

# TOWARD DEFORMATION MODELING WITH HAPTIC FEEDBACK

ZhuLiang Cai

John Dill

Shahram Payandeh

Graphics and Multimedia Research Laboratory, and Experimental Robotics Laboratory  
Center for System Science  
Simon Fraser University  
Burnaby, B.C., Canada V5A 1S6  
{zcai, dill, shahram} @cs.sfu.ca

## ABSTRACT

A fast deformation modeling method is presented. Based on elastic theory, an algorithm for animating viscoelastic systems is proposed. The integration scheme derived from implicit integration allows developers to obtain interactive realistic animation of models of viscoelastic objects. This method provides a simple, stable and tunable model for deformable object suitable for virtual reality environments. An implementation demonstrates this deformation analysis approach.

As an important application of virtual reality, Surgical simulation is becoming increasingly popular. Three major problems in surgical simulation are: 1) Collision detection. 2) Deformation animation. 3) Force feedback/haptic rendering.

In our prior work[18], we presented basically a collision detection algorithm for this application. In this paper, we focus on other problems: Deformation animation and force feedback/haptic rendering.

Haptic Rendering, by simulating the forces generated by contact with a virtual model, provides a means of interrogating the model by touch, an additional means of interacting with a virtual environment.

## 1.1 Review of previous work and relative papers

Based on elastic theory [19], various deformation-modeling methods have been studied in computer graphics. All methods are based on non-rigid/elastic physical models using different mathematical-physics approaches [5] to solve a series of differential equations. The set of methods can be classified into those that use traditional differential equation solution such as [1,2,3,6,7], those that use a mass-spring model [5,8,10,11] and those that use FEMs (finite element method) [4,9,12,15,16].

Terzopolous et al [1], and Platt, and Barr [2] have shown the advantages of physically based models over kinematic models for computer animation. The particular shape an elastic body is a function of both the internal stress and strain within the object and external forces applied to it. Generally, some modification or simplification based on numerical analysis methods is made to the elastic theory in order to give a particular behavior to the deformable body [2].

Among the representations used for deformable surfaces/volumes, the most widely used are parametric model with B-spline representation and finite element models introduced for computer aided design and applied to computer animation by Qin, Terzopolous and Gourret etc [13,14]. In this work, a physical based generalization of Dynamic Non-Uniform B-Spline (D-Nurbs) is presented that incorporates mass distribution, internal deformation energies, forces, strains and other physical quantities into the Nurbs geometric substrate. Gourret et al [14] described a system for modeling the human hand with a finite element volume meshed around bones. They formulated and solved a set of static equations for skin deformation based on bone kinematics and a hand/object contact point grasping task. Ma et al [17] gave a method for surface deformation based on deforming a hand surface. A bicubic B-spline surface was interpolated or approximated by key data points of a sensor glove to generate the deformation.

For precisely evaluating elastic deformation, the FEM is an efficient and powerful tool among the various numerical solution methods. S. Cotin et al [4,9] presented a FEM to simulate elastic object deformation. The physical properties are based on linear elasticity, and a pre-processing technique derived from a finite element method allows real-time computation of deformations and forces. S. Dombrowski et al [12] also used a particular FEM to simulate virtual surgery. This

FEM was based on thin plate theory and developed by A. Shabana [3]. O.R. Astley [20] used FEM to simulate viscoelastic three dimension bodies. A multi-layer mesh deformation problem can be solved by a finite element approach with hierarchy layer mesh structure. In J.P.Thirion's report [15], a detailed finite element method for volume variation to solve tissue deformation is described. Currently, Yan uses 2-D FEM to integrate with haptic rendering system in virtual reality area [30]. The FEMs, inherited from engineering using material properties to model structure, are a more sensible approach and have a precise accurate performance for deformation and plastic modeling. However, FEMs are less widely used due to the difficulty of their implementation and large computation time. Hence, some simplified methods have been studied as alternatives to the FEMs.

In modeling deformable objects, an important aspect of realistic animation, researchers have focused on mass-spring methods due to their simplicity of implementation and their less complex algorithms. Miller [8] proposed a model for animating leg-less figures such as snakes and worms using mass-spring systems. Animating spring tensions simulates muscle contractions. Due to the surface structure, directional friction is included in the dynamics model and leg-less figure locomotion results. Instead of real complex internal structures for natural snakes and worms, the simplified model proposed in Miller's work provides an elegant way to simulate the motion dynamics of these creatures.

