution (i.e. Lagrange multipliers) and evaluate the involved cost :

$$\begin{bmatrix} M & & & -L^{xT} \\ & M & & -L^{yT} \\ & & M & -L^{zT} \\ L^x & L^y & L^z & 0 \end{bmatrix} \cdot \begin{bmatrix} \ddot{q}^x \\ \ddot{q}^y \\ \ddot{q}^z \\ \Lambda \end{bmatrix} = \begin{bmatrix} W^x \\ W^y \\ W^z \\ D \end{bmatrix}$$

If the L^α are time independent the whole matrix can be built and inverted once. Thus the price for accuracy is reduced and Lagrange multipliers method is practically usable.

If the L^α vary over time, the size increase and, worst, the time dependence of the whole matrix could impose the less accurate but more efficient penalty method.

NUMERICAL RESOLUTION

Classically rewritten as a Cauchy problem the linear system becomes :

$$\dot{\xi} = F(\xi, t) = \begin{bmatrix} \dot{q} \\ f(q, \dot{q}, t) \end{bmatrix}, \qquad \xi = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

where f is the numerical function solving the linear system of the unknowns $\ddot{q}$.

The temporal integration of this system is a classical mathematical problem for which a set of well known solutions exist, (see [Press 88] : Euler, prediction/correction, Runge-Kutta second or fourth order methods, …). These methods differ according to their precision, the number of evaluations of F (i.e. resolution of the system) per time step and their ability to automatically adapt the time step as needed by the actual temporal "regularity" of the movement.

The later property (i.e. ability to adapt time step automatically) is quite relevant since it permits to either slow down resolution in order to properly handle a singularity, or to speed up resolution when movement is regular enough.

According to the precision and efficiency needed, one can choose any one of these (or other) methods.

SOME RESULTS

This work was initially developed in the domain of knitted cloth modelling. A previous study ([Nourrit 96]) demonstrated a static modelling of stitches. In this model yarns were defined as successions of Catmull-Rom spline segments [Catmull 74]. Wishing to obtain a mechanical model of knitted cloth, we came to the idea of animating these Catmull-Rom spline segments. However the idea has been generalized as shown in this paper. And the initial goal is achieved as a particular case of this general scheme. Hence the examples shown below (fig 2, 3 and 4) use Catmull-Rom spline segments.

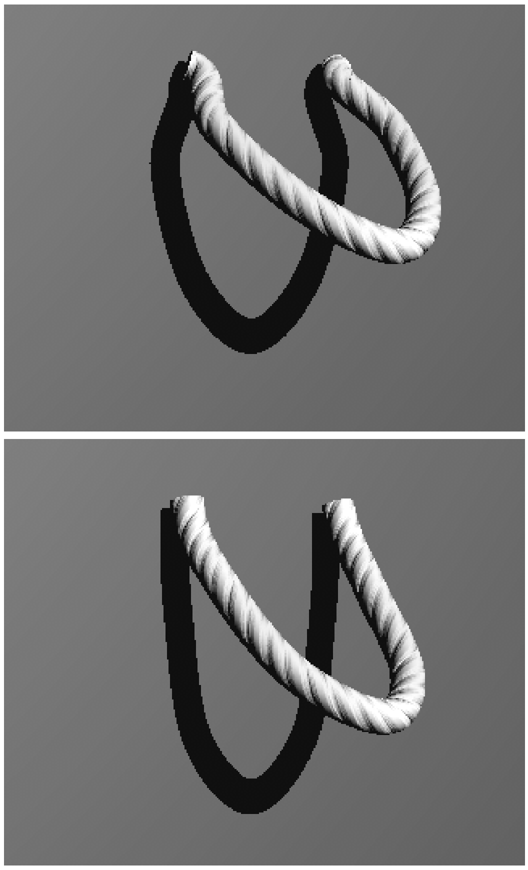
In these examples, forces used are gravity, viscosity, and non-linear springs similar to those in [Nocent 97], approximating the non-linear elasticity of the rope and the yarns, set between successive control points.

Furthermore point trajectory constraints are set in order to make some control points motionless or following a pre-set trajectory. These constraints lead to time independent matrices L^α and are handled by Lagrange multipliers.

Numerical resolution is achieved by a fourth order Runge-Kutta method using a fixed time step.

Rope example

In this example (figure 2), a rope is dropped without velocity from the position shown in the first image, while its extremities are held solid by constraints.



