A framework for interactive and physically realistic cloth simulation

- Report by Jonathan Rubin for COMPSCI 780 SS C.
- Supervised by Dr Burkhard Wünsche.

1. Abstract

Cloth simulation is an important research area in the field of deformable object modelling in computer graphics. Deformable objects allow much greater realism to be achieved in graphically generated scenes. However, in interactive environments rigid bodies usually dominate due to their computational efficiency. Cloth simulation addresses many issues in computer graphics such as deformable object modelling, numerical stability and computational constraints. In the past there have been many methods proposed to deal with deformable objects in general. Each of the current approaches has their various benefits and deficits. For instance, *Mass-Spring Systems* are simple to implement and are reasonably efficient, however when used with explicit integration methods they suffer from stability issues. On the other hand the *Finite Element Method* provides very realistic results, but due to its computational complexity it cannot be used in interactive applications. We have chosen to investigate the application of a novel technique for simulating deformable objects (*Meshless Deformations*) to the area of cloth simulation. We began by examining a standard *Mass-Spring System* approach to cloth simulation so that we could compare the results of our novel application. We have also developed a hybrid system which combines various aspects from the above two approaches. We have found that applying *Meshless Deformations* to the area of cloth simulation can produce good results for small deformations, however it is unsatisfactory when applying a moderate to large deformation. Our hybrid approach improves on some of the undesirable animation produced and also takes advantage of the computational speed and numerical stability that *Meshless Deformations* offers.

2. Contents

1. Abstract………………………………………………………………………. 1
2. Contents……………………………………………………………………… 2
3. Introduction………………………………………………………………….3
4. Previous Work……………………………………………………………….. 5
5. Design………………………………………………………………………… 6
    5.1 Mass-Spring System Design…………………………………………….. 6
        5.1.1 Particles and Particle Systems…………………………………….6
        5.1.2 Forces……………………………………………………………... 7
        5.1.3 Springs…………………………………………………………….. 8
        5.1.4 Numerical Integration……………………………………………..9
    5.2 Meshless Deformations Design………………………………………… 10
        5.2.1 Shape Matching……………………………………………………...10
        5.2.2 Deformation Modes………………………………………………… 11
        5.2.3 Numerical Integration……………………………………………….. 12
6. Implementation……………………………………………………………........ 12
    6.1 Mass-Spring Implementation……………………………………………..12
    6.2 Meshless Deformations Implementation………………………………….. 14
        6.2.1 Applying Meshless Deformations to the cloth model………………15
        6.2.2 Quadratic Deformation……………………………………………… 16
        6.2.3 Interacting with the cloth model…………………………………….. 17
    6.3 Hybrid Implementation………………………………………………….... 18
7. Results…………………………………………………………………………... 20
    7.1 Cloth parameters……………………………………………………………. 22
        7.1.1 Mass-Spring System Parameters…………………………………… 22
        7.1.2 Meshless Deformation Parameters………………………………….. 25
        7.1.3 Hybrid Parameters………………………………………………….. 25
8. Conclusion……………………………………………………………………… 26
9. Future Work……………………………………………………………………...27
10. Acknowledgements……………………………………………………………. 28
11. References……………………………………………………………………….. 29

3. Introduction

Cloth simulation is an area that has received a lot of attention from the computer graphics community in the past. It is important because it adds to the realism of a graphically generated environment. For example, cloth simulation is required in order to achieve convincing life-like avatars in virtual environments (e.g. movie special effects). Pixar's animated feature film, *The Incredibles*, employed extensive use of cloth simulation to clothe the animated characters. It has also been used in other applications such as Computer Aided Design and Manufacturing (CAD/CAM), especially garment design software. For instance ClothReyes is a commercial fabric simulation plug-in for 3ds Max. Recently, simple cloth simulations have been used in various computer games as well, once again to add to the realism of the environment.

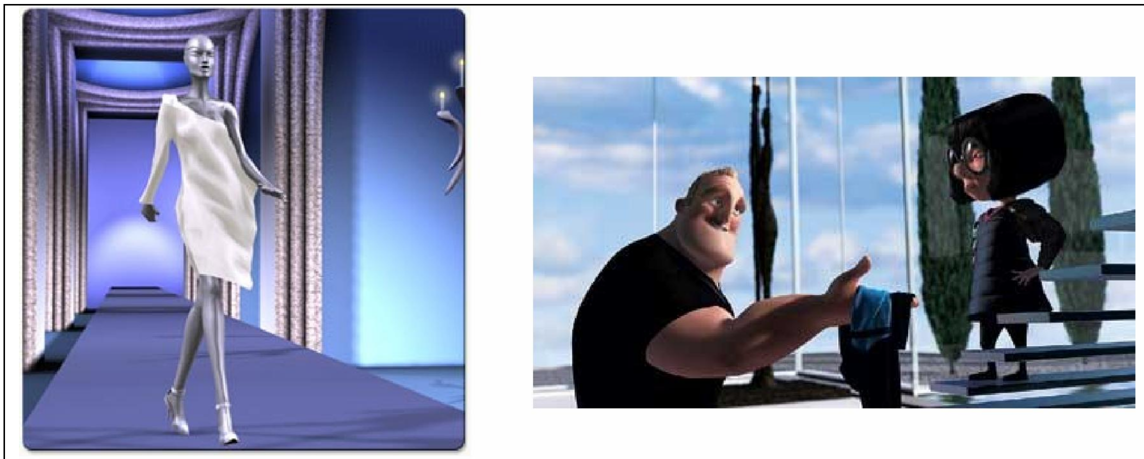


Figure 1: Depicted on the left is an image from the ClothReyes fabric simulator plug-in. On the right Bob hands Edna his superhero costume from Pixar's *The Incredibles*.

Traditional interactive cloth simulation methods generally suffer from several shortcomings (instability, computational expense, lack of control). In this project we wanted to investigate alternative methods and evaluate whether they offer any advantages over previously proposed techniques. We have implemented a cloth simulation demo using a standard approach and a novel approach and we have compared the results of the two techniques. Firstly, we implemented a cloth simulation system using a standard *Mass-Spring system*, an approach that was first described by [Baraff and Witkin, 1997]. Following this, we have used the recently proposed technique of *Meshless Deformations* [Muller et al. 2005] and applied it to the area of cloth simulation. *Meshless deformations* was chosen because it is a simple approach which is very fast and doesn't suffer from stability issues therefore it looks like a promising candidate for cloth simulation. Finally, in an attempt to harness the advantages of the separate techniques, we have also developed a hybrid system which combines various aspects from the above two approaches.

The rest of this report will proceed as follows. Section three will outline the major contributions from the past to the area of deformable object modelling and cloth simulation. Section four will describe the theoretical aspects behind *mass-spring systems* and the *meshless deformations* technique. The following section (section five) will detail our actual implementation of each system. The results are presented in section six, including a comparison between the three approaches. Lastly, the conclusion is presented along with possible future research considerations.

4. Previous Work

The simulation of deformable objects was first addressed by [Terzopoulos et al. 1987] who used elasticity theory to construct partial differential equations to model a deformable object's shape and motion. By numerically solving these partial differential equations realistic motion was produced. [Baraff and Witkin, 1997] detail a simple particle system approach together with the solving of differential equations using explicit numerical integration for use in physically based animation in their SIGGRAPH '95 course notes: 'An Introduction to Physically Based Modeling: Particle System Dynamics' and 'An Introduction to Physically Based Modeling: Differential Equation Basics'. Because of accuracy and stability problems associated with explicit numerical integration, Baraff and Witkin subsequently introduced a robust implicit integration scheme for use specifically in the area of cloth simulation [Baraff and Witkin, 1998].