Provot [5] proposes a model for animating cloth objects using a network of springs and masses. A cloth object is approximated to a deformable surface by using the numerical integration method with the fundamental law of dynamics. Provot also presents a new method, derived from dynamic inverse procedures, to adopt his model to the particular non-linear stiffness properties of tentacles.

The major advantages of the mass-spring model approaches are:

1. They are the simplest of physically-based modeling approaches.
2. They can achieve real-time performance.

A similar approach is the viscoelastic physical model, which has the same advantages as the mass-spring model but has other advantages as well, and is the method we use. Compared with the spring-mass model, the viscoelastic model can describe the elastic material's damping behavior, making the deformation more realistic. Due to its ease of implementation, the fact that it is parallelizable, and that its computational demands are low, it seems a good candidate for simple virtual reality applications.

The contributions of this paper are:

- Instead of a simple mass-spring model, a more realistic and accurate physical model, namely a viscoelastic model, is applied.
- A new fast discrete numerical method for elastic deformation analysis is presented.
- A simple approximation approach for weighting the force has been used which can offer a fast and more accurate deformation solution.

The paper is organized as follows: In section 2, a viscoelastic deformable system will be described with a detailed integration algorithm. In section 3, a force feedback subsystem that uses the deformable model is presented. We will discuss and illustrate our experiment results in section 4. For section 5, the conclusion and future steps will be presented.

## VISCOELASTICITY MODEL FOR DEFORMATION ANALYSIS

We begin the mathematical development by giving the equations of motion governing the dynamics of our deformable models under the influence of applied forces. The equations of motion are obtained from elastic mechanics which balance the externally applied forces with the forces due to the deformable model.

Much research has been focused on the elastic deformation of dynamic system [19,21,22]. In general, there are two methods to solve the deformation analysis: Calculus schemes such as [24], and partial differential schemes such as [25]. Both approaches are efficient classical mathematical physics methods. Our scheme uses an elastic dynamics system expression, based on partial differential equations (*Lagrange's* form) [4].

The equations governing a deformable model motion can be written in *Lagrange's* form [23,26] as follows:

$$\frac{\partial}{\partial t} \left( \mu \frac{\partial r}{\partial t} \right) + \gamma \frac{\partial r}{\partial t} + \frac{\partial \mathcal{E}(r)}{\partial \gamma} = f(r, t) \quad (1)$$

where  $a$  is the coordinate of an elastic body  $\Omega$ .  $r(a, t)$  is the position of the particle  $a$  at time  $t$ ,  $\mu(a)$  is the mass density of the body at  $a$ ,  $\gamma(a)$  is the damping density, and  $f(r, t)$  represents the

net externally applied forces and  $\mathcal{E}(r)$  is a functional which measures the net instantaneous potential energy of the elastic deformation of the body.

Expression (1) for an elastic object is continuous in the material coordinates of the deformable surface. For simulating the dynamics of the elastic model, (1) should be discretized by applying finite element or finite difference approximation methods [25]. Discretization transforms the partial differential equations of motion (1) into a system of linked ordinary differential equations. We illustrate the discretization step using finite difference approximations.

Consider, for simplicity, the case of time invariant mass density  $\mu(a, t)=\mu[a_1, a_2]$  and damping density  $\gamma(a, t)=\gamma[a_1, a_2]$  in equation (1). The resulting discrete densities are  $\mu[n]$  and  $\gamma[n]$ . Let  $M$  be the mass matrix, a diagonal  $N \times N$  matrix with the  $\mu[n]$  variables as diagonal components, and let  $C$  be the damping matrix ( $N \times N$ ) constructed similarly from  $\gamma[n]$ . The discrete form of the equations of motion (1) can be expressed in mesh vector form using spring forces by the following coupled system of second-order ordinary differential equations:

$$M \frac{d^2 r}{dt^2} + C \frac{dr}{dt} + K(r) = F \quad (2)$$

where  $F$  is the vector representing the discrete net external forces.

## 2.1 INTEGRATION METHOD

To simulate the dynamics of an elastic model, the system of differential equations (2) is integrated through time. In fact, the numerical integration method is a numerical step-by-step procedure, which converts the system of non-linear ordinary differential equations into a sequence of linear algebraic systems.

