The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/0955-6222.htm

Study and comparison techniques in fabric simulation using mass spring model

Vajiha Mozafary and Pedram Payvandy
Department of Textile Engineering, Yazd University, Yazd, Iran

Received 18 March 2015
Revised 19 January 2016
Accepted 14 April 2016

Abstract

Purpose – The purpose of this paper is to conduct a survey on research in fabric and cloth simulation using mass spring model. Also in this paper some of the common methods in process of fabric simulation in mass spring model are discussed and compared.

Design/methodology/approach – This paper reviews and compares presented mesh types in mass spring model, forces applied on model, super elastic effect and ways to settle the super elasticity problem, numerical integration methods for solving equations, collision detection and its response. Some of common methods in fabric simulation are compared to each other. And by using examples of fabric simulation, advantages and limitations of each technique are mentioned.

Findings – Mass spring method is a fast and flexible technique with high ability to simulate fabric behavior in real time with different environmental conditions. Mass spring model has more accuracy than geometrical models and also it is faster than other physical modeling.

Originality/value – In the edge of digital, fabric simulation technology has been considered into many fields. 3D fabric simulation is complex and its implementation requires knowledge in different fields such as textile engineering, computer engineering and mechanical engineering. Several methods have been presented for fabric simulation such as physical and geometrical models. Mass spring model, the typical physically based method, is one of the methods for fabric simulation which widely considered by researchers.

Keywords 3D, Fabric, Simulation, Mass spring, Physical model

Paper type Literature review

1. Introduction

Since the 1980s, by the development of computer technology, fabric simulation has been considered in many fields, such as computer engineering, textile engineering, cloth design and also in industries like 3D games, animations and so on. The simulation of fabric is complex, due to the properties of fabric, for instance, large deformation, non-linear stress (buckle and wrinkle); the simulation of fabric is complex (Ng and Grimsdale, 1996). In the past three decades, many methods have been suggested by researchers for fabric modeling. Nowadays these methods can be roughly divided into three classes: geometrically based, physically based and hybrid methods. In the geometric models, complex deformations of fabric will be simulated with geometrical functions. In this method, the physical properties of the fabric like the fabric mass, pattern and the elastic coefficient are not considered. So it is not necessary to calculate a lot of complex equations about fabric physical state. The advantages of this method are simple and fast. However, this method has disadvantage which cannot simulate fabric behavior realistically (Weil, 1986; Hinds and McCartney, 1990). In physically based models, fabric is represented as a collection of numerous tiny elements. The forces or energies associated with each element are calculated to simulate the behavior of fabric. This method can simulate more realistically than geometric models. So physically based models are suitable for the simulation of fabric with large deformation (Chen and Govindaraj, 1995; Hu and Chen, 2000; Young *et al.*, 2001). Considering the advantages

International Journal of Clothing Science and Technology
Vol. 28 No. 5, 2016
pp. 634-689

0955-6222
DOI 10.1108/IJCST-03-2015-0033

and disadvantages of geometrical and physically based method, researchers combine these methods to produce a new hybrid method (Vassilev, 2001; Baraff and Witkin, 1998; Yang and Thalmann, 1993).

Physically based modeling methods include finite element (Ascough *et al.*, 1996; Gan *et al.*, 1995; Aono, 1996; Etzmuss *et al.*, 2003), finite volume (Hu and Chen, 2000), elasticity (Terzopoulos *et al.*, 1987; Feynman, 1986), particle system (Dai *et al.*, 2003; Zhong and Wang, 2001; Eberhardt *et al.*, 1996) and mass spring model. Among these models, the mass spring model is a simple and powerful approach for fabric simulation. In this model, fabric is represented as a mesh of mass points that masses connected together by elastic linkage (spring). In mass spring modeling, there is no need to solve equation simultaneously, which it is an advantage of mass spring model compared to the other physically models like finite element (Breen *et al.*, 1994).

This paper reviews conducted research about fabric simulation using mass spring model. In addition in this paper, some of common methods in fabric simulation are compared to each other. And by using examples of fabric simulation, advantages and limitations of each technique are mentioned. In all examples, Matlab R2014 software was employed to programming codes and for final execution.

This paper is organized as following. First section covers the introduction of the 3D simulation methods of fabrics. Second section investigates the presented mesh types in mass spring model by researchers. Second section presents the forces applied on model and measurement methods of forces. Third section describes the super elastic effect and ways to settle the super elasticity problem. Fourth section reviews numerical integration methods for solving equations. Fifth section covers collision detection and its response, and finally in sixth section, few of empirical samples presented by researchers about fabric simulation are mentioned.

2. Mesh

In mass spring model, fabric is represented as a grid of mass points; namely, called as mesh. In the mesh, connections between mass points are through elastic linkage (spring). Each mass point has position, velocity, acceleration and responds to both internal and external forces. By considering linkage between mass and springs, different types of mesh have been presented by researchers are listed in Table I.

2.1 Rectangular mesh

Rectangular mesh is the most common type of mesh used in the mass spring model, which was first proposed in 1995 by Provot. In this mesh, the fabric model is constructed by a rectangular grid of mass points (also called particles) with two principal directions representing the warp and weft directions, respectively. Each mass is linked to its neighbors by spring, as shown in Figure 1. Depending on the type of connection, there are three different types of springs defined as: structural springs, shear springs and bend springs (Provot, 1995):

- Structural springs that connect each particle with its closest particle in horizontal and vertical direction. These springs model stretching/compression behavior of fabric.
- Shear springs that connect each particle with its adjacent diagonal particle. These springs model shearing behavior of fabric.
- Bend springs that connect each mass with its neighbors in every other row in horizontal and vertical directions. These springs model bending behavior of fabric.

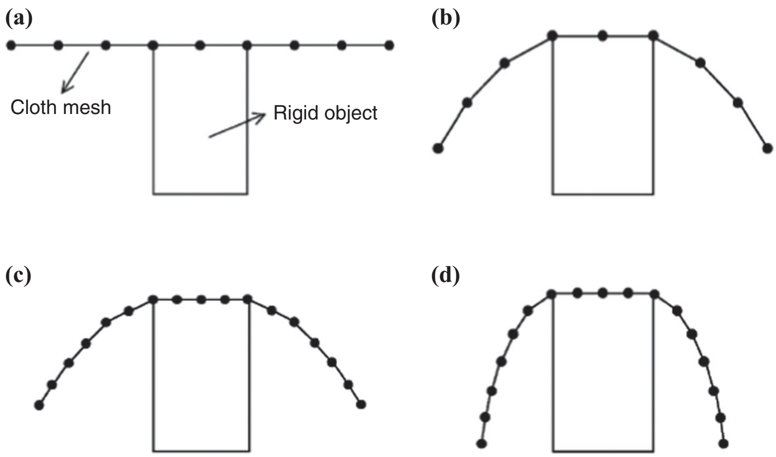
Table I.
Presented mesh types by researchers

S. No.	Mesh type	References
1.	Rectangular mesh	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Bayraktar (2002), Bridson *et al.* (2002), Connor and Stevens (2003), Dochev and Vassilev (2003), Ji *et al.* (2006a, b), Zhibin and Zhanli (2006), Charfi *et al.* (2006), Wenhsiao and Chen (2006), Durupinar and Gudukbay (2007), Horiba *et al.* (2007), Haiyan and Zhaofeng (2008), Wang *et al.* (2009), Zhong and Xu (2009), Ozgen *et al.* (2010), Qing *et al.* (2010), Mongus *et al.* (2012), Liu *et al.* (2012a, b), Shou *et al.* (2013), Zhang *et al.* (2012), Bender *et al.* (2009), Smith (2011), Ye *et al.* (2009), Goldenthal (2010), Yang and Shang (2013), Vassilev and Rousev (2008), Raymaekers *et al.* (2005), Tang *et al.* (2013), Chittaro and Corvaglia (2003)
2.	Multi-level mesh	Zhang and Yuen (2001)
3.	Responsive mesh	Guimaraes and Silva (1991), Zhou *et al.* (2005), Choi and Ko (2002), Asmaa and Elsayed (2011), Aileni *et al.* (2011), Zhou *et al.* (2008)
4.	Bilayred mesh	Kang and Cho (2002a, b)
5.	Adaptive mesh	Villard and Borouchaki (2005), Hutchinson *et al.* (1996), Simnett (2009)
6.	Triangular mesh	Selle *et al.* (2009), Fuhrmann *et al.* (2003), Oh *et al.* (2004, 2006), Wang and Devarajan (2008), Mesit *et al.* (2007), Cho *et al.* (2010), Li *et al.* (2013), Liu (2010)
7.	Simplified mesh	Huang *et al.* (2014) and Hu *et al.* (2013)
8.	Mesh with virtual springs	Yuan *et al.* (2010)
9.	Mesh with support springs	Zhengdong and Shuyuan (2011) and Vassilev and Spanlang (2014)
10.	Arc mesh	Rusinko and Swan (2012)

Figure 1.
Rectangular mesh for fabric simulation

2.2 Multi-level mesh

Zhang *et al.* presented multi-level mesh in 2001. In this mesh, process of fabric simulation divides into several phases. Initially, the fabric is modeled with a coarse mesh. In the following levels, finer mesh replace to a coarse mesh. As shown in Figure 2(a), the fabric is in its initial position, and it is represented with a coarse mesh. When it reaches the position as shown in Figure 2(b), finer mesh replace to a coarse mesh (Figure 2(c)), and continue simulation until the fabric reaches its final position (Figure 2(d)). Since the computational cost in simulation of a coarse mesh is much less than the fine mesh, using a coarse mesh at the beginning of simulation can make the process fast simulation (Zhang and Yuen, 2001).