Another approach that has proved successful at simulating deformable objects is the *Finite Element Method*. The *Finite Element Method* was derived from the theory of mechanics. Simplifying assumptions are made to make the mathematics and calculation easier. Rather than using particles with connectivity information the *Finite Element Method* is concerned with modelling elements. Since the differential equations are solved over the entire region of an element a much more realistic animation is produced. The *Finite Element Method* suffers from the drawback that it is computationally expensive, hence it is not usually an interactive approach. It is also much more difficult to implement than previous approaches, say for example *mass-spring systems*.

A novel approach to modelling deformable objects was proposed by [Muller et al. 2005] in their paper *'Meshless Deformations Based on Shape Matching'.* Although the technique is geometrically based rather than physically based, it gives results comparable to previous physically based approaches. The *meshless deformations* approach also overcomes various problems in previous research. For instance, although explicit numerical integration is simple and fast to implement it suffers from stability issues. Stability is dependent on the step size that has been chosen. The *meshless deformation* technique overcomes this weakness and guarantees unconditional stability. Moreover, the technique is computationally inexpensive and therefore, it allows the modeling of deformable objects to be performed at interactive rates.

5. Design

5.1 Mass-Spring System Design

5.1.1 Particles and Particle Systems

Mass-spring systems are defined by a set of particles (masses) and an explicitly defined connection between particles (springs). A particle is simply an entity, usually a point or a sphere, which holds information about its current position and velocity. Particles usually also have an associated mass. For physically based models the total, current force acting upon a particle is also kept track of. This is represented pictorially below:

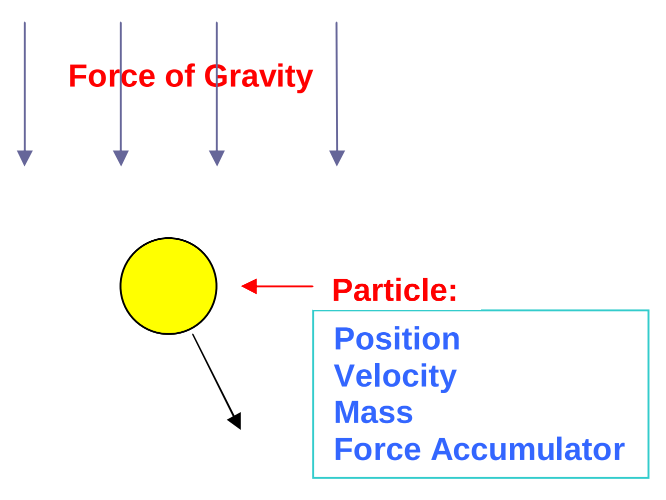


Figure 2: A particle being accelerated by external forces.

A collection of particles is known as a particle system.

5.1.2 Forces

To describe the motion of a particle system we need to include internal and external forces. In figure 2 the external force of gravity is being applied to the particle. This is a unary force as it is applied to each individual particle separately. To apply a unary force you simply need to add to the particle's 'force accumulator'. When we have a collection of particles there also exist *n*-ary forces acting between the particles. The most common *n*-ary force is a spring force between two particles:

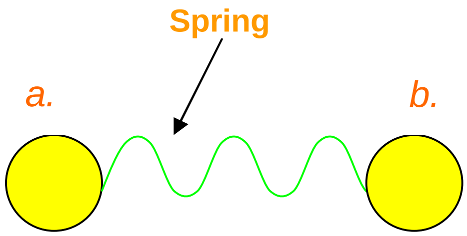


Figure 3: A spring force between a pair of particles.

The force that acts between the two particles located at positions *a* and *b* is calculated using Hooke's spring law as follows:

$$\mathbf{f}_a = -\left[k_s(|l| - r) + k_d \frac{\frac{dl}{dt} \cdot l}{|l|} \right] \frac{l}{|l|} \qquad (1)$$

Where, $l = a - b$, r is the rest length between the two particles, k_s is a spring constant, k_d is a damping constant, $\frac{dl}{dt}$ is the derivative of l which is given by the difference between the two velocities, i.e. $v_a - v_b$. $\mathbf{f}_a$ contains the total force applied to particle *a* due to particle *b*. Now, from Newton’s third law we know that particle *a* must apply an equal, but opposite force to particle *b*, therefore we also have:

$$\mathbf{f}_b = -\mathbf{f}_a. \tag{2}$$

5.1.3 Springs

When we have a particle system that uses spring forces to describe the forces between particles we have what is called a *mass-spring system*. For cloth simulation we use three separate types of springs: structural, shear and bend springs. These are illustrated below:

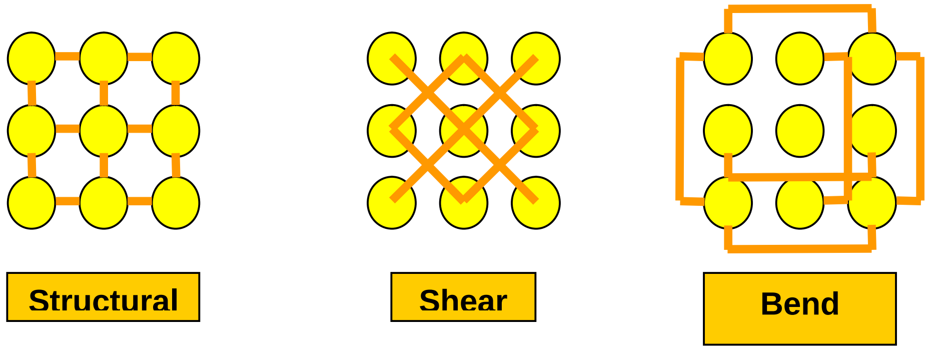
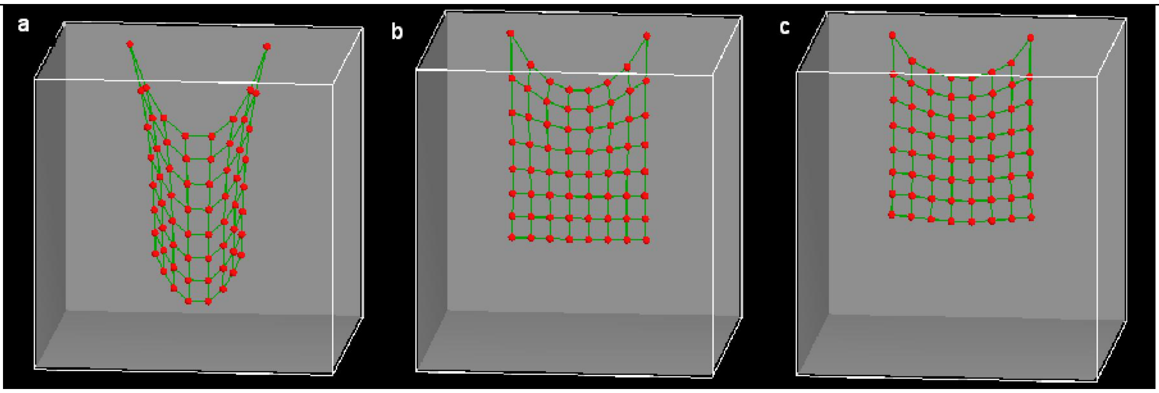


Figure 4: Various spring configurations used in cloth simulation.

Structural springs apply forces between directly opposite particles. Shear springs apply forces between diagonally oriented particles and bend springs apply forces between every alternate particle. As their names suggest the structural springs define the basic cloth structure. Shear springs ensure that the distance between diagonal particles will not become too close together or too far away, therefore they reduce the amount of shearing which will occur. Finally, by adding springs to every alternate particle, bend springs control the amount of bending which the cloth is able to undergo.