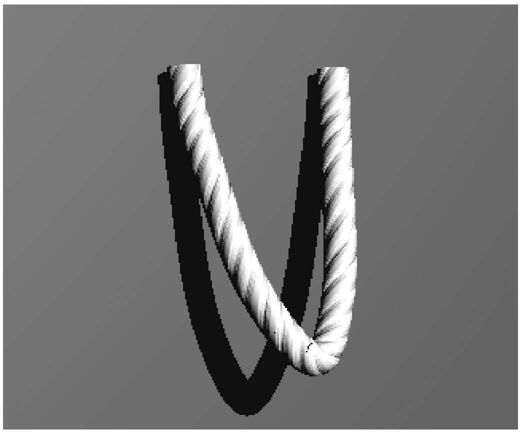


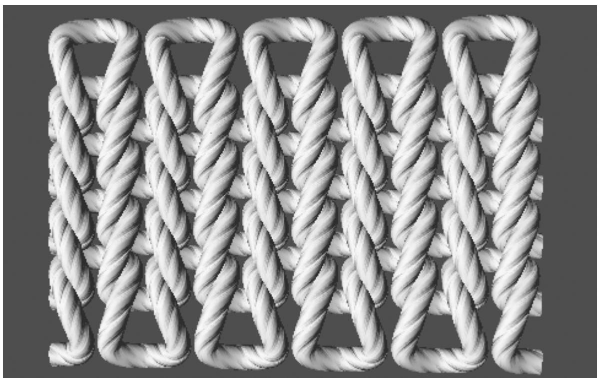
Figure 2 : Rope ; t = 0s, 0.08s, 0.24s

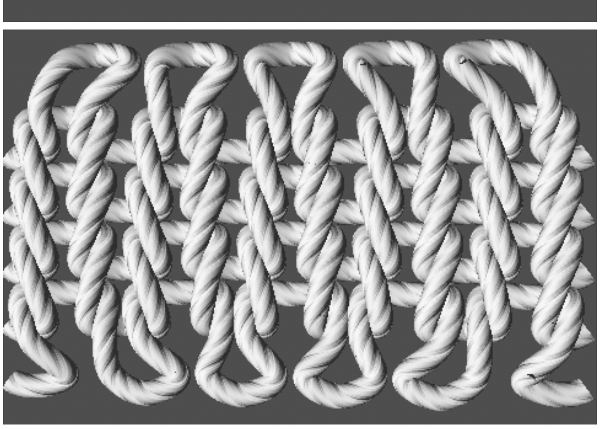
Knitted Clothe examples

The frames in figures 3 and 4 were taken from animations computed for knitted clothes respectively 5 and 10 stiches both large and high.

Stitch binding modelling is obtained by specific forces and constraints, the description of which is beyond the scope of this paper.

These clothes are held by their upper and lower control points and stretched with periodical trajectories at all their side control points with a common 2 Hz frequency.





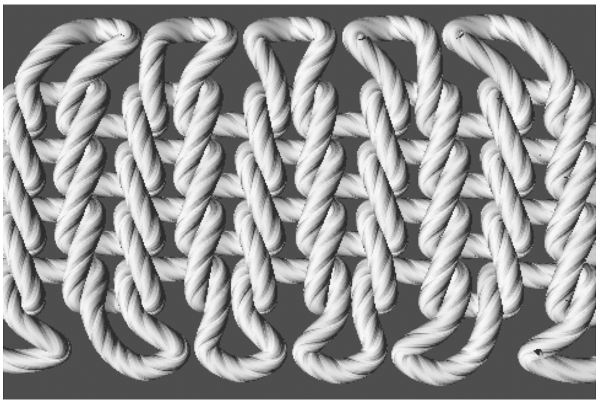


Figure 3 : Jersey 5 × 5; t = 0s, 0.16s, 0.24s[1]