Notes: (a) Initial position with large mesh size; (b) final position of phase 1; (c) replacing small mesh size with large mesh size; (d) final position of phase 2

Source: Zhang and Yuen (2001)

Figure 2. Multi-level meshes for fabric simulation

2.3 Responsive mesh

Responsive mesh is described by Choi and Ko (2002). At first this model looks very similar to Provot's model, but particles connections are different (Figure 3). In this mesh, two kinds of connections are defined as:

(1) Connections of all neighboring particles: it corresponds to structural and shear springs on Provot model. These springs model stretching, compression and shearing behavior of fabric.

(2) Connections of every other particle: it corresponds to the bend springs on Provot model, but it also includes particles in the diagonal direction. These springs model bending behavior of fabric (Choi and Ko, 2002).

2.4 Bilayered mesh

Kang and Cho (2002a) were presented uses bilayered mesh that as shown in Figure 4. The bilayered mesh structure is composed of two meshes of various numbers of mass points that are called sparse and dense mesh. Figure 4(a) shows the sparse mesh that represents the global properties of the fabric motion and Figure 4(b) shows the dense mesh for realistic appearance of the fabric model. The sparse mesh is used for

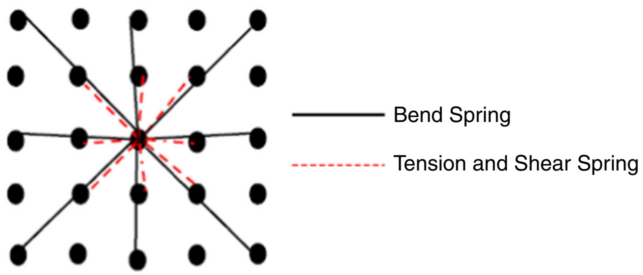


Figure 3. Responsive mesh for fabric simulation

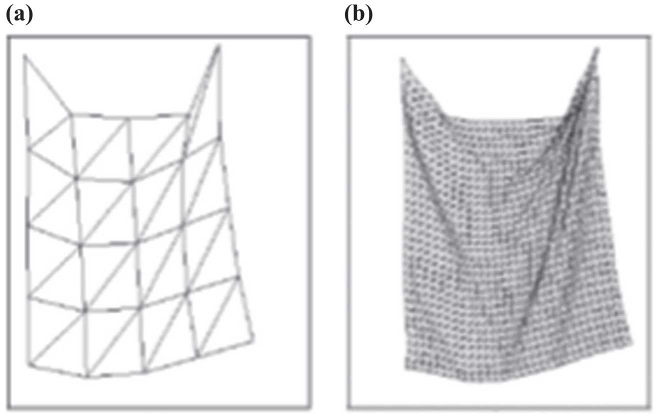


Figure 4.
Bilayered mesh

Notes: (a) Sparse mesh; (b) dense mesh
Source: Kang and Cho (2002a)

generating the global motion by considering both internal and external forces but the dense mesh just considers the internal forces. Therefore, the external forces do not effect on the behavior of dense mesh (Kang and Cho, 2002a, b).

2.5 Adaptive mesh

Villard *et al.* proposed a new method based on adaptive mesh in 2005. In the adaptive mesh by reducing the number of mesh elements, computational time will be reduced. Fabric surface is initially defined by a large mesh size. When the fabric falls, the corresponding mesh must be adapted to its geometrical shape. The base of adopting mesh is when the curvature of an area exceeds to a given threshold, finer mesh replace to a coarse mesh. An example of adaptation in three stages is shown in Figure 5 (Villard and Borouchaki, 2005).

2.6 Triangular mesh

Triangular mesh presented by Selle *et al.* (2009). As shown in Figure 6, the model consists of mass points which each mass point links to its neighbors. The linkage between neighbors is achieved in the two different following ways:

(1) Tension springs that link a mass to the nearest mass and would use all of the edges of the mesh. Tension springs model stretching and compression behavior of fabric.

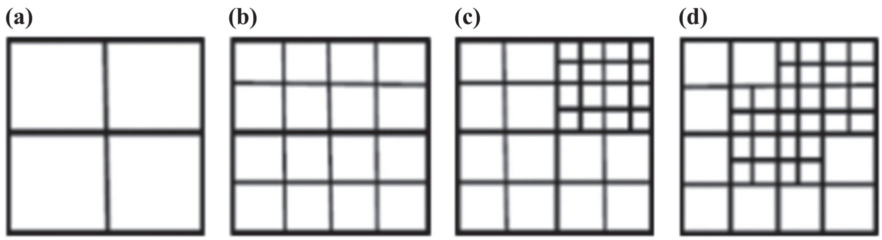


Figure 5.
Adaptive mesh for fabric simulation

Notes: (a) Large mesh size; (b) subdividing every elements to four smaller elements; (c) subdividing an element to four smaller elements; (d) subdividing other element to four smaller elements

(2) Bend springs that link two masses that are on two triangles which share the edge. Bend springs model bending behavior of fabric.

Since the triangular mesh itself can model shearing behavior of fabric, so for simulation the shearing behavior of fabric; linkage with spring is not needed. Comparing triangular to rectangular meshes makes clear that for each mass, there are less connected springs to triangular mesh, therefore; this mesh has the short computation as the rectangular mesh (Selle *et al.*, 2009).

2.7 Simplified mesh

Huang *et al.* (2014) simplified the Provot's model to improve the efficiency of computing. Results of Huang's study showed that it does not affect the model significantly whether it is with two shear springs or one shear spring, therefore; in this mesh, one of the shear springs is eliminated to simplify the model. The simplified mass spring model is shown in Figure 7 (Hu *et al.*, 2013).

2.8 Mesh with virtual springs

Yuan *et al.* (2010) introduced a new mesh based on virtual springs. This mesh is same as rectangular mesh but in addition to the structural, shear and bend springs, virtual springs are also used. Virtual springs simulate the return behavior of mass points to their original position by modeling the internal force below the fabric. Structure of mesh with virtual mesh is shown in Figure 8 (Yuan *et al.*, 2010).

2.9 Mesh with support springs

Zhengdong proposed a new mesh in 2011. There are two connections between springs: Tension spring and support spring. All surface vertices are connected to each other with tension springs. These springs model the elastic segment of the body. Each vertex is connected to the center with a support spring that models the content of the 3D object. Response of support spring depends on the state of the whole system and preserves the volume of an object (Zhengdong and Shuyuan, 2011).

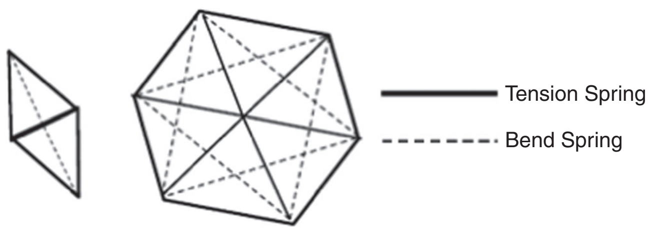


Figure 6. Triangular mesh for fabric simulation

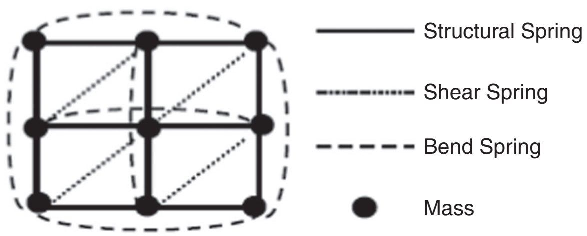


Figure 7. Simplified mass spring for fabric simulation

2.10 Arc mesh

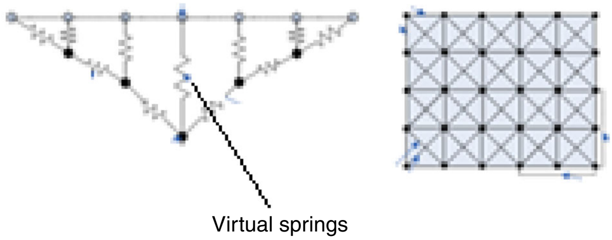
Rusinko *et al.* presented Arc mesh in 2012. As shown in Figure 9, Arc mesh is constructed of concentric circles. On each of these circles there are a number of equally spaced mass points. Rusinko compared fabric simulation using triangular, rectangular and Arc mesh on a real fabric to evaluate the drape coefficient (DC). Results showed that Arc mesh gives a more realistic DC by comparing experimental results than the other mesh. Also computational time for fabric modeling using Arc mesh is less than triangular and rectangular mesh (Rusinko and Swan, 2012).

2.11 Comparison of the above-mentioned four kinds of mesh

As mentioned above, various types of mesh are proposed by researchers. In this part, four kind of meshes that mostly used in fabric simulation are compared to each other. The considered meshes as follows: rectangular, triangular, simplified and responsive mesh. Also in each mesh, fabric simulation in eight mesh size (various numbers of mass points) is performed. Simulation results are compared as follows: computational time to reach balance situation, convergence of system and fabric appearance at balance situation.

Table II and Figure 10 show fabric drape on table for four meshes with different numbers of mass points. For rectangular mesh with 3,481 mass points, fabric drape from both top and front view are shown in Figure 10. And for other cases, only top views of samples are illustrated in Table II to summarization.

As are shown in Table II, through increasing mass points, drape fabric can be simulated better and the simulation is also much closer to the reality. Computation time to reach balance position for various meshes and mass points in fabric simulation is illustrated in Figure 11. The basic properties of the fabric model are the same. Only the type mesh and mass points are changed.