Structural | Structural + Shear | Structural + Shear + Bend

Figure 5: The effect that different types of springs have on the outcome of the simulation. It can be seen that structural springs alone do not allow the cloth is retain its shape.

5.1.4 Numerical Integration

Once all the external forces and the spring forces have been added the next step is to integrate the system so that the updated positions and velocities of each particle can be calculated. The simplest integration scheme available is the explicit Euler scheme:

$$x(t+h) = x(t) + hv(t)$$

$$v(t+h) = v(t) + h\frac{F}{m}. \qquad (3)$$

Where, *x* is a particle's position, *v* is that particle's velocity, *t* is the current time, *h* is the step size, *F* is the total accumulated force and *m* is the particle's mass. So, as can be seen by equation (3) a particle's updated position is calculated based on its current position and current velocity. And, the particle's updated velocity is calculated based on its current velocity and its current acceleration. So, basically the scheme can be summed up as follows:

> *'Given a current known position of some particle, explicit integration methods will try to guess where the next position of the particle will be.'*

This causes some problems when the guess overshoots the mark. So, the step size, *h*, usually has to be very small to avoid introducing accuracy errors into the simulation.

Another problem also occurs due to the springs which are used to enforce the constraints in the cloth model. The spring constants need to be quite large to satisfy the constraints of the model, this gives rise to stiff differential equations. A stiff differential equation is one where a small change in force (the derivative of the solution vector) can lead to a large change in position (the solution vector). Using explicit integration methods to solve stiff differential equations gives results that do not represent the actual solutions to the equations. To avoid dealing with stiff equations usually implicit integration is needed.

5.2 Meshless Deformations Design

5.2.1 Shape Matching

The *mass-spring system* described above takes into account the total accumulated force acting on each particle and uses this and the particle's mass to calculate the current position and velocity of the deformed object. The *meshless deformations* approach differs from this in that it uses *shape matching* to describe how to reshape a deformed object back to its original shape. External forces like gravity are still present, but internal forces (i.e. the spring forces) are no longer needed. The basic idea behind the shape matching algorithm is as follows. A set of particles known as the initial configuration, x_i^0, describe the initial undeformed state of the object. After a deformation has occurred a new set of particle positions is given describing the current, deformed shape of the object, x_i . Given the current, deformed object state and the initial, undeformed object state a set of goal positions, g_i , can be derived by constructing a transformation matrix. This is illustrated pictorially below:

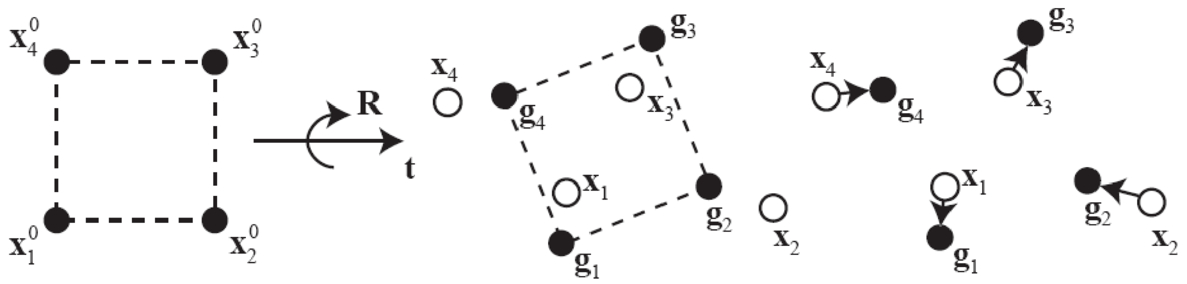


Figure 6: The shape matching algorithm. Image sourced from [Muller et el. 2005].

5.2.2 Deformation Modes

There are three deformation modes that *meshless deformation* allows: Rigid, Linear and Quadratic. When the transformation matrix only contains rotation and translation operations then we have a rigid body simulator. When the transformation matrix also allows shearing to occur we have linear deformations and finally, when the transformation matrix also allows bending and twisting we have what is called quadratic deformations. Because a piece of cloth allows deformations to occur such as shearing, bending and twisting the quadratic mode of the *meshless deformations* technique is required so that a large range of motion is able to be handled by the simulation. For the mathematical details involved in constructing the appropriate transformation matrices we refer the reader to [Muller et al. 2005].

5.2.3 Numerical Integration

The updated integration scheme now looks as follows:

$$\underline{v}_i(t+h) = \underline{v}_i(t) + \overset{\text{Shape Matching}}{\overbrace{\frac{\alpha\left(\underline{g}_i(t) - \underline{x}_i(t)\right)}{h}}} + h * \overset{\text{External forces}}{\overbrace{\frac{\underline{f}_{ext}(t)}{m_i}}}$$

$$\underline{x}_i(t+h) = \underline{x}_i(t) + h * \underline{v}_i(t+h) \qquad (4)$$

The above equations describe a slightly modified explicit Euler scheme as was given in equations 3. Once again, v_i , refers to the particle's velocity and x_i refers to the particle's position. In the shape matching component α refers to a stiffness parameter between the values 0 and 1 and g_i refers to the *i'th* particle's goal position.

Now, with this extra shape matching component added to the integration scheme we ensure that particles do not overshoot their goal positions and therefore we avoid the stability problems that explicit integration schemes suffer from. In fact, the *meshless deformations* technique guarantees unconditional stability provided that any forces involved in the simulation are applied to all particles independent of their location (for example the force of gravity) or the force is applied instantaneously (such as a collision response force). For a proof of stability see section 3.4 in [Muller et al. 2005].

6. Implementation

6.1 Mass-Spring Implementation

Our cloth simulation demo was implemented in the C/C++ programming language using the OpenGL graphics library (http://www.opengl.org/). Two main cloth configurations were experimented with, vertical and horizontal:

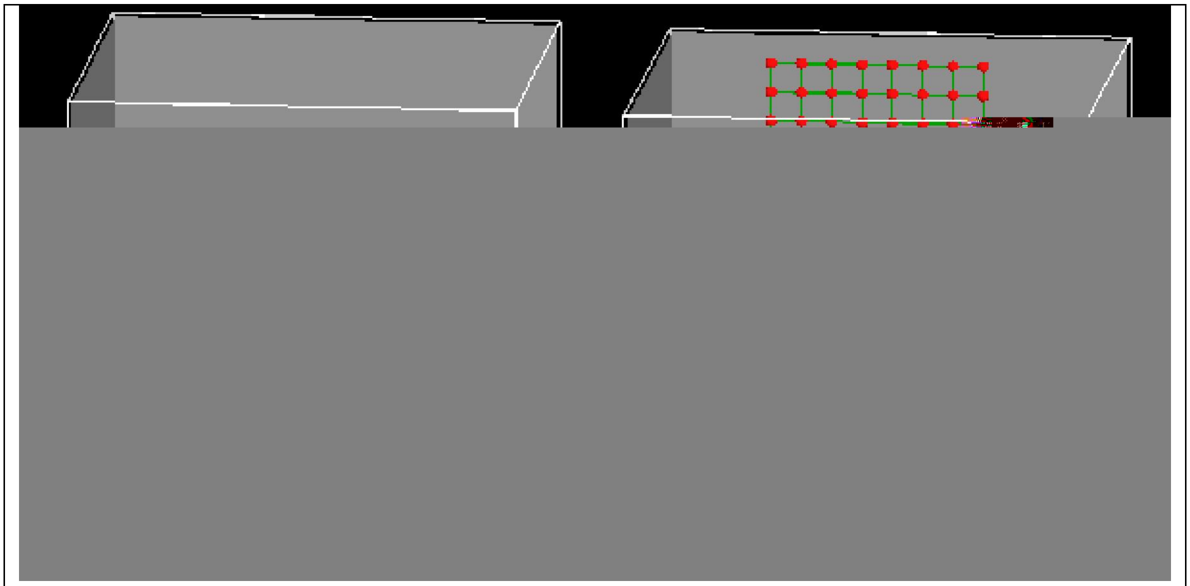
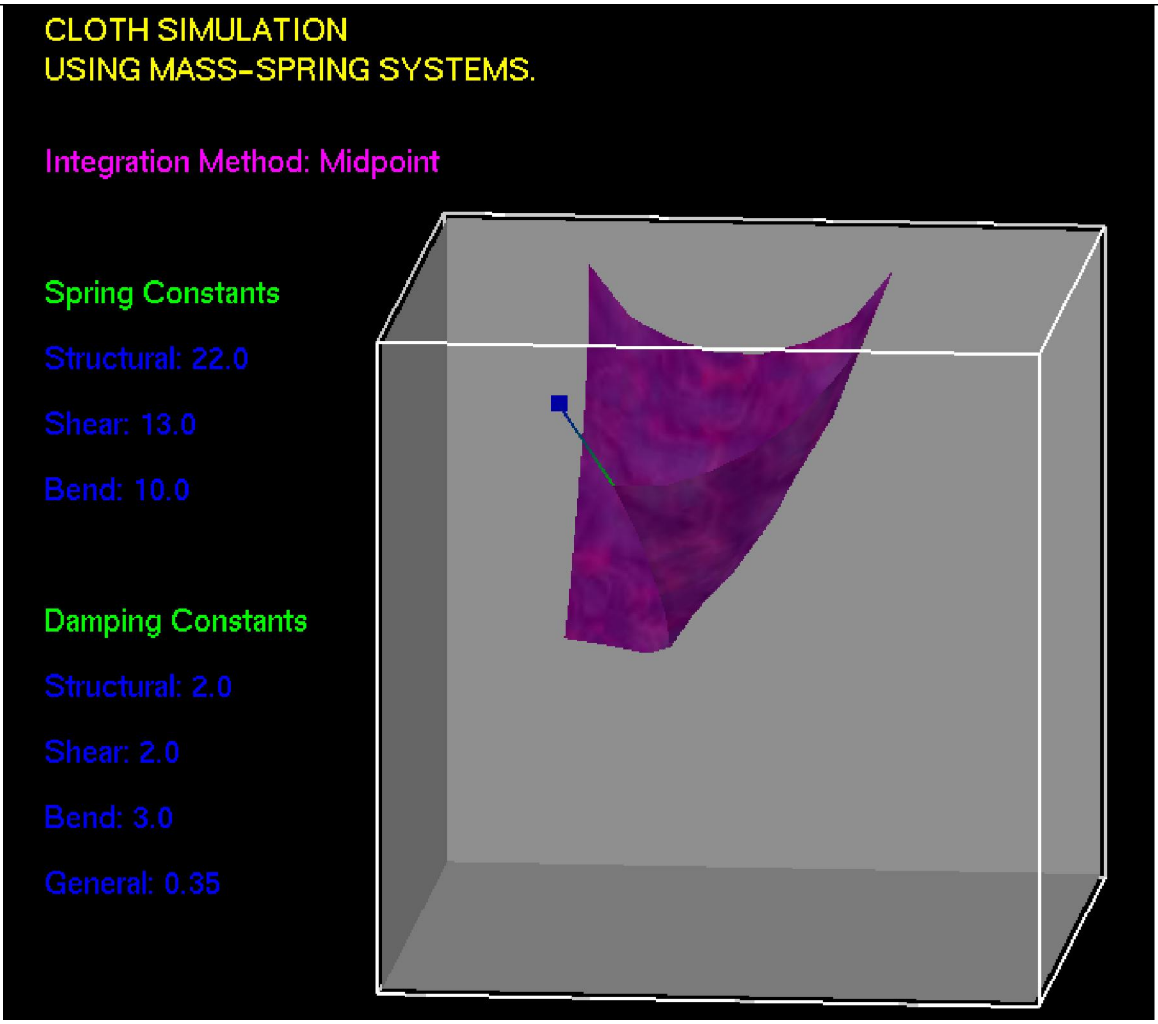


Figure 7: Initial horizontal and vertical cloth configurations.

The demo consisted of the graphical cloth simulation as well as a user interface to display and manipulate the various parameters involved with the animation, e.g. spring and damping constants. The user is able to select a particle and drag it around the scene and the cloth model will animate accordingly. A snapshot of the user interacting with the cloth model and the current parameters being used is given below. See section 7.1 for a discussion on the choice of parameters used and how different effects can be achieved by varying the parameters.



CLOTH SIMULATION
USING MASS-SPRING SYSTEMS.

Integration Method: Midpoint

Spring Constants

Structural: 22.0

Shear: 13.0

Bend: 10.0

Damping Constants

Structural: 2.0

Shear: 2.0

Bend: 3.0

General: 0.35

Figure 8: Snapshot of the cloth simulation demo. A user interacting with the cloth model while the current simulation parameters are being displayed.

The Explicit Euler and the Midpoint Method integration schemes, as described in [Baraff and Witkin, 1995], have been used. Collision detection was also implemented between the cloth and various objects, e.g. spheres and planes.

6.2 Meshless Deformations Implementation

A separate simulation was developed for the *meshless deformations* demo. Once again C/C++ and OpenGL were used. *Meshless deformations* requires multidimensional matrices and vectors in its calculations, the Boost: Basic Linear Algebra Library was used for this purpose. Once again the user is able to select a particular particle and move it and the user is able to modify the parameters involved in the simulation through a user interface. At present only a vertical cloth configuration has been tested and no collision detection has currently been implemented.

6.2.1 Applying Meshless Deformations to the cloth model

Because the cloth begins as a planar or relatively planar object we are effectively working in two dimensions initially. Now, the m*eshless deformations* technique as described by [Muller et al. 2005] is primarily intended for three dimensional deformable objects, therefore we cannot apply the three dimensional equations to an effectively two dimensional object. We have overcome this difficulty by extruding the cloth in the direction of the missing dimension. This gives two sets of particles to work with and effectively makes the cloth three dimensional. This is illustrated pictorially below:

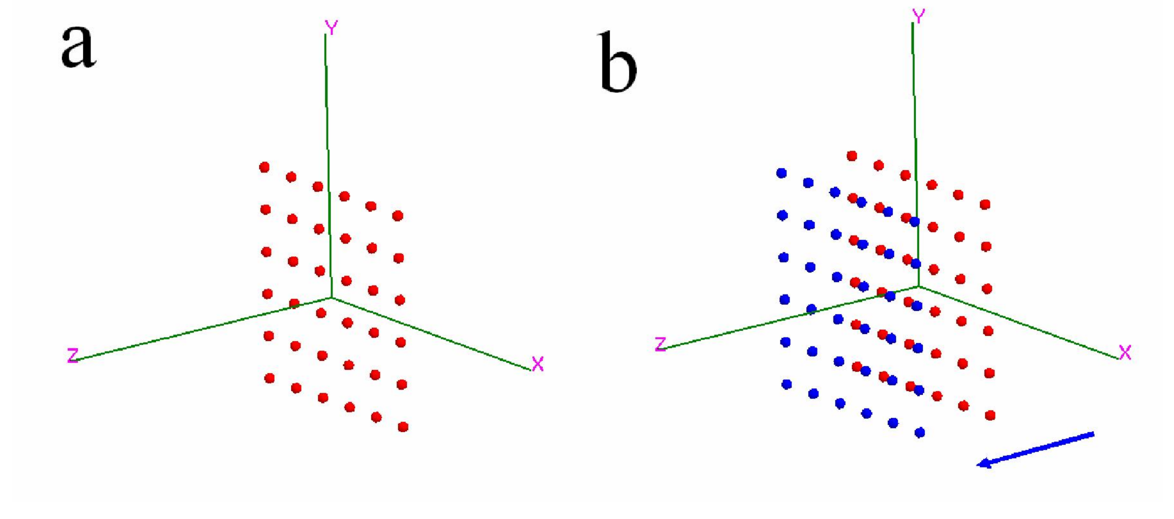


Figure 9: a) The initial particles that make up the cloth model are shown in red in the XY plane. b) The particles have now been extruded in the positive Z direction giving the new blue set of particles and a three dimensional structure.