Various integration schemes can be used to solve the elastic dynamics equations: *explicit Euler* method, *mid-point* integration, *Runge-Kutta* (higher-order scheme) method, etc. had been used in solving differential equations. Each has its limitations.

The *explicit Euler* and *mid-point* methods [24] require the square of the integration time step  $dt$  to be inversely proportional to the stiffness (Courant condition [21]). Otherwise, the system will produce large errors.

The *Runge-Kutta* method is not appropriate for collision handling [27]. But for virtual-reality systems, dynamic collision detection must be considered.

Semi-implicit integration [28] which has a smooth result and numerical accuracy can be considered as an ideal scheme. However, it requires longer run-times and results in more complex algorithms.

*Implicit Euler* integration has proven to be better adapted to solve such deformation problems [29]. The idea is to replace the forces at time step  $t$  by the forces at time step  $t+1$ . Based on the implicit method, we obtain an increment of the velocity vector:

$$\begin{aligned} \Delta V^{n+1} &= (r^{n+1} - r^n) / \Delta t \\ \Delta V^{n+1} &= W (F^n + KV^n \Delta t + CV^n) \frac{\Delta t}{m} \\ \text{where } W &= \left( 1 - K \frac{\Delta t^2}{m} - C \frac{\Delta t}{m} \right)^{-1} \end{aligned} \quad (3)$$

If  $F^n + KV^n \Delta t + CV^n$  is defined as the node force  $F_{node}$ , it is found that the corresponding change of velocity through multiplication of the node force  $F_{node}$  by the inverse of a constant matrix:

$$W = \left( 1 - K \frac{\Delta t^2}{m} - C \frac{\Delta t}{m} \right)$$

(where  $w_{ij}$  is an arbitrary element of the  $W$  matrix) reflects the relationship between the  $j^{\text{th}}$  particle's force and  $i^{\text{th}}$  particle's strain force. Equation (3) actually becomes a force expression when both sides of (3) are multiplied by  $m/\Delta t$ .

$$F_{weight} = WF_{node}$$

According to elastic theory [19],  $F_{weight}$  can be regarded as Green strain tension (because all components of  $F_{node}$  are entirely along the object's surface plate. Hence,  $F_{weight}$  is defined as the strain force  $F_{strain}$ .

For a discrete particle system, the strain force for the  $i^{\text{th}}$  particle can be written as:

$$F_{strain} = \sum_1^N w_{ij} F_j$$

## 3 IMPLEMENTATION

Although general elastic theory and numerical integration can help us solve the deformation analysis, there are some problems in developing a practical implementation.

First, how should we select a suitable physical model for the deformable objects? Second, how should we select a suitable geometry representation approach for the deformable objects? Finally, how should we implement the deformation animation based on our numerical-physical method as illustrated in following subsections?

Viscoelasticity is generalization of elasticity and viscosity. It is characterized by the phenomenon of creep which manifests itself as a time dependent deformation under constant applied force.

The ideal linear viscous unit is the dashpot. A dashpot is supposed to produce a velocity proportional to the load at any instant. ( $c$  is the viscosity constant and  $\dot{u}$  is linear velocity)

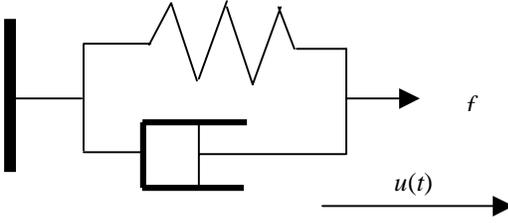
The elastic and viscous units are combined to model linear viscoelasticity, so that the internal forces depend not just on the magnitude of deformation, but also on the rate of deformation. Fig. 1 illustrates a two-linear-unit viscoelastic model, a parallel assembly of the so-called *Voigt* viscoelastic model. The local deflection relationship for the Voigt model is:

$$F = k \cdot u(t) + c \cdot \dot{u}(t)_{u(0)=0}$$

A creep function for the *Voigt* model can be easily derived by solving this equation when  $F(t)$  is a unit-step function  $l(t)$ . This is

$$u(t) = l/k (1 - e^{-(k/c)t}) l(t)$$

Physically,  $u(t)$  is the elongation caused by a unit step function. As described above under deformation modeling, the viscoelastic model (also called spring-damping model in a discrete system) is applied as the deformable material property model. Other viscoelastic models (Kelvin, Maxwell [33]) are possible as well. Among the rectangular grid and polygonal meshes, triangle mesh is fast, memory efficient and robust for the deformable analysis approach. As is well known, guaranteed quality 2-D meshes are generated using a *Delaunay triangulation* refinement algorithm [31].