Source: Yuan *et al*. (2010)

Figure 8.
Structure of mesh with virtual mesh

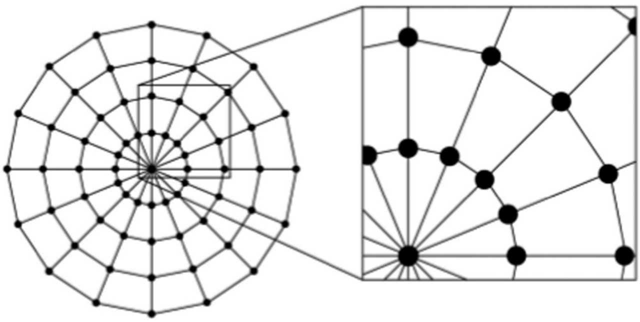


Figure 9.
Arc mesh for fabric simulation

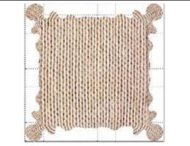
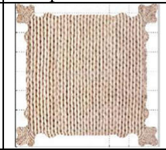
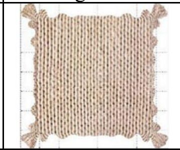
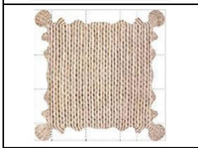
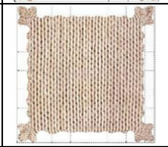
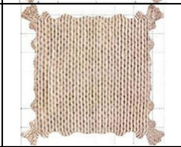
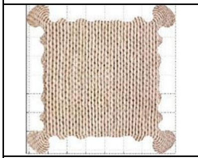
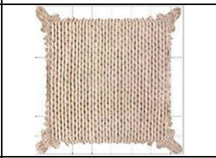
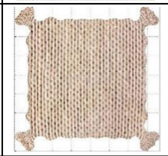
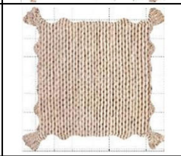
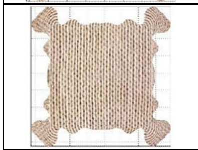
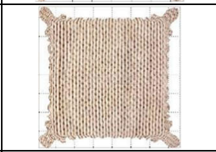
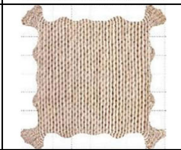
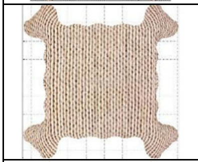
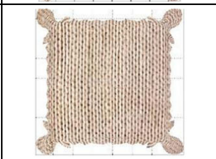
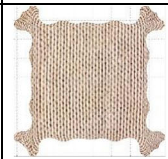
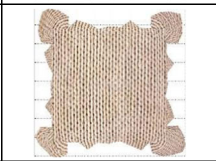
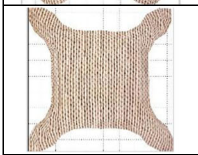
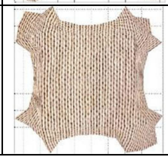
Responseive mesh	Triangular mesh	Simplified mesh	Rectangular mesh	Mass points
				3,481
				3,136
				2,500
				1,764
				1,089
				625
				289
				100

Table II.
Fabric simulation with various meshes and mass points (top view)

Results of Figure 11 show the computational time of triangle mesh is the minimum due to the less spring connection in its structure. Also responsive mesh has the greatest computational time because this mesh has the most spring connection. Therefore, it can be calculated that computational time has a direct relation with number of spring

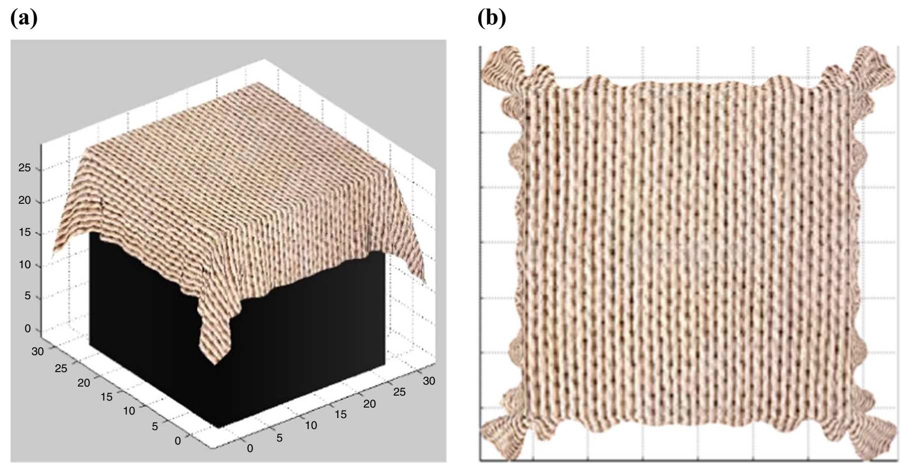


Figure 10.
Fabric simulation with Rectangular mesh and 3,481 mass points

Notes: (a) 3D view; (b) top view

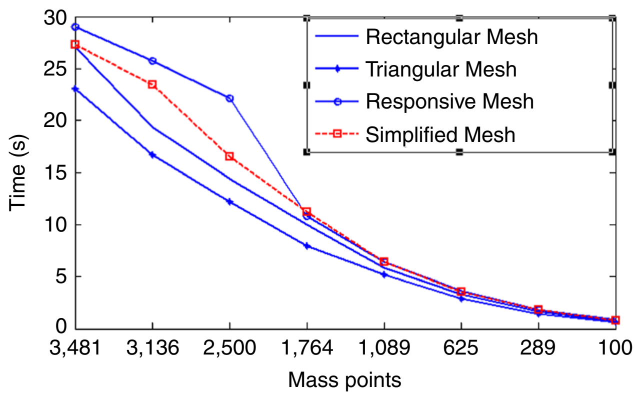


Figure 11.
Comparison of computation time to reach balance position for various meshes and mass points in fabric simulation

connection in mesh structure which is consistent with results of other researchers (Selle *et al.*, 2009; Choi and Ko, 2002; Hu *et al.*, 2013).

Also Figure 11 shows the difference of computational time between various mesh is reduced by decreasing number of mass points. This decreasing is made possible by the fact that the time of collision detection and collision response is reduced as the number of mass points is reduced.

Figure 12 shows comparison of fabric positions in balance situation for four meshes. The difference percent between the positions of the mass points in mentioned meshes are represented in Figure 12.

It is observed from Figure 12 that rectangular and simplified mesh have the greatest similarity. However, triangular and rectangular meshes have least similarity among mesh types.

Simulated fabric appearance in simplified and rectangular mesh is almost the same. However, computation time is less in simplified mesh than rectangular mesh. Therefore, simplified mesh can be used instead of rectangular mesh in fabric simulation.

3. Forces analysis

In mass spring model, the position of each particle depends on both internal forces and external forces applied on. And the position of all particles reflects the appearance of the fabric. Position of each particle is determined by Newton's Second Law, in accordance of the following equation:

$$F_i = m_i a_i \tag{1}$$

where m_i is the mass of particle *i*th, a_i the acceleration of particle *i*th and F_i the sum of both internal and external forces applied on particle *i*th. The internal forces are considered by researchers for fabric simulation based on mass spring model, shown in Table III.

3.1 Internal forces

Internal forces determine the mechanical properties of the fabric and mainly include stretch, shear and bend forces. Most researchers use Hook's law to calculate internal forces. The internal forces at each mass point are the whole results from the forces of all springs linking this point to its neighbors. According to Figure 13, the internal force in P_i can be represented in the following equation:

$$F(P_i) = -K(L - L_0) \tag{2}$$

where L is spring length, L_0 the natural length of spring, F the force applied at P_i and K the spring stiffness coefficient connected P_i and P_j (Provot, 1995).

Mongus *et al.* (2012) used genetic algorithm to find best value for spring stiffness coefficient through minimization error between model and experimental data.

Some researchers presented other method for internal forces calculation to more realistic simulation of fabric behavior that discussed as follows.

3.1.1 Using surface and volume difference. In the model proposed by Zhengdong which used the springs support; the force applied on the P_i due to the *i*th support spring is computed as in the following equation:

$$F(P_i) = -K(\Delta l_{tot}(t) + C\Delta l_i(t))u_i \tag{3}$$

where K is the spring stiffness coefficient, u_i the unit vector, C the body center and a coefficient in (0, 1), Δl_{tot} the volume differences per area unit and Δl_i the length changes of spring *i*th (Zhengdong and Shuyuan, 2011).