The blue particles are then made as close as possible to the initial red particles. We have called the blue particles the *interface* particles (for reasons which will be detailed below).

6.2.2 Quadratic Deformation

This section will briefly overview the implementation of quadratic deformation. For more detailed information please consult [Muller et al. 2005]. To achieve quadratic deformation the *meshless deformations* method begins by considering each particle relative to the cloth's centre of mass, x_{cm}. Recall from section 5.2.1 that there exist an initial set of particles, x_i^0, and a current set of particles, x_i. Therefore we have:

$$\mathbf{q}_i = \underline{x}_i^0 - \underline{x}_{cm}^0,$$

$$\mathbf{p}_i = \underline{x}_i - \underline{x}_{cm}. \qquad (5)$$

Where q_i is an initial particle relative to the initial centre of mass and p_i is a current particle relative to the current centre of mass. For quadratic deformation we extend the three dimensional q_i vector into a nine dimensional vector $\tilde{q}_i$ like so:

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \xrightarrow{\qquad\qquad} \begin{pmatrix} q_x \\ q_y \\ q_z \\ q_x^2 \\ q_y^2 \\ q_z^2 \\ q_x q_y \\ q_y q_z \\ q_z q_x \end{pmatrix}$$

The initial *q* vector in R^3. The extended $\tilde{q}$ vector in R^9.

Now, the first three components in our new *q* vector are simply the same as our initial vector. The quadratic terms, (q_x^2, q_y^2, q_z^2), are introduced to allow for bending to occur and the mixed terms, (q_xq_y, q_yq_z, q_z,q_x), are introduced to allow for twisting. So, now we can compute the goal points with the following formula:

$$g_i = (\beta\tilde{A} + (1-\beta)\tilde{R})\tilde{q}_i + x_{cm} \tag{6}$$

Where, $\tilde{A}$ is the 3x9 optimal quadratic transformation matrix, $\tilde{R}$ is a 3x9 rotation matrix and β is a scalar value between 0 and 1 which controls how much of the quadratic transformation matrix will be applied to the goal positions.

6.2.3 Interacting with the cloth model

When the user selects a particle for manipulation a force is added to the particle's 'force accumulator' which is proportional to the distance between the mouse position and the position of the particle:

$$p_f = p_f + (m_x - p_x)/k \tag{7}$$

Where, p_f is the particle's total force accumulator, m_x is the current position of the mouse, p_x is the current position of the particle and k is a proportional constant. Now, because this force is not applied independent of the particle's location we have violated the conditions for an unconditionally stable system. This effectively means that the particle which is being manipulated experiences some instability in reforming to its correct location, causing it to oscillate around its goal position. This causes undesirable artifacts while the cloth is reforming. So, rather than directly accessing the particles involved in the cloth model, the user controls and manipulates the *interface* particles. These interface particles act as a window into the actual cloth particles, hence the name interface. The *interface* particles are hidden to the user and the result is the smooth and continuous motion of the cloth model in response to user applied forces. Figure 9 shows the cloth model with and without the *interface* particles visible.

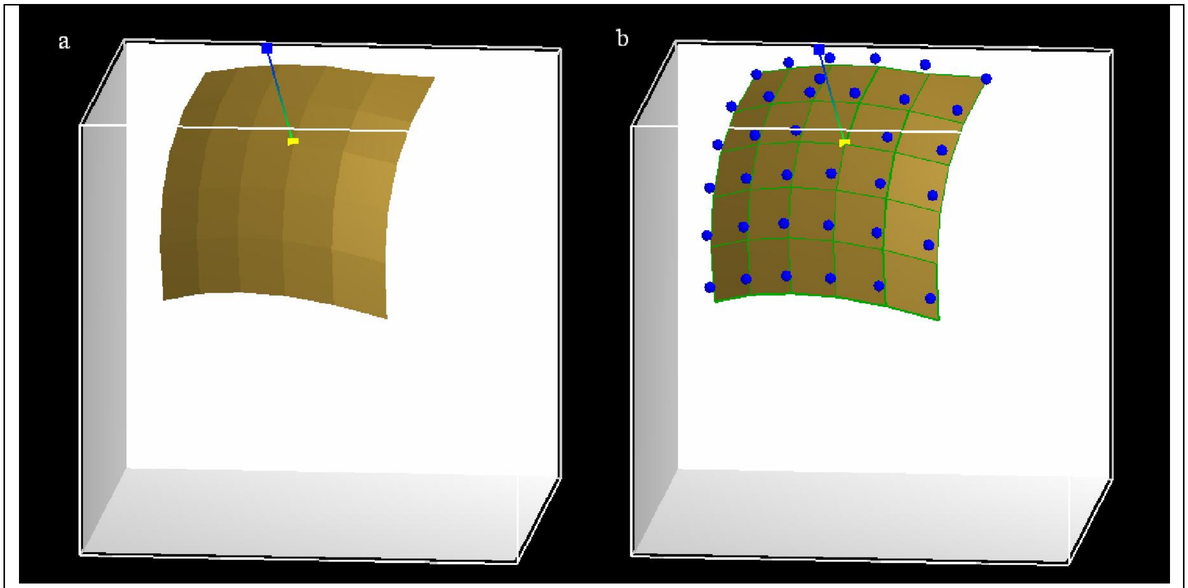


Figure 10: a) The cloth model being manipulated by the user. b) The same cloth model, but with the *interface* particles visible.

6.3 Hybrid (Mass-Spring + Meshless Deformation) Implementation

Finally, we have also implemented a hybrid approach which combines various aspects of *mass-spring systems* with *meshless deformations*. The *meshless deformation* method alone relies on the shape matching procedure to define the motion of the cloth model. Through our implementation we found that this approach doesn’t yield very realistic cloth motion. Our hybrid method adds to the *meshless deformation* approach by including internal cloth forces. Structural springs were used to represent the internal forces, i.e. forces between each neighboring particle. We found that the addition of internal forces added to the realism of the animation, although it did reduce the speed of the simulation (see section 7). Figure 11 shows a comparison between the *meshless deformations* approach alone and the hybrid method.