**Figure 1: Uniaxial viscoelastic model: two-element Voigt viscoelastic model**

### 3.1 IMPLEMENTATION DETAILS

A deformation animation algorithm has been successfully implemented in C++. The system combines an implicit integration algorithm with OpenGL™, a computer graphics library, and third-party freeware to generate a mesh. Here some of the implementation issues are described.

The current system runs on one computer and employs a haptic device ProCAT™ (a 2D force feedback device). The system consists of three modules: Deformation Animation (graphics rendering), Collision Detection, and Haptic Rendering (force feedback).

Multiple external forces with multiple constraints can be processed as serial external force collisions in one simulation step. External forces can be changed using a mouse.

The system detects a collision as the cursor is moved to the object. External forces are shown as arrows and are updated in the process. And as the inverse dynamics process keeps the viscoelastic system from being over-stretched, it enhances the realism of the overall animation. The algorithm also takes into account the effect of force producing torques ( $\delta T$ ). Pseudo-code for the deformation analysis algorithm is shown in Fig. 2. Note that our algorithm to calculate the matrix  $W$  utilizes the positive definite nature of the matrix to be inverted [32].

### 4 EXPERIMENT AND RESULTS

We implemented 2D and 3D versions of the algorithm.. Fig. 3 shows three frames of a thin polygonal plate being deformed by a single step function force. The frames show the initial state and the deformation after (approximately) 5 and 10 sec. The object has 517 nodes and the simulation ran at 25 frames/sec. The same object with four applied forces ran at the

same rate. A similar object but with a coarser mesh (123 nodes) ran at 100 frames/sec. The simulation was run on an NT workstation with a Pentium III-450 CPU and 128MB of memory.

$$\text{Precompute } W = \left( I - K \frac{\Delta t^2}{m} - C \frac{\Delta t}{m} \right)^{-1}$$

At each  $\Delta t$  calc. int. forces  $F_i$  due to springs, viscosity and damping:

$$X_G = 0$$

For each mass point  $i$

$$F_i = 0$$

For each neighbor mass point  $j$

$$F_i = F_i + k_{ij} \left( |x_i - x_j| - l_0^{ij} \right) \frac{x_i - x_j}{|x_i - x_j|}$$

$$F_i = F_i + k_{ij} \Delta t (v_j - v_i)$$

$$F_i = F_i + c_{ij} (v_j - v_i)$$

$$X_G = X_G + x_i$$

$$X_G = X_G / n; \quad \delta T = 0;$$

For each mass point  $i$ , calc. approx. strain forces:

$$F_i^{filter} = \sum_j F_j W_{ij} \quad (\text{point } j \text{ is point } i\text{'s neighbor})$$

$$\delta T = \delta T + F_i^{filter} \times x_i$$

$$v_i^{n+1} = v_i^n + \left( F_i^{filter} + F_i^{ext} \right) \frac{\Delta t}{m}$$

$$x_i^{new} = x_i + v_i^{n+1} \Delta t$$

For each mass point  $i$  (to preserve coupled momentum)

$$F_i^{correct} = (x_G - x_i) \times \delta T \Delta t$$

$$x_i^{new} = x_i^{new} + F_i^{correct} \frac{\Delta t^2}{m}$$

For each mass point  $i$ : (post-step update)