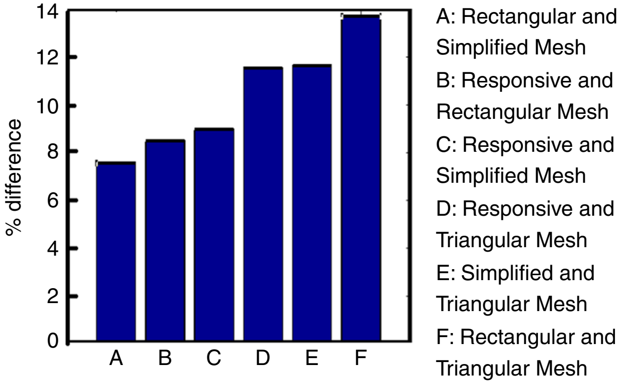


Figure 12. Difference percent between the positions of the mass points in four meshes with 2,500 mass points

Internal force	Calculation of force	References
Stretch	Using Hook's law	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Connor and Stevens (2003), Dochev and Vassilev (2003), Zhibin and Zhanli (2006), Haiyan and Zhaofeng (2008), Qing *et al.* (2010), Liu *et al.* (2012a, b), Fuhrmann *et al.* (2003), Mesit *et al.* (2007), Cho *et al.* (2010), Rusinko and Swan (2012), Huang *et al.* (2014), Zhang and Yuen (2001), Kang *et al.* (2000), Smith (2011), Goldenthal (2010), Yang and Shang (2013), Vassilev and Rousev (2008), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Simnett (2009), Yuan *et al.* (2010), Kang and Cho (2002a, b), Zhou *et al.* (2008), Chittaro and Corvaglia (2003)
	Using Kawabata test	Zhong and Wang (1998), Ji *et al.* (2006a, b), Charfi *et al.* (2006), Chen *et al.* (2012), Dai *et al.* (2001), Zhang *et al.* (2012), Han *et al.* (2009)
	Using plate theory	Wang and Devarajan (2008)
	Surface and volume difference	Zhengdong and Shuyuan (2011), Vassilev and Spanlang (2014)
	Using genetic algorithm for determining of spring stiffness coefficient	Louchet *et al.* (1995), Mongus *et al.* (2012)
Shear	Using Hook's law	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Connor and Stevens (2003), Dochev and Vassilev (2003), Zhibin and Zhanli (2006), Haiyan and Zhaofeng (2008), Qing *et al.* (2010), Liu *et al.* (2012a, b), Fuhrmann *et al.* (2003), Mesit *et al.* (2007), Cho *et al.* (2010), Rusinko and Swan (2012), Huang *et al.* (2014), Zhang and Yuen (2001), Kang *et al.* (2000), Smith (2011), Goldenthal (2010), Yang and Shang (2013), Vassilev and Rousev (2008), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Simnett (2009), Yuan *et al.* (2010), Kang and Cho (2002a, b), Zhou *et al.* (2008), Chittaro and Corvaglia (2003)
	Using Kawabata test	Zhong and Wang (1998), Ji *et al.* (2006a, b), Charfi *et al.* (2006), Chen *et al.* (2012), Dai *et al.* (2001), Zhang *et al.* (2012), Han *et al.* (2009)
	Using plate theory	Wang and Devarajan (2008)
	Surface and volume difference	Zhengdong and Shuyuan (2011), Vassilev and Spanlang (2014)
	Using genetic algorithm for determining of spring stiffness coefficient	Louchet *et al.* (1995), Mongus *et al.* (2012)
	Using twisting spring	Dai *et al.* (2003)
Bend	Using Hook's law	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Connor and Stevens (2003), Dochev and Vassilev (2003),

(*continued*)

Table III.
Internal forces are considered by researchers for fabric simulation based on mass spring model

Internal force	Calculation of force	References
		Zhibin and Zhanli (2006), Haiyan and Zhaofeng (2008), Qing *et al.* (2010), Liu *et al.* (2012a, b), Fuhrmann *et al.* (2003), Mesit *et al.* (2007), Cho *et al.* (2010), Rusinko and Swan (2012), Huang *et al.* (2014), Zhang and Yuen (2001), Kang *et al.* (2000), Smith (2011), Goldenthal (2010), Yang and Shang (2013), Vassilev and Rousev (2008), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Simnett (2009), Yuan *et al.* (2010), Kang and Cho (2002a, b), Chittaro and Corvaglia (2003)
	Using Kawabata test	Zhong and Wang (1998), Ji *et al.* (2006a, b), Charfi *et al.* (2006), Chen *et al.* (2012), Dai *et al.* (2001), Zhang *et al.* (2012), Han *et al.* (2009)
	Using plate theory	Wang and Devarajan (2008), Zhou *et al.* (2008)
	Surface and volume difference	Zhengdong and Shuyuan (2011), Vassilev and Spanlang (2014)
	Using genetic algorithm for determining of spring stiffness coefficient	Louchet *et al.* (1995), Mongus *et al.* (2012)
	Using non-linear spring	Asmaa and Elsayed (2011)
	Using beam's mechanical behavior	Villard and Borouchaki (2005)
	Using a model based on curvature and change rate of the angle	Liu *et al.* (2012a, b), Bridson *et al.* (2003)

Table III.

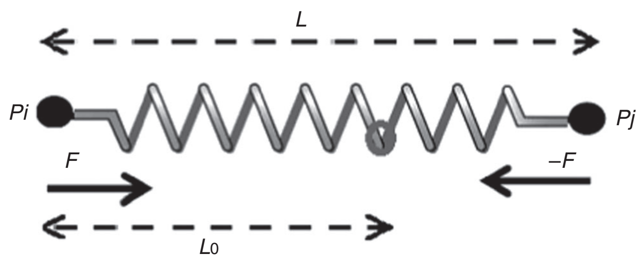


Figure 13.
Spring force between two mass points

3.1.2 Using Kawabata test. Deriving internal forces from the experimental data produced by the Kawabata technique has recently been considered. Zhong and Wang (1998) (2010), used polynomial approximation to Kawabata bending and shearing plots is adopted to construct the empirical constitution equations of bending and shearing springs (Zhong and Wang, 1998). Feng Ji *et al.* (2006a, b) calculated stretch force from Kawabata test by assuming when the tensile force is exerted, all the warp-structural springs are uniformly loaded. Also the measured shear behavior of the fabric using the tensile tests in the directions of 45° to the warp direction (Ji *et al.*, 2006b). Zhang *et al.* (2012) calculated bend force of fabric from Kawabata test. In the Kawabata bending test, the results are as the curve of bending moment and sample curvature. In the mass-spring model, the relationship between the force of the spring and the distance of its both ends are needed. So in order to calculate the mechanical properties of the bending spring, the testing results from Kawabata bending instrument transformed. The moment transformed into the force and the curvature transformed into the distance (Zhang *et al.*, 2012).

3.1.3 Using a model based on curvature and change rate of the angle. Bridson *et al.* (2003) presented a new method for bend behavior of fabric in mass spring model. In this method, bend force measured by curvature angle among two adjacent triangles which shares the same edge and change rate of angle (Bridson *et al.*, 2003).

3.1.4 Using beam's mechanical behavior. Villard and Borouchaki (2005) proposed a new approach for the computation of bending force, based on the beam mechanical behavior. As shown on the Figure 14, the force applied to the end of the beam defined as in the following equation (Villard and Borouchaki, 2005):

$$F = \frac{E \cdot I \cdot \alpha}{l^2} \tag{4}$$

where E is the young modulus, I the moment of inertia, l the beam length and α the angle of surface deviation. In the method for calculating of bend springs parameters, EI considered being equivalent to a spring stiffness coefficient and α/l^2 being equivalent to a displacement.

3.1.5 Using non-linear spring. Asmaa and Elsayed (2011) used Hook law for modeling the stretch and sheer behavior of fabric and used a non-linear spring for modeling the bend behavior of fabric. In this method Equation (5) is presented for behavior of non-linear spring (Asmaa and Elsayed, 2011):

$$F = k_b k_2 \left(\cos\left(\frac{kl}{2}\right) - \sin c\left(\frac{kl}{2}\right)\right)^{-1} \frac{x_{ij}}{|x_{ij}|} \tag{5}$$

where K_b is the flexural rigidity, K the curvature of the spring connecting the mass points i and j due to the exerted bending force. l is the natural length of spring and x_{ij} the distance between the two mass points i and j.

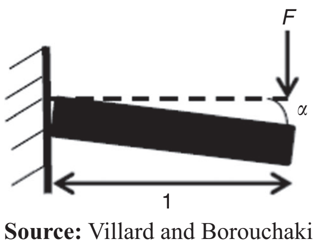
3.2 External forces

The external forces are various according to the environment condition which is mentioned for fabric simulation. The common external forces are used by researchers for fabric simulation based on mass spring, shown in Table IV.

3.2.1 Gravity. Gravity force is considered by all the researchers as an external force. Gravity force defined as in the following equation:

$$F_{gravity} = mg \tag{6}$$

where m is the particle mass, g the acceleration of gravity and $F_{gravity}$ the gravity force.



Source: Villard and Borouchaki (2005)

Figure 14.
Simulation of fabric bend behavior by beam mechanical behavior

S. No.	External force	References
1.	Gravity	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Bayraktar (2002), Dochev and Vassilev (2003), Ji *et al.* (2006a, b), Zhibin and Zhanli (2006), Charfi *et al.* (2006), Horiba *et al.* (2007), Haiyan and Zhaofeng (2008), Wang *et al.* (2009), Ozgen *et al.* (2010), Qing *et al.* (2010), Mongus *et al.* (2012), Shou *et al.* (2013), Selle *et al.* (2009), Oh *et al.* (2004), Wang and Devarajan (2008), Mesit *et al.* (2007), Cho *et al.* (2010), Li *et al.* (2013), Rusinko and Swan (2012), Zhengdong and Shuyuan (2011), Vassilev and Spanlang (2014), Villard and Borouchaki (2005), Huang *et al.* (2014), Zhang and Yuen (2001), Asmaa and Elsayed (2011), Chen *et al.* (2012), Dai *et al.* (2001), Zhang *et al.* (2012), Meyer *et al.* (2001), Zhenfang and Bing (2012), Kang *et al.* (2000), Bender *et al.* (2009), Oh *et al.* (2002), Smith (2011), Volino and Thalmann (2002), Ye *et al.* (2009), Goldenthal (2010), Yang, Shang (2013), Vassilev and Rousev (2008), Kang *et al.* (2014), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Zhou *et al.* (2008), Chittaro and Corvaglia (2003)
2.	Air resistance	Zhong and Wang (1998), Bayraktar *et al.* (2007), Ji *et al.* (2006a, b), Zhibin and Zhanli (2006), Horiba *et al.* (2007), Wang *et al.* (2009), Ozgen *et al.* (2010), Villard and Borouchaki (2005), Asmaa and Elsayed (2011), Zhang *et al.* (2012), Meyer *et al.* (2001), Kang *et al.* (2000, 2014), Oh *et al.* (2002), Goldenthal (2010), Yang and Shang (2013)
3.	Damping	Vassilev (2000, 2001), Provot (1995), Louchet *et al.* (1995), Vassilev and Spanlang (2000, 2014), Bayraktar (2002), Dochev and Vassilev (2003), Zhibin and Zhanli (2006), Charfi *et al.* (2006), Haiyan and Zhaofeng (2008), Qing *et al.* (2010), Shou *et al.* (2013), Mesit *et al.* (2007), Zhengdong and Shuyuan (2011), Huang *et al.* (2014), Zhang and Yuen (2001), Chen *et al.* (2012), Vassilev and Rousev (2008), Hu *et al.* (2013), Liu (2010), Chittaro and Corvaglia (2003)
4.	Wind	Vassilev (2001), Provot (1995), Louchet *et al.* (1995), Bayraktar *et al.* (2007), Bayraktar (2002), Ji *et al.* (2006a, b), Zhibin and Zhanli (2006), Horiba *et al.* (2007), Wang *et al.* (2009), Qing *et al.* (2010), Mongus *et al.* (2012), Shou *et al.* (2013), Zhengdong and Shuyuan (2011), Huang *et al.* (2014), Zhang and Yuen (2001), Meyer *et al.* (2001), Kang *et al.* (2000, 2014), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Chittaro and Corvaglia (2003)
5.	Sewing	Durupinar and Gudukbay (2007) and Zhong and Xu (2009)
6.	Water	Ozgen *et al.* (2010) and Chen *et al.* (2012)