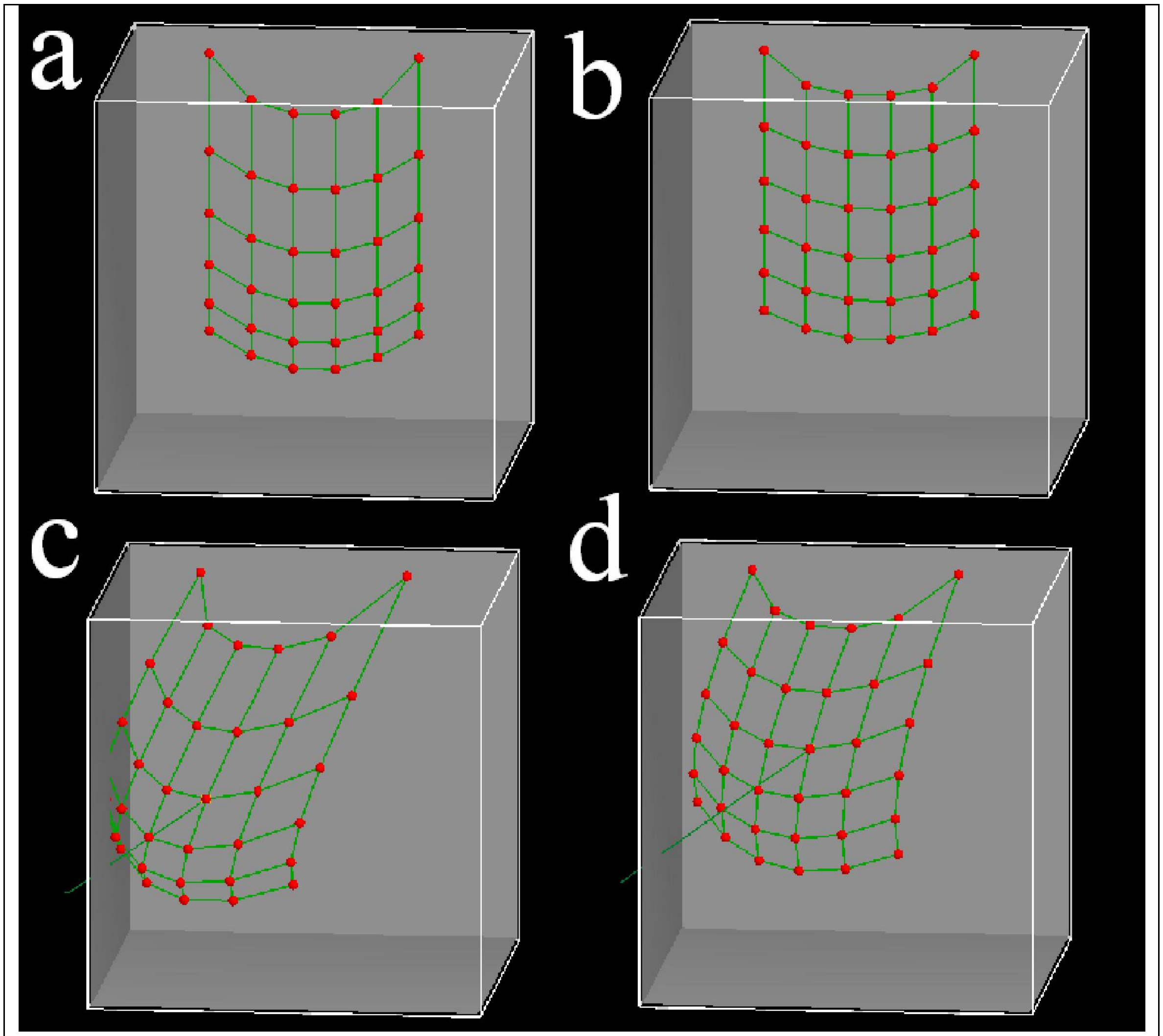


Figure 11: a) Shows the cloth in its rest state using the *meshless deformations* method, b) Shows the cloth in its rest state using the hybrid method. c) Depicts the user interacting with the *meshless deformations* demo and d) depicts the user interacting with the hybrid approach.

Looking at figure 11 (a) we can see that with *meshless deformations* there is no control over the distance between the neighbouring particles. Some particles are further away from each other and some particles are closer together. This can lead to unsatisfactory results when the user is manipulating the cloth model as in (c). By adding structural springs the distance between particles is controlled and this improves the quality of the animation.

7. Results

Overall for the *Mass-Spring System* using a relatively simple integration scheme we achieve a realistic cloth motion in response to various forces, e.g. collision forces, user forces. With careful tuning of the cloth parameters it is possible to simulate a variety of cloth types (section 7.1.1 below describes the actual parameters we have used). Because a small time step, h, has been used in the integration scheme (see equations 3) our cloth model is fairly accurate and stable. One of the main drawbacks, however, is that due to the necessity of the small step size the simulation can run quite slowly. [Muller et al. 2005] propose that the *Meshless Deformations* method is advantageous in this respect due to its computational efficiency. Below is a comparison between the computational costs of the three approaches. These tests were carried out on an Intel Pentium 4 CPU 3.00 GHz. The times shown are in milliseconds (ms).

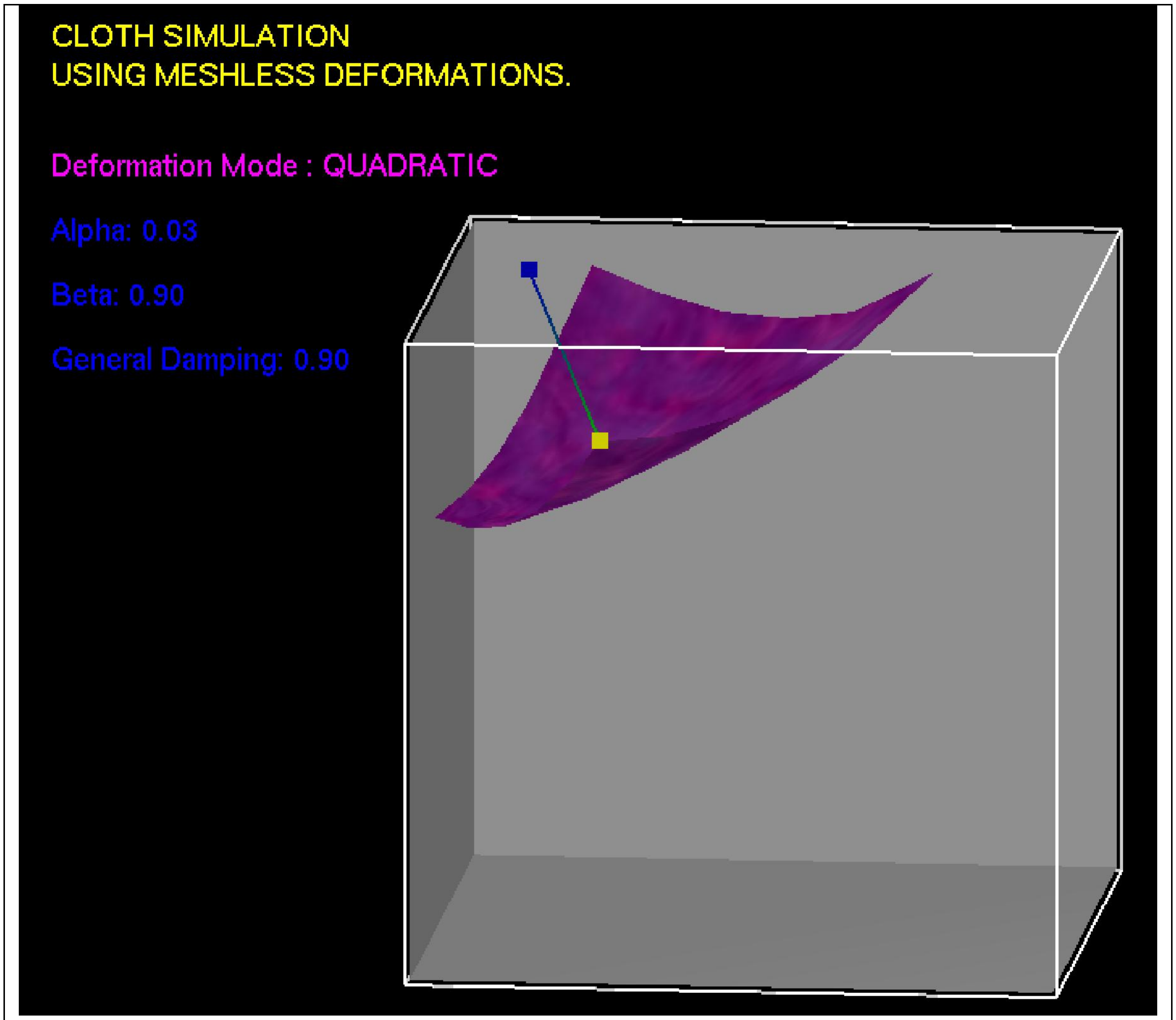
	Mass-Spring System	*Meshless Deformations*	*Hybrid Method*
64 Particles	1.73589	0.421016	1.51736
256 Particles	7.56833	1.71157	6.13846
1024 Particles	31.4719	6.44642	26.3644

Table 1: A comparison between the computational costs of the three approaches.