For each neighbor point  $j$

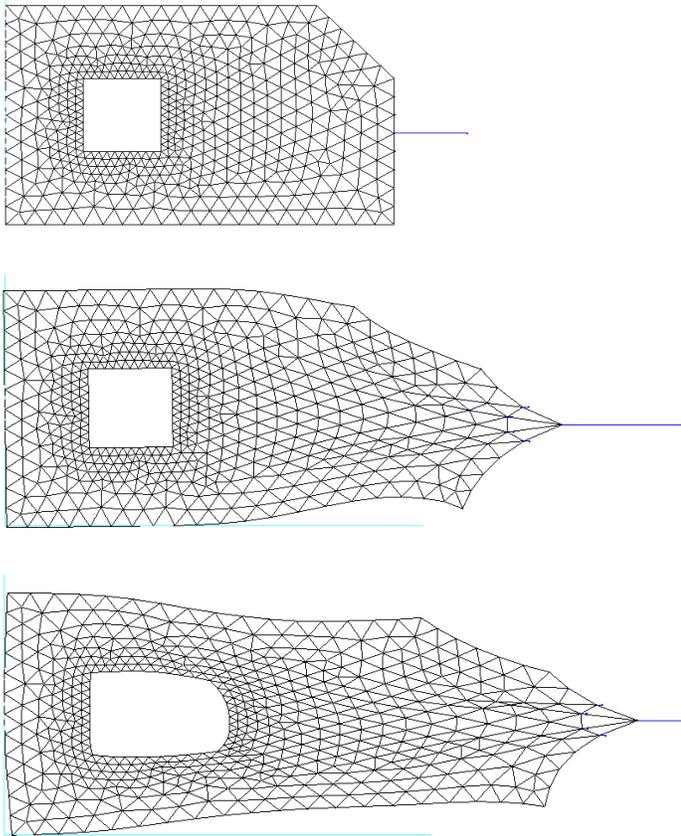
$$\text{if } \left( \frac{|x_i - x_j|}{l_0^{ij}} \right) > \text{threshold}$$

if point  $i$  not constrained

change  $x_i$  to ideal position

update real velocity, position

**Figure 2: Deformation Algorithm pseudo-code.**



**Figure 3: Initial shape and deformation of polygonal object after 5 sec and 10 sec. The force is indicated by the horizontal line to the right.**

## 5 CONCLUSIONS

We have presented a new algorithm to animate efficiently any viscoelastic deformable system. As we can perform the time integration for any time step size with a constant computational time, this method is well-suited to real-time interaction in Virtual Reality environments. This algorithm is an approximated implicit integration scheme, but important physical quantities such as linear and angular momentum, vital for realism, are preserved. This technique can thus be classified as an implicit predictor/corrector scheme. It handles constraints and collisions in a very simple way and uses non-linear Voigt viscoelastic deformable objects that enhance the visual behavior of the motion.

## REFERENCES

- [1] D. Terzopoulos, K. Fleischer, 1988, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture," *ACM Computer Graphics*, 22(4) August 1988, 269-278.
- [2] A. Witkin, K. Fleischer, A. Barr, 1987, "Energy Constraints On Parameterized Models," *ACM Computer Graphics*, 21(4), July 1987, 225-232.
- [3] A. A. Shabana, A.P. Christensen, 1997, "Three-Dimensional Absolute Nodal Co-ordinate Formulation: Plate Problem," *International Journal for Numerical Methods in Engineering*, Vol.40, 1997, 2775-2790.
- [4] S. Cotin, H. Delingtte, 1998, "Real-time Surgery Simulation with Haptic Feedback using Finite Elements," *Proceedings of 1998 IEE, International Conference on Robotic & Automation*, May 1998, 3739-3744.
- [5] X. Provot, 1995, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior," *Graphics Interface '95*, 147-154.
- [6] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, 1987, "Elastically Deformable Models," *ACM Computer Graphics*, 21(4) July 1987, 205-214.
- [7] J. Platt, A. Barr, 1988, "Constraints Methods for Flexible Models," *ACM Computer Graphics*, 22(4) August 1988, 279-288.
- [8] G.S.P. Miller, 1988, "The Motion Dynamics of Snakes and Worms," *ACM Computer Graphics*, 22(4) August 1988, 169-173.
- [9] S. Cotin, H. Delingette, N. Ayache, 1999, "Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Trans. Visualization and Computer Graphics*, 5(1), January-March, 1999, 62-73.
- [10] U. Gudukbay, B. Ozguc, Y. Tokad, 1997, "A Spring Force Formulation For Elastically Deformable Models", *Computer & Graphics*, 21(3), 1997, 335-346.
- [11] M. Desbrun, P. Schroder, A. Barr, 1999, "Interactive animation of structure deformable objects," *unpublished paper, Caltech*, 1999.
- [12] S. von Dombrowski, R. Schwertassek, A. Shabana, J. Vogel, G. Hirzinger, 1998, "FEM Simulation of Deformable Objects for Virtual Surgery," [http://www.op.dlr.de/FF-DR-RS/MEDICINE/fem\\_sim.html](http://www.op.dlr.de/FF-DR-RS/MEDICINE/fem_sim.html).
- [13] H. Qin, D. Terzopoulos, 1996, "D-NURBS: A Physics-Based Framework for Geometric Design," *IEEE Trans. Visualization and Computer Graphics*, 2(1), March 1996, 85-96.
- [14] J. P. Gourret, N. M. Thalmann, D. Thalmann, 1989, "Simulation of object and human skin deformations in a grasping task." *Computer Graphics* 23 (3), pp. 21-30.
- [15] J-P. Thirion, G. Calmon, "Analyse de deformations pour detecter et quantifier les lesions evolutives dans les sequences d'images 3D," INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France) ISSN 0249-6399.
- [16] N.M. Briggs, N.J. Avis, "Real Time Soft Tissue Modeling," unpublished report, *The Centre for Virtual Environments, TIME Research Institute, University of Salford*.
- [17] L. Ma, R.W.H. Lau, J. Feng, Q. Peng, J. Wong, 1997, "Surface Deformation Using the Sensor Glove," *ACM Symposium on Virtual Reality Software and Technology*, 1997, 189-195.
- [18] Z.L. Cai, J. Dill, S. Payandeh, 1999, "Haptic Rendering: Practical Modeling and Collision Detection," *IMECE99/DSCD*,