Table IV. External force used by researchers for fabric simulation based on mass spring

3.2.2 Air resistance. Some researchers have considered air resistance in mass spring model to make simulation of the fabric dynamic behavior more realistic. When fabric surface is large, air resistance force cannot be neglected. Air resistance is applied to all mass points and represented kinetic energy dissipation of the model. Air resistance force is calculated as in the following equation:

$$F_{air} = -C_{air} V \tag{7}$$

where F_{air} is the air resistance force, C_{air} the air resistance coefficient and V the particle velocity.

3.2.3 Damping. Damping force is necessary to maintain the stability of the system. The role of this damping is in fact to model approximation the dissipation of the mechanical energy of model. It is introduced as an external force but could be considered as an internal force as well. Damping force can be represented as in the following equation:

$$F_{damping} = -C_{damping} V \tag{8}$$

where $F_{damping}$ is the damping force, $C_{damping}$ the damping coefficient and V the particle velocity.

3.2.4 Wind. To simulated the fabric exposed to the wind fluid; wind force must be taken as external force to make the performance more realistic. Wind fluid moving at a uniform velocity exerts force on the surface of a fabric. If wind changes are ignored when encounters with the fabric, the wind force is defined as in the following equation:

$$F_{wind} = -C_{wind}[n \cdot (V_{wind} - V)]n \tag{9}$$

where F_{wind} is the wind force, C_{wind} the wind coefficient, n the unit normal on the fabric surface.

3.2.5 Sewing force. Zhong and Xu (2009) considered sewing force between two panels of fabric as external force in mass spring model. Sewing forces are imposed on the particles on the seam lines of two fabric panels, and the force direction are determined by the directional vectors between P_i and P_j (Figure 15). Sewing force is calculated as in the following equation (Zhong and Xu, 2009):

$$F_{sewing(i)} = \begin{cases} \varepsilon X_{ij} - C_{air} V_i + \Delta A_a & |X_{ij}| > \delta \\ c\frac{X_{ij}}{|X_{ij}|} & |X_{ij}| \leqslant \delta \end{cases} \tag{10}$$

where $F_{sewing(i)}$ is the sewing force at P_i, X_{ij} is distance between P_i and P_j, C_{air} the air resistance coefficient, V_i the velocity of P_i, ΔA_a the acceleration correction, ε and c are correction coefficients of distance and δ the distance tolerance.

3.2.6 Water force. Chen *et al.* (2012) simulated behavior of wet fabric. Fick's law used to compute the water diffusion during time in the fabric structure. For this purpose, mass of the particles in the model is assumed to be time dependent (Chen *et al.*, 2012).

Friction force and collision response are other external forces. Because of the importance of these forces in sixth section, we discuss them in details.

4. The super elasticity effect

In the method of fabric simulation based on spring-mass model, behavior of force-elongation is assumed to be linear. When a small element of the fabric is exposed to a large concentrated force, large spring deformation will cause unnatural stretching and compression of fabric simulation. Local deformation appears less realistic than the real fabric deformation; this is called as super elasticity effect (Zhenfang and Bing, 2012). An example of super elasticity effect is shown in Figure 16. Figure 16(a) shows the initial position of fabric hanging by two fixed corner. Figure 16(b) shows the second position of fabric after passing time. It is observed from Figure 16 that deformation of the fabric is very locally concentrated around the corners and the deformation rate decreases very fast as being away from corners (Provot, 1995).

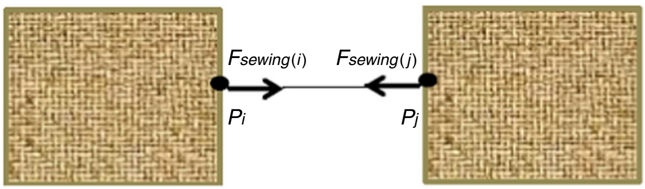


Figure 15. Sewing force between P_i and P_j

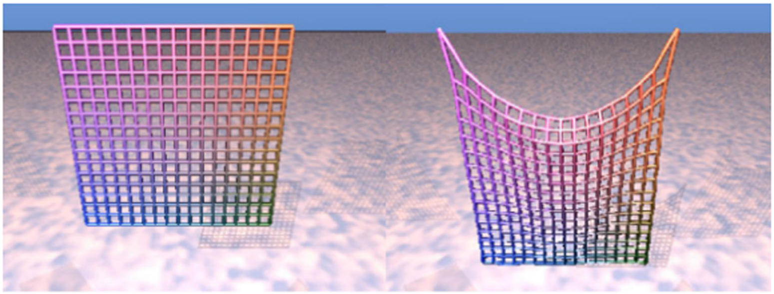


Figure 16. The super elasticity effect

Notes: (a) Initial position of fabric; (b) fabric position after passing time
Source: Provot (1995)

Such a local deformation never occurs in real fabrics, because real fabrics do not have super elasticity properties. The deformation of fabric is nonlinear to force, and their stiffness increases very fast when the deformation rate increases. When very high loads are applied rupture occurs before any large deformation can take place. Most researchers considered maximum deformation rate of most woven fabrics around 10 percent. But it is less for some linen and calico fabric for instance. A simple way to settle the super elasticity problem is to increase spring stiffness coefficient. But by changing the spring stiffness coefficient, instability in model can be happened. According to Equation (10) for a given time step *dt* and a mass *m*, there is a critical stiffness value K_c. If spring stiffness coefficient be considered more than the critical stiffness, the system is divergent:

$$K_c = m\frac{dt^2}{\pi^2} \tag{11}$$

According to Equation (11) to increase stiffness, time step (*dt*) have to decrease. So computation cost will be increased.

Some methods presented by researchers to settle the super elasticity problem are shown in Table V.

4.1 Position correction

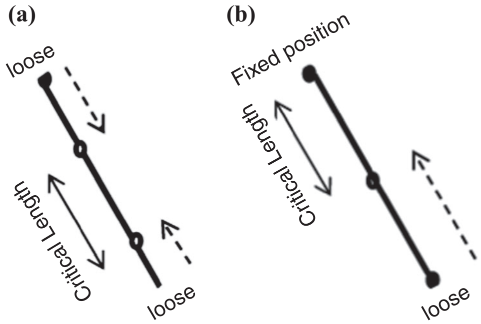
Provot (1995) used position correction to avoid the super elasticity effect. In this method, deformation rates of all springs are computed at each time step. If deformation rate of a spring is greater than a critical threshold, then two ends of the spring move toward each other along their axis so that its deformation rate exactly equals critical threshold. If only one end of the spring is loose, that spring end will move (Figure 17) (Provot, 1995).

When the length of each spring is adjusted, the length of other springs should be modified according to this change. Provot did not present how to determine the order in which the super elongated springs are adjusted at each time step, which it is a disadvantage of position correction. So position correction method is effective when only a few of springs are super elongated at each time step.

Though the position correction method can effectively restrain the exceeded elongation and compression springs, it is possibility to introduce extra self-collision among fabric. Another advantage of this method is that primary it allows the springs to over elongate and compressed and then tries to adjust their length by position modifying.

Table V.
The methods presented by researchers to settle the super elasticity problem

S. No.	Method	References
1.	Position correction	Provot (1995), Louchet *et al.* (1995), Rusinko and Swan (2012), Meyer *et al.* (2001), Desbrun *et al.* (1999), Simnett (2009)
2.	Position correction with ordering	Kang *et al.* (2000)
3.	Velocity correction	Vassilev (2000, 2001), Vassilev and Spanlang (2000), Bridson *et al.* (2002), Vassilev *et al.* (2001)
4.	Position and velocity correction	Zhong and Xu (2009), Liu *et al.* (2012a, b), Selle *et al.* (2009)
5.	Position and velocity correction by ordering	Dochev and Vassilev (2003) and Zhenfang and Bing (2012)
6.	*Using non-linear spring*	
	Using non-linear curve for spring behavior	Zhibin and Zhanli (2006)
	Using modified structural spring	Han *et al.* (2009)
	Using Biphasic springs	Chen *et al.* (1998)
7.	Adjusting spring stiffness coefficient	Bayraktar (2002), Chittaro and Corvaglia (2003)



Notes: (a) Spring with two loose ends; (b) spring with one loose end

Figure 17.
Position correction method to settle the super elasticity problem

4.2 Position correction with ordering

Kang *et al.* (2000) modified position correction method by putting in order of spring length adjustment. In this method, a bucket sorting algorithm is used to determine the adjustment order of springs. The ordering is performed by a two-phase process. First, the maximum and minimum elongation lengths of springs out of allowed range are fined. After finding the maximum and minimum elongation of springs, buckets are created. If there are *m* buckets, the interval [minimum, maximum] is divided into *m* subintervals. Each bucket is assigned as one subinterval. In the second phase, the locational assignment of each spring is determined. After all the springs are classified into *m* buckets, each spring is adjusted in accordance with the order of buckets (Kang *et al.*, 2000).