From table 1 we can conclude that the *Meshless Deformations* technique does achieve a greater computational efficiency than *Mass-Spring Systems*. The hybrid approach is only slightly less costly than a *mass-spring system.*

The *Meshless Deformations* technique will only yield relatively realistic cloth behaviour in response to a small user applied force (at present collisions with external objects have not been implemented in our *Meshless Deformations* demo). If a large deformation is applied then the resulting animation is no longer physically plausible. A large deformation will cause somewhat erratic motion from the cloth model. In general a very high ‘general damping’ constant had to be used to restrict the motion.

Comparisons of the animation produced by the three approaches yields some minor differences in the motion of the cloth model. In particular the *Meshless Deformations* approach generates a more rigid cloth model than the *Mass-Spring System*, it allows a lot less bending of the material to occur. Below is an illustrative example. Compare this with figure 7. Approximately the same deformation is applied by the user, but as can be seen a lot less bending is allowed to occur.



Deformation Mode : QUADRATIC

Alpha: 0.03

Beta: 0.90

General Damping: 0.90

Figure 12: The *Meshless Deformations* technique yields a much more rigid cloth simulation. Compare with figure 7 above.

Finally, by combining elements from *Mass-Spring Systems* to our *Meshless Deformations* implementation we found our hybrid method gives a slight improvement in the animation produced. We found that the best compromise between speed and realism was the addition of structural springs to the *Meshless Deformations* technique.

7.1 Cloth Parameters

Mass-spring systems are notorious for the difficulty involved in tuning the relevant parameters to obtain realistic behavior. In this section we will detail the parameters that worked for our implementation and some of the difficulties involved.

7.1.1 Mass-Spring System Parameters

There are a relatively large amount of parameters involved in a mass-spring system. For example, the structural, shear and bend springs all involve corresponding spring constants and damping constants. Moreover, the number of particles involved in the cloth model, the masses of each particle and the distances between the particles all also affect the final results of the animation. We found that modifying one of these factors resulted in the necessity to alter most of the other parameters to overcome undesirable results. As an example, increasing the number of particles in our cloth model led to the need to increase the spring and damping constants as well.

Specifically, we achieved the most realistic results for the mass-spring system implementation when our cloth model was made up of an 8 x 8 (64 particles in total) particle configuration where each particle had a mass of unity. The spring coefficients used were relatively high:

Structural spring coefficient:	*22.0*
Shear spring coefficient:	*13.0*
Bend spring coefficient:	*10.0*

And, the damping coefficients were as follows:

Structural damping coefficient:	*2.0*
Shear damping coefficient:	*2.0*
Bend damping coefficient:	*3.0*
General damping:	*0.35*

The use of these parameters gives the realistic impression of a cloth model with high elasticity for example a cotton T-shirt. However, we found that with the above values if you were to increase the number of particles in the cloth model the quality of the final animation would suffer and the above values would need to be modified to compensate for this. Below is an example of the cloth model colliding with a sphere and the different results which were generated.

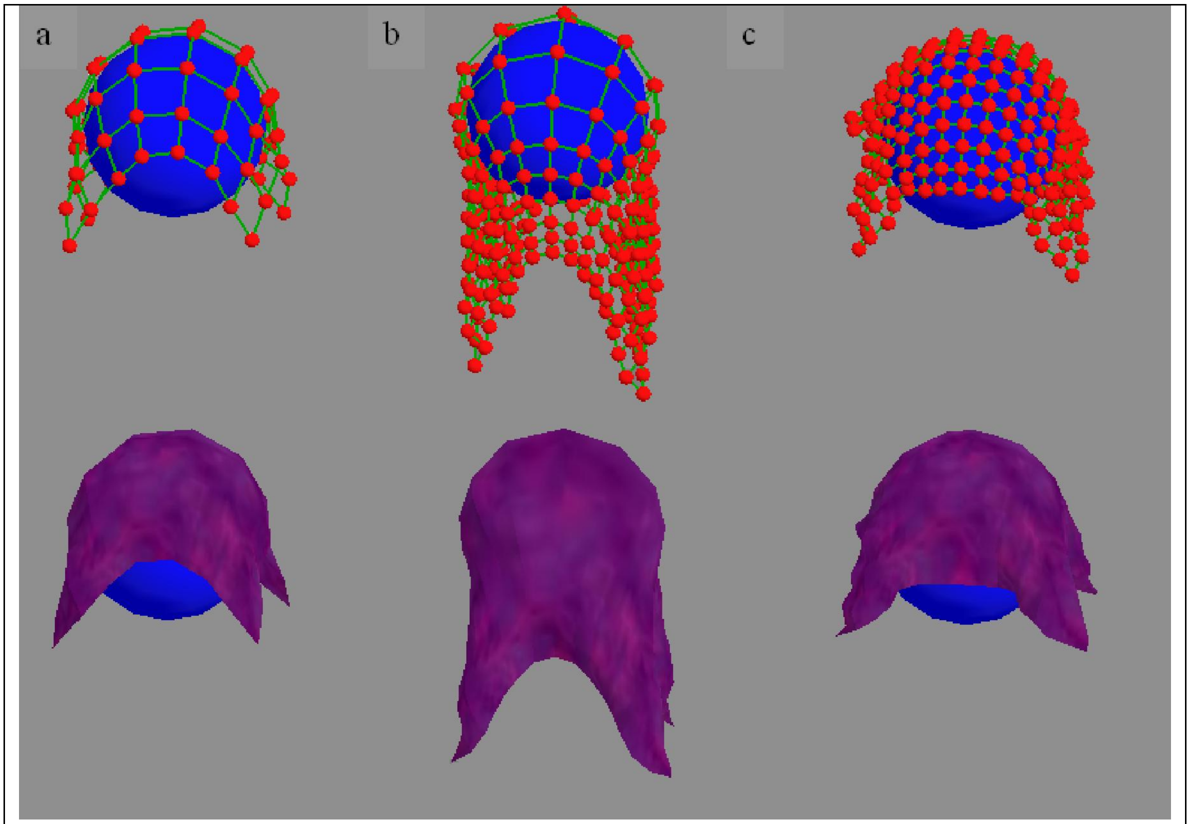


Figure 13: a) Shows the standard 8x8 cloth model described above. b) Shows what happens when the amount of particles is increased to 16x16 and c) Shows the 16x16 cloth model, but with the spring and damping coefficients modified to compensate for the increased number of particles.

It can be seen in figure13 that by increasing the number of particles without modifying the other parameters an unrealistic animation is produced. The cloth model now looks more like melted plastic than a cloth. After increasing the spring and damping constants the cloth model now behaves more realistically.

Below is a pictorial summary of the approximate material properties that are simulated when the structural, shear and bend springs are manipulated.