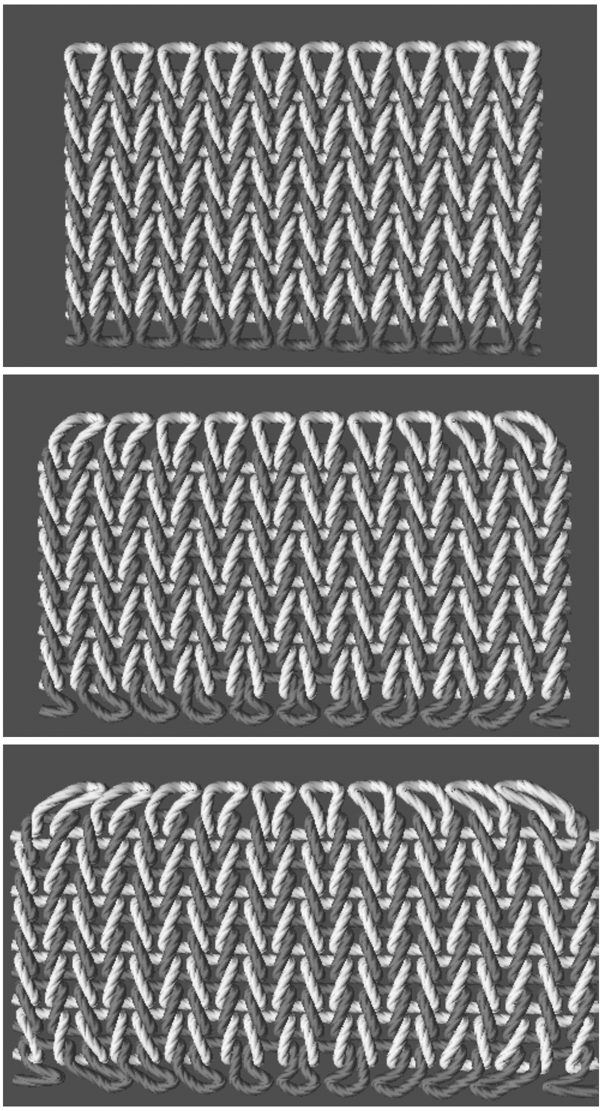


Figure 4 : Jersey 10 × 10 ; t = 0s, 0.16s, 0.24s[1]

[1] These frames are taken from movies presented at http://www.univ-reims.fr/leri/membre/jmn/anim.html

CONCLUSION

Through this paper we have presented a generic dynamic animation engine handling thin curve-like objects. These objects are modelled as successions of spline segments collectively controlled by a set of control points. These control points are considered as the degrees of freedom of the system rather than weighty points, hence the dynamics computed are those of continuous objects, and even field forces such as gravity or viscosity are actually considered over the continuous object.

The only methodological approximation exists in the assumption that the object shape always conforms to the chosen spline modelisation. Such an hypothesis seems necessary since the number of degrees of freedom has to remain finite for a practical resolution.

Furthermore this assumption insures that the geometrical properties implied by the spline model are kept over time ; this can represent a smart ability if some of these geometrical properties were important topics for the object designer.

Moreover, the proposed dynamic animation engines most important properties are :

- **generality** : the spline segments succession model is generic enough to encompass a great deal of classical 3D curves modelisations
- **accuracy** : the actual continuous objects are handled by accurate numerical equations
- **efficiency** : the linear system to be solved has very interesting properties (axes separation and common constant matrix for the 3 axes)
- **versatility with respect to constraints handling** : according to their properties and number, a suitable constraints handling method has to be chosen ; the choice between Lagrange multipliers or penalty methods has been discussed.

PROSPECTS

As this paper clearly shows, different spline types can be handled and more have to be practically integrated for increased versatility.

Furthermore, the extension of the theory to smooth surfaces or volumes defined as parametric functional combinations of control points seems straightforward and attractive.

ACKNOWLEDGMENTS

This work has been supported by a research technological allowance of the “European Social Fund” in the framework of an industrial collaboration with “L’Institut Textile de France” in Troyes, France. The authors would like to thank both the programmation comity of ESF and ITF Troyes for their support.

REFERENCES

Catmull E., Rom R. 1974 *A class of local interpolating splines*, Computer Aided Geometric Design, Academic Press.

Farin G. 1990 *Curves and surfaces for Computer Aided Geometric Design a practical guide*, Academic Press, second edition.

Germain P. 1986 *Mécanique X*, tome I, Ecole Polytechnique, Ellipses.

Nocent O. 1997 *Animation réaliste de textiles*, Cinquièmes journées de l’A.F.I.G., pp. 227-235.

Nourrit J.M., Desjardin E., Secroun C. 1996 *Modélisation de textiles à base de mailles*, IVèmes Journées de l’AFIG, pp. 52-60.
Also in : Revue internationale de CFAO et d’informatique graphique, vol. 12, n°4, pp. 373-385, septembre 1997.

Platt J. 1992 *A generalization of dynamic constraints*, Graphical Models and Image Processing, Vol. 54, No. 6, pp. 516-525.

Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., 1988 *Numerical Recipes in C*, Cambridge University Press.

Qin H., Terzopoulos D., 1996 *D-NURBS : A physics-based geometric design framework*, IEEE Transactions on Visualization and Computer Graphics,2(1), March, 1996, pp. 85-96.

Terzopoulos D., Qin H., 1996 *Dynamic NURBS with Geometric Constraints for Interactive Sculpting*, ACM Transactions on Graphics, 13(2), April, 1994, pp. 103-136.

Watt A. 1989 *Fundamentals of three-dimensional computer graphics*, Addison-Wesley.