4.3 Velocity correction

Vassilev and Spanlang (2000) used velocity correction method to eliminate large deformation. In this method, at each iteration it checks for each spring whether it deformation rate is greater than a critical threshold or not. If this is the case, the velocities are modified, so that further elongation or compression is not allowed. Let P_1 and P_2 be the positions of the end points of a spring found as over elongated and compressed, and V_1 and V_2 be their corresponding velocities (Figure 18). The velocities V_1 and V_2 are split into two components V_{1t} and V_{2t}, along the line connecting P_1 and P_2, and V_{1n} V_{2n}, perpendicular to this line. It can be observed from Figure 15 that the components causing the spring to stretch, are V_{1t} and V_{2t}. Then in this method, for every over elongated and compressed spring, V_{1t} and V_{2t} are set to eros. Therefore further deformation of the fabric is not allowed (Vassilev and Spanlang, 2000).

Bridson *et al.* (2002) adapted velocity correction to settle the super elasticity problem. Different between Bridson and Vassilev method is that Bridson used momentum conserving law to corrective velocities of over elongated and compressed springs to ensure that all springs are deformed by a maximum critical threshold at the end of the step time (Bridson *et al.*, 2002).

4.4 Position and velocity correction

Zhong and Xu (2009), applied position and velocity correction method to prevent the super elasticity effect. In this method, if deformation rate of a spring is greater than a critical threshold, then two ends of the spring move each other along their axis so that its deformation rate exactly equals critical threshold. After position correction, the velocities of the two end point of spring are adjusted. The disadvantage of this method is time consuming than position correction method (Zhong and Xu, 2009).

4.5 Position and velocity correction by ordering

Zhenfang and Bing (2012) used position and velocity correction by ordering to overcome the super elasticity problem. This method requires less computation than position and velocity method. At first, each spring is marked with a priority level. Once the spring has been corrected, the priority level of the spring will be one level lower. This method is shown in Figure 19.

In Figure 19(a), springs P_0P_1 and P_1P_2 are at initial state, distance L refers to natural spring length between them. After simulation for a time step, these two springs have super elasticity problem (Figure 19(b)). At first, length of spring P_0P_1 is adjusted, and then priority of P_1 by one level decreased (Figure 19(c)). And then length of spring P_1P_2

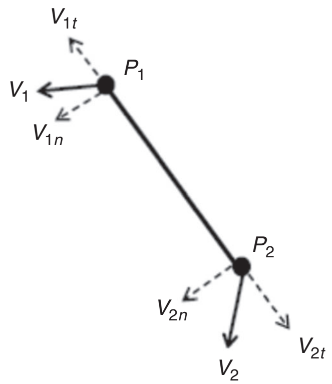


Figure 18.
Velocity correction to eliminate the super elasticity problem

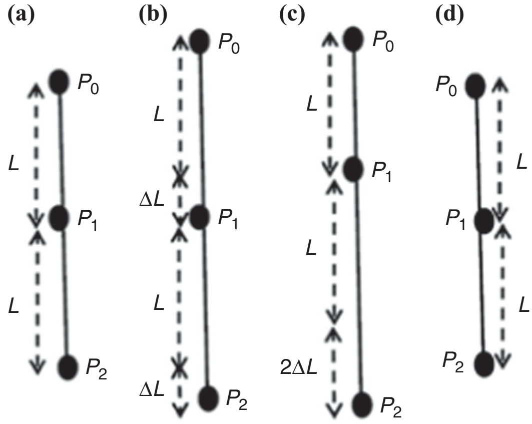


Figure 19. Position and velocity correction by ordering to overcome the super elasticity problem

Notes: (a) Initial state; (b) super elasticity; (c) the first correction; (d) the second correction

is corrected. Since priority of P_1 is lower, correct position of P_2 will be corrected (Figure 19(d)). It can be observed from Figure 16 that the two springs after correction have no phenomenon of super elasticity (Zhenfang and Bing, 2012).

4.6 Adjusting spring stiffness coefficient

Bayraktar *et al.* (2007), modified spring stiffness coefficient to prevent over elongation or compression. In this method, the spring coefficient is adjusted according to the following equation (Bayraktar *et al.*, 2007):

$$K_{new} = \begin{cases} k_{orginal} \times \frac{L}{L_0} & \text{if} \quad L \geqslant l_{allowed} \\ k_{orginal} & \text{else} \end{cases} \tag{12}$$

where K_{new} is the new stiffness coefficient, $K_{orginal}$ the original stiffness coefficient, L the spring length, L_0 natural length of spring and $l_{allowed}$ the maximum spring length that is allowed without modifying the spring coefficient.

Chittaro and Corvaglia (2003) adopts the following approach to prevent super elasticity. If a fast simulation is required by setting a low resolution for the mesh, it does not take actions to prevent super elasticity because elasticity is minimal. The super elasticity effect tends to affect meshes whose resolution (i.e. the number of mass and spring elements) is high. If in simulation the resolution of the mesh is high, the system offers the possibility of increasing rigidity settings to minimize super elasticity effects and decreasing the size of the integration step to prevent instability (Chittaro and Corvaglia, 2003).

4.7 Using non-linear spring

Zhibin and Zhanli (2006) designed a non-linear adjustment curve to overcome the super elasticity problem. The presented curve for spring behavior is expressed as Figure 20 (Zhibin and Zhanli, 2006).

The equation of the curve which shown in Figure 20 is as follows:

$$F = \frac{0.2 \times L_0}{\pi} \times \arctan\left(\frac{5\pi \times \Delta L}{L_0}\right) \tag{13}$$

where F is the force, ΔL the spring elongation and L_0 the natural length of spring.

Han *et al.* (2009) presented a kind of modified structural spring which uses a non-linear function to describe the spring force-elongation relationship. In this method the velocities tangential components of over elongated and compressed springs decreased. After several iterations of this procedure, springs lengths reach the critical threshold (Han *et al.*, 2009).

Chen *et al.* (1998) presented biphasic spring. Coefficient of Biphasic spring under small displacement is constant. And at large displacements, the stiffness of Biphasic spring is increased according to the amount of displacement (Chen *et al.*, 1998).

4.8 Comparison of the above-mentioned five kinds of methods to overcome super elasticity problem

In order to compare the performance of different methods, five common methods for settling super elasticity in fabric simulation are implemented and compared to each other. The considered methods as follows: position correction, position correction with ordering, velocity correction, position and velocity correction, position and velocity correction by ordering method.

Figure 21 shows result of fabric simulation without any methods for overcome of super elasticity problem.

It is observed from Figure 21 that large spring deformation cause unnatural stretching of fabric simulation. Figure 21 clearly shows the elongation of the springs in the corners is very high compared to all the other positions. Therefore the deformation of the fabric is very locally concentrated around the corners. Such a local deformation never occurs in fabrics. So; various methods are used to settle super elasticity problem.

Table VI shows simulation of fabric drape on table by using five mentioned methods to overcome the super elasticity. Simulation parameters as follows: mesh type: rectangular, number of mass points: 2,500, rate of super elasticity: 10 percent.

According to the images from Table VI, the differences between the five approaches are not so noticeable in the simulation results.

Figure 22 shows the time used in the five common methods for settling super elasticity to make the cloth to become balance.

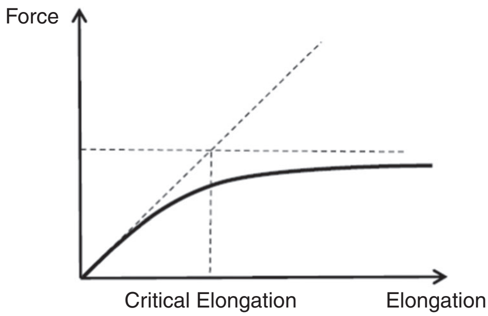


Figure 20. Using non-linear spring to settle the super elasticity problem

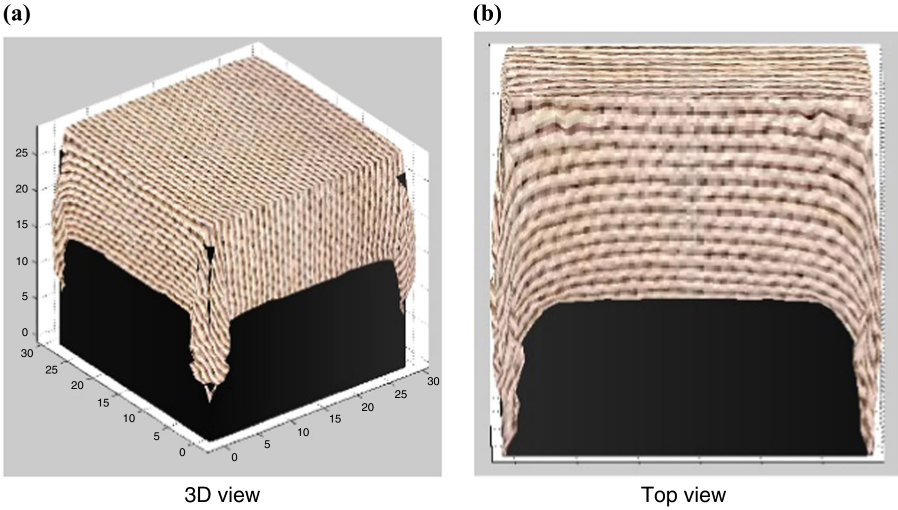


Figure 21.
Fabric drape in the presence of super elasticity (top view)