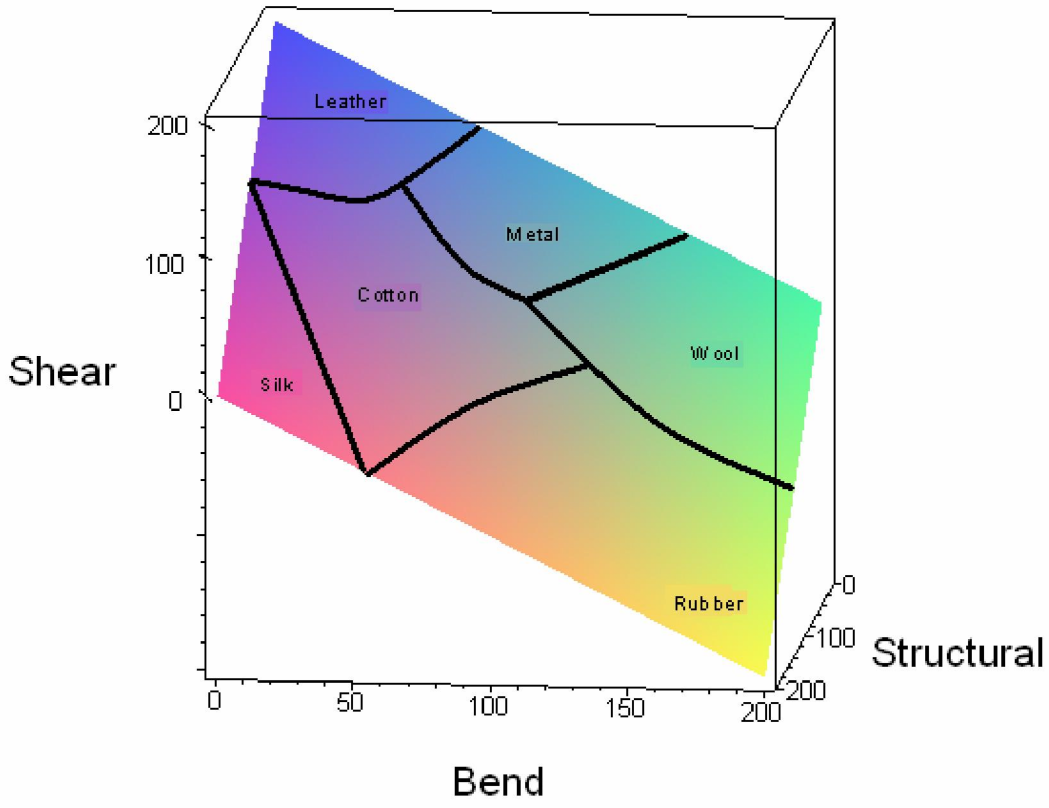


Figure 14: The approximate effects of modifying the structural, shear and bend coefficients.

Using a high shearing coefficient and a low to moderate bending coefficient gives results similar to the material of leather. Whereas, using low shearing and bending coefficients and a moderate structural coefficient gives a more silk like appearance. Interestingly, when shear is very high, structural is moderate and bend is low to moderate the simulation produces an animation more like metal than that of a cloth.

7.1.2 Meshless Deformation Parameters

There are a lot less parameters involved in the *meshless deformations* approach. Spring and damping constants are replaced with α and β . Recall from section *5.2.3* that α is a rigidity factor. The β parameter is a scalar value between 0 and 1 which controls how much of the quadratic transformation matrix will be applied to the goal points. While, $(1-\beta)$ controls how much of the rotation matrix will be applied to the goal points. The values that we have used in our implementation are as follows:

Alpha:	*0.03*
Beta:	*0.90*
General Damping:	*0.90*

We have used a 6x6 cloth model (36 particles in total). In general, modifying the number of particles that make up the cloth model has little affect on the results generated, however, making small changes to the α and β components can significantly change the animation produced. We have had to use an extremely low rigidity level to achieve cloth like effects.

7.1.3 Hybrid Parameters

Our hybrid approach uses the same *meshless deformation* parameters as in section 7.1.2. For the structural springs we use a spring constant of 22.0 and an increased damping constant of 20.0. The increased damping was needed to further control the erratic movement that the *meshless deformation* approach generates for large deformations.

8. Conclusion

In conclusion, we have implemented and investigated various aspects of cloth simulation. We began with a *Mass-Spring System* and implemented features such as collision detection/collision response with external objects and user interaction with the cloth model. Particular attention was paid to the structural, shear and bend parameters and the effect that varying these parameters had on the animations produced. We then applied a new technique (*Meshless Deformations*) for simulating deformable objects to the area of cloth simulation and compared the results obtained with our *Mass-Spring System*. We found that while it was possible to apply the m*eshless deformations* technique to the area of cloth simulation (with some minor adjustments) the results produced were only satisfactory for small deformations to the cloth model. Moderate to large deformations tended to produce unrealistic animations. We therefore continued by implementing a hybrid approach to try to take advantage of various aspects from both techniques. By combining the *Meshless Deformation* technique with structural springs a greater degree of realism is achieved and the computational costs of *mass-spring systems* are reduced.

9. Future Work

As mentioned above collision detection/collision response has not been implemented for the *Meshless Deformations* demo. This would prove a valuable future addition. Especially if a cloth model needed to be used in an application like a computer game. Collisions would need to be handled and they would need to be handled very quickly. Because of the computational benefits of the method it would be well worth adding this functionality.

In our implementation we ran into a few troubles with the *meshless deformation* calculations being applied to our semi two dimensional cloth model. We have used the original equations put forth by [Muller et al. 2005], however it may be worthwhile investigating a modified set of equations which would work with this two dimensional type of configuration.

10. Acknowledgements

The demo I produced in which the *meshless deformations* technique was applied to the area cloth simulation was based on a previous group project for COMPSCI 715 – Advanced Computer Graphics, in which the *meshless deformation* technique was implemented for regular three dimensional objects.

11. References

A. Witkin. An Introduction To Physically Based Modeling, chapter Particle System Dynamics. SIGGRAPH Course Notes, ACM SIGGRAPH, 1995.

A. Witkin, D. Baraff. An Introduction To Physically Based Modeling, chapter Differential Equation Basics. SIGGRAPH Course Notes, ACM SIGGRAPH, 1995.

Boost C++ Libraries: Basic Linear Algebra:
http://www.boost.org/libs/numeric/ublas/doc/

Collision Response: Bouncy, Trouncy, Fun:
www.darwin3d.com/gamedev/articles/col0399.pdf

ClothReyes: Cloth Simulation Featuring Tearing and Dynamics:
http://www.digimation.com/home/?Content=softwarehome.html

David Baraff , Andrew Witkin, Large steps in cloth simulation, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, p.43-54, July 1998

Demetri Terzopoulos , John Platt , Alan Barr , Kurt Fleischer, Elastically deformable models, ACM SIGGRAPH Computer Graphics, v.21 n.4, p.205-214, July 1987

Devil in a Blue Faceted Dress: Real-time cloth animation:
http://www.darwin3d.com/gamedev/articles/col0599.pdf

Hero Animation: Pixar borrows live-action techniques for its latest animated feature, *The Incredibles*:
http://cgw.pennnet.com/Articles/Article_Display.cfm?Section=Articles&Subsection=Display&ARTICLE_ID=215639

A. Kass. An Introduction To Physically Based Modeling, chapter Energy Functions and Stiffness. SIGGRAPH Course Notes, ACM SIGGRAPH, 1995.

M. Mueller, B. Heidelberger, M. Teschner, M. Gross:
Meshless Deformations Based on Shape Matching
Proceedings of SIGGRAPH'05, Los Angeles, USA, July 31 - August 4, 2005

Wünsche, B. C. (2004), A Toolkit for the Visualization of Tensor Fields in Biomedical Finite Element Models, PhD thesis, University of Auckland.
(Section 2.4: The Finite Element Method)