*Proceedings of the ASME Virtual Environment and Teleoperator System symposium* Nov. 1999. Nashville, pp. 81-86.

[19] Y.C. Fung, 1965, *Foundations of Solid Mechanics*, Prentice-Hall, Inc, 1965.

[20] O.R. Astley, V. Hayward, 1998, "Multirate Haptic Simulation Achieved by Coupling Finite Element Meshes through Norton Equivalents," *Proc. 1998 IEEE International Conference on Robotic & Automation*, May 1998, 989-994.

[21] O.Z. Zienkiewicz, R.L. Taylor, 1991, "The Finite Element Method", *McGraw-Hill*, 1991.

[22] J.M.T. Thompson, G.W. Hunt, 1973, *A General Theory of Elastic Stability*, Wiley & Sons, NY, 1973.

[23] R. Courant and D. Hilbert, 1953, "Methods of Mathematical Physics", *Interscience I*, London, 1953.

[24] I.M. Gelfand., and S.V. Fomin, 1965, *Calculus of Variations*, Prentice-Hall, Englewood Cliffs, NJ, 1965.

[25] L. Lapidus., and G.F. Pinder., 1982, *Numerical Solution of Partial Differential Equations in Science and Science*, Wiley, New York, NY, 1982.

[26] H. Goldstein., 1950, *Classical Mechanics*, Addison-Wesley, Reading, MA, 1950.

[27] J.R. Cash, A.H. Karp, 1990, "A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Right Hand Sides", *ACM Trans. Math Software*,. 16(3), 1990, 201-222.

[28] G. Bader, P. Deulhard, 1983, "A Semi-Implicit Rule for Stiff Ordinary Differential Equations", *Number Math.*,. 41, 1983., 373-398.

[29] W.F. Ames, 1977, *Numerical Methods for Partial Differential Equations*, 2<sup>nd</sup> ed., Academic Press, 1977.

[30] J. Yan, P.K. Scott, R.S. Fearing, 1999, "Inclusion Probing: Signal Detection and Haptic Playback of 2D FEM and Experiment Data," *IMECE99/DSCD, Proceedings of the ASME Virtual Environment and Teleoperator System symposium* Nov. 1999. Nashville, pp. 203-210.

[31] R.S. Jonathan, 1996 "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator" *First Workshop on Applied Computational Geometry (Philadelphia, PV)*, Assoc. for Computing Machinery, May 1996, 124-133.

[32] A. Jennings, Alan., *Matrix computation*, John Wiley & Sons, New York, 1992.

[33] D. Terzopoulos, K. Fleischer, 1988, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fractures", *Computer Graphics*,. 22,(4), Aug. 1988, pp. 269-278