Table VI.
Simulation of fabric drape on table by using common methods to overcome the super elasticity (Front and top view)

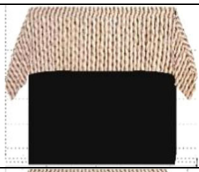
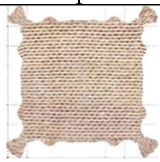
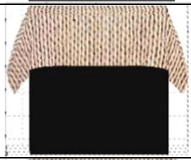
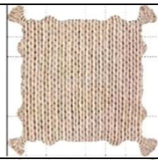
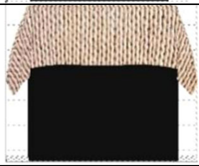
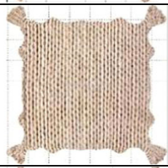
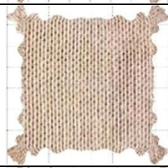
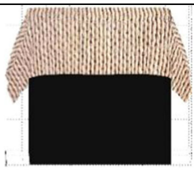
Method	Front view	Top view
Position correction		
Position correction with ordering		
Velocity correction		
Position and velocity correction		
Position and velocity correction by ordering		

Figure 22 shows that position correction with ordering method has the least computational cost. On the contrary, position and velocity correction method has the highest computational cost. According to the Figure 22, computational time of the ordering technique is about 27 percent lesser than methods without ordering, which is due to the fact that in ordering methods, less iterations of the super elasticity handling is necessary per time step.

In methods where both velocity and position are corrected, computational cost is about 5 percent more than techniques that only position or velocity is corrected. The increase of 5 percent is due to an increase in the overall execution time as a result of additional calculation.

By considering that fabric appearance is almost similar in various methods, it is better that position correction with ordering method to be used in real-time simulation that time is very important.

5. Methods of numerical integration

To solve the differential equations of physical simulation based on mass spring, integration is needed. Integration is a process of simulation for calculating mass point positions and velocities in fabric model by considering applied force on the points. There are several methods of numerical integration to solve the differential equations in mass spring model. Integration efficiency depends on four factors: computing time in time step, time step, accuracy and stability. The most common of integration methods used by researchers for fabric simulation based on mass spring model are shown in Table VII.

5.1 Explicit method

Explicit method calculates mass point position at next time step by using direct extrapolation of the pervious state. This method is simple and fast, but the time step must be short to obtain stability and accuracy. So, using this method cannot be simulated the fabric behavior on real time. Among the explicit integrators, most important are the Euler, Midpoint, Runge-Kutta and Verlet methods (Durupinar, 2004).

5.1.1 Euler explicit method. In this method, end position of time step will be predicted by using the slope (first derivative) at beginning of time step that is shown in Figure 23. Euler explicit method is the simplest method for numerical integration, but the convergence of this method is low. The accuracy of this method depends on the time step and by increasing time step, error will be increase. In Euler explicit, mass point position,

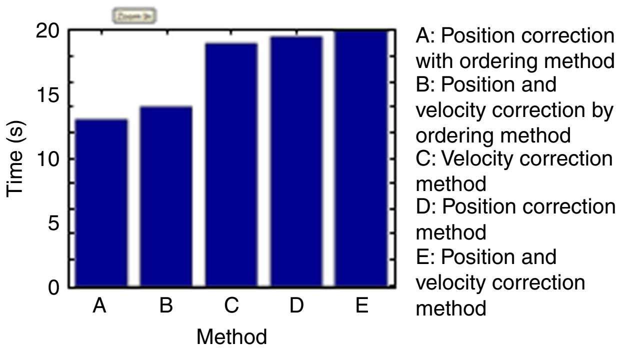


Figure 22. Computational time in fabric simulation using five methods of super elasticity

S. No.	Integration method	References
1.	*Explicit*	
	Euler	Vassilev (2001), Provot (1995), Louchet *et al.* (1995), Vassilev and Spanlang (2000), Dochev and Vassilev (2003), Ji *et al.* (2006a, b), Zhibin and Zhanli (2006), Haiyan and Zhaofeng (2008), Ozgen *et al.* (2010), Qing *et al.* (2010), Fuhrmann *et al.* (2003), Cho *et al.* (2010), Zhengdong and Shuyuan (2011), Villard and Borouchaki (2005), Zhang and Yuen (2001), Dai *et al.* (2001), Zhang *et al.* (2012), Bender *et al.* (2009) Oh *et al.* (2002), Yang and Shang (2013), Vassilev and Rousev (2008), Chittaro and Corvaglia (2003)
	Midpoint	Volino *et al.* (1995), Yang and Shang (2013), Chittaro and Corvaglia (2003)
	Forth order Runge-Kutta	Zhong and Wang (1998), Bayraktar *et al.* (2007), Connor and Stevens (2003), Li *et al.* (2013), Yang and Shang (2013), Tang *et al.* (2013), Chittaro and Corvaglia (2003)
	Adaptive Runge-Kutta	Wang *et al.* (2009)
	Verlet	Mongus *et al.* (2012), Asmaa and Elsayed (2011), Smith (2011), Raymaekers *et al.* (2005), Simnett (2009), Yuan *et al.* (2010)
2.	Implicit	Young *et al.* (2001), Baraff and Witkin (1998), Horiba *et al.* (2007), Liu *et al.* (2012a, b), Mesit *et al.* (2007), Huang *et al.* (2014), Chen *et al.* (2012), Liu *et al.* (2012a, b), Meyer *et al.* (2001), Desbrun *et al.* (1999), Kang *et al.* (2000), Thomaszewski *et al.* (2009), Volino and Thalmann (2002), Goldenthal (2010), Kang *et al.* (2014), Hu *et al.* (2013), Liu (2010), Tang *et al.* (2013), Zhou *et al.* (2008)
3.	Semi implicit	Oh *et al.* (2004, 2006), Zhou *et al.* (2005), Choi and Ko (2002), Ye *et al.* (2009), Tang *et al.* (2013)
4.	Combine Runge-Kutta with Euler implicit	Bridson *et al.* (2003) and Han *et al.* (2009)

Table VII.
The most common of integration methods used by researchers for fabric simulation based on mass spring model

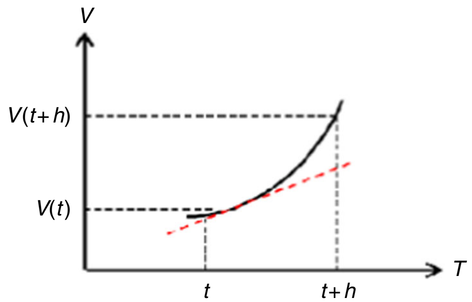


Figure 23.
Explicit Euler method

velocity and acceleration are defined as in the following equation (Durupinar, 2004):

$$a_i(t+h) = \frac{1}{m}F_i(t)$$
$$V_i(t+h) = V_i(t)+h \times a_i(t+h)$$
$$p_i(t+h) = p_i(t)+h \times V_i(t+h) \qquad (14)$$

where a is the particle acceleration, F the applied force on particle at time t, V the particle velocity, P the particle position and h the time step.

5.1.2 Midpoint method. Midpoint method is the modified Euler method. This method uses Euler method to predict mass point position at the midpoint of the time step. Then, this predicted position is used to calculate a slope in midpoint. This slope is then used to extrapolate linearly value of mass point position in at the end of the time step that is shown in Figure 24. Accuracy of this method is more than Euler method. In Midpoint method, mass point position, velocity and acceleration are defined as in the following equation (Chapra and Canle, 2010):

$$a_i(t+h) = \frac{1}{m}F_i\left(t+\frac{h}{2},\ V_i(t)+\frac{h}{2}\right)$$

$$V_i(t+h) = V_i(t)+h \times a\left(t+\frac{h}{2}\right)$$

$$P_i(t+h) = P_i(t)+h \times V_i(t+h) \tag{15}$$

5.1.3 Fourth order Runge-Kutta method. The most popular Runge-Kutta methods are fourth order. In this method, at each step of the integration, four slopes are calculated as follow: slop at initial point (k_1), slope at midpoint (k_2), slope correction at midpoint (k_3) and slope at end point (k_4), that are shown in Figure 25 (Etzmuss *et al.*, 2003). Accuracy and convergence of fourth order Runge-Kutta method is more than Euler method, but this method has high computational costs. In this method, mass point position, velocity and acceleration are defined as in the following equation

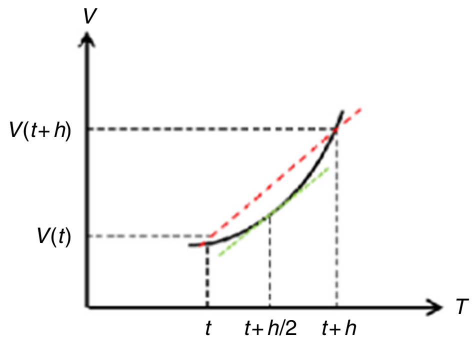


Figure 24.
Midpoint method

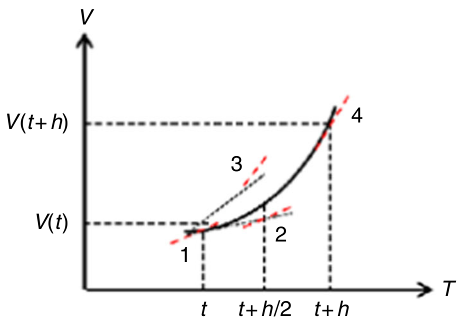


Figure 25.
Fourth order Runge-Kutta method

(Chapra and Canle, 2010):

$$a_i(t+h) = \frac{1}{6}(k_1+2k_2+2k_3+k_4)$$

$$V_i(t+h) = V_i(t)+\frac{h}{6}(k_1+2k_2+2k_3+k_4)$$

$$P_i(t+h) = P_i(t)+\frac{h}{6}(k_1'+2k_2'+2k_3'+k_4') \tag{16}$$

where coefficients k_1'-k_4' and k_1-k_4 are defined as in the following equations:

$$k_1 = F(t,\ V_i(t));\quad k_1' = V(t, P_i(t)) \tag{17}$$

$$k_2 = F\left(t+\frac{h}{2}, V_i(t)+\frac{h}{2}k_1\right);\ k_2' = V\left(t+\frac{h}{2}, P_i(t)+\frac{h}{2}k_1'\right) \tag{18}$$

$$k_3 = F\left(t+\frac{h}{2}, V_i(t)+\frac{h}{2}k_2\right);\ k_3' = V\left(t+\frac{h}{2}, P_i(t)+\frac{h}{2}k_2'\right) \tag{19}$$

$$k_4 = F(t+h, V_i(t)+hk_3);\ k_4' = V\left(t+h, P_i(t)+hk_3'\right) \tag{20}$$

5.1.4 Adaptive Runge-Kutta method. Wang *et al.* (2009) proposed adaptive Runge-Kutta method to solve differential equations in fabric simulation based on mass spring model. Difference between this method with Euler, Midpoint and Runge-Kutta methods is that the stable region of these three methods is fixed. If the selected time step is too large, it will lead to results divergence. If the selected time step size is too short, it will waste a lot of computing time. But adaptive Runge-Kutta method can automatically change time step according to the specific situation of the simulation. Wang study showed that accuracy of this method is more than Euler, Midpoint and Runge-Kutta methods. In adaptive Runge-Kutta method, mass point position, velocity and acceleration are defined as in the following equation (Wang *et al.*, 2009):

$$a_i(t+h) = \left(\frac{16}{135}k_1+\frac{6{,}656}{12{,}825}k_3+\frac{28{,}561}{56{,}430}k_4-\frac{9}{50}k_5+\frac{2}{55}k_6\right)$$

$$V_i(t+h) = V_i(t)+h\left(\frac{16}{135}k_1+\frac{6{,}656}{12{,}825}k_3+\frac{28{,}561}{56{,}430}k_4-\frac{9}{50}k_5+\frac{2}{55}k_6\right)$$

$$P_i(t+h) = P_i(t)+h\left(\frac{16}{135}k_1'+\frac{6{,}656}{12{,}825}k_3'+\frac{28{,}561}{56{,}430}k'_4-\frac{9}{50}k_5'+\frac{2}{55}k_6'\right) \tag{21}$$

where coefficients k_1-k_6 and k_1'-k_6' are defined as in the following equations:

$$k_1 = F(t, V_i(t));\ k_1' = V(t, P_i(t)) \tag{22}$$

$$k_2 = F\left(t+\frac{h}{4}, V_i(t)+\frac{h}{4}k_1\right);\ k_2' = V\left(t+\frac{h}{4}, P_i(t)+\frac{h}{4}k_1'\right) \tag{23}$$

$$k_3 = F\left(t+\frac{3}{8}h, V_i(t)+\frac{3}{32}hk_1+\frac{9}{32}hk_2\right); \; k'_3 = V\left(t+\frac{3}{8}h, P_i(t)+\frac{3}{32}hk'_1+\frac{9}{32}hk'_2\right) \quad (24)$$

$$k_4 = F\left(t+\frac{12}{13}h, V_i(t)+\frac{1{,}932}{2{,}197}hk_1-\frac{7{,}200}{2{,}197}hk_2+\frac{7{,}297}{2{,}197}hk_3\right)$$

$$k'_4 = V\left(t+\frac{12}{13}h, P_i(t)+\frac{1{,}932}{2{,}197}hk'_1-\frac{7{,}200}{2{,}197}hk'_2+\frac{7{,}297}{2{,}197}hk'_3\right) \quad (25)$$

$$k_5 = F\left(t+h, V_i(t)+\frac{439}{216}hk_1-8hk_2+\frac{3{,}680}{513}hK_3-\frac{845}{4{,}104}hk_4\right)$$

$$k'_5 = V\left(t+h, P_i(t)+\frac{439}{216}hk'_1-8hk'_2+\frac{3{,}680}{513}hK'_3-\frac{845}{4{,}104}hk'_4\right) \quad (26)$$

$$k_6 = F\left(t+\frac{h}{2}h, V_i(t)-\frac{8}{27}hk_1+2hk_2-\frac{3{,}544}{2{,}565}hk_3+\frac{1{,}859}{4{,}104}hk_4\right)$$

$$k'_6 = V\left(t+\frac{h}{2}h, P_i(t)-\frac{8}{27}hk'_1+2hk'_2-\frac{3{,}544}{2{,}565}hk'_3+\frac{1{,}859}{4{,}104}hk'_4\right) \quad (27)$$

5.1.5 Verlet method. Verlet method avoids velocity and acceleration calculation for determining of mass point position. In this method, mass point position at next time will be predicted using difference between the current and previous positions. The accuracy and stability of this method is more than Euler method and computation cost of Verlet method is as same as Euler method. In this method, calculation of mass point position is expressed as in the following equation (Smith, 2011):

$$P_i(t+h) = P_i(t)+(P_i(t)-P_i(t-h))+a_i(t) \times h^2 \quad (28)$$

Verlet method does not consider the loss of energy due to friction within fabric. Therefore, an equilibrium state for the model is never achieved. Thus, Mongus *et al.* (2012) used Verlet optimized method to solve differential equations in fabric simulation based on mass spring model. Equation (30) is Verlet optimized equation by considering dumping factor:

$$p_{ij}(t+h) = p_{ij}(t)+D \times \left(p_{ij}(t)-p_{ij}(t-h)\right)+a \times h^2 \quad (29)$$

where D is the damping factor which is a coefficient from the interval [0, 1]. Mongus considered the value $D = 0.99$ in their experiments.

5.2 Implicit method

Implicit method deduces the mass point position at next time step indirectly from an extrapolation of the next state. This method, in spite of explicit method is stable at long time step. But disadvantage of this method is more computation cost than explicit. The most common implicit method is Euler implicit.

5.2.1 Euler implicit method. In Euler implicit method, end position of time step will be predicted using the slope (first derivative) at end of time step that is shown in Figure 26. The accuracy and stability of this method are more than Euler explicit. In this method, mass point acceleration is defined as in the following equation:

$$a_i(t+h) = \frac{1}{m} F_i(t+h) \tag{30}$$

where $F_i(t+h)$ is applied force on particle at time $t+h$, which can be approximated with a first-order derivative as in the following equation:

$$F_i(t+h) = F_i(t) + H \times (\Delta P_i(t+h)) \tag{31}$$

where matrix H denotes Hessian matrix which is related to the fabric deformation and defined as Equation (32). Also $\Delta P_i(t+h)$ is calculated as Equation (33):

$$\frac{\partial F_i(t)}{\partial P_i(t)} \tag{32}$$

$$\Delta P_i(t+h) = (V_i(t) + \Delta V_i(t+h)) \times h \tag{33}$$

where $\Delta V_i(t+h)$ can be rewritten as:

$$\left(I - \frac{h^2}{m} H \right) \times (\Delta V_i(t+h)) = (F_i(t) + hHV_i(t)) \frac{h}{m} \tag{34}$$

where I is identity matrix.

Euler implicit method is able to use longer time steps without loss of stability, but it has a critical weakness that involves matrix $I - (h^2/m)H$, which is a large-sized matrix. Because of this matrix, a large linear system must be solved at every integration step. Desbrun *et al.* (1999) proposed an efficient method that approximates the Hessian matrix (H). In the method, H_{ij}, the entry of the Hessian matrix at the ith row and the jth column, was approximated as in the following equation (Desbrun *et al.*, 1999):

$$\begin{aligned} &H_{ij} = K_{ij} \\ &H_{ii} = -\sum_{i \neq j} K_{ij} \quad \text{if } i \text{ connected with } j \\ &H_{ii} = 0 \qquad\qquad \text{if } i \text{ not connected with } j \end{aligned} \tag{35}$$

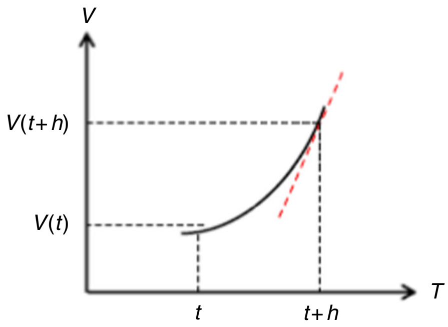


Figure 26.
Euler implicit method

where K_{ij} is the stiffness coefficient of the spring between the ith and the jth mass points. K_{ij} is 0 when the ith and the jth mass springs are not linked. Then, the matrix $I-(h^2/m)H$ remains constant during simulation. This technique produces simple calculations and stable results.

However, Debsrun method requires much time to computation cost. Also in this method the mass, stiffness coefficient and time step cannot be changed. Kang *et al.* (2000) proposed a fast and simple technique to approximate the Hessian matrix. In the method to simplify Hessian matrix, stiffness coefficient is assumed to be constant for all the spring in the mesh. So, Hessian matrix is rewritten as in the following equation:

$$\begin{aligned} H_{ij} &= K \\ H_{ii} &= n_i K \end{aligned} \tag{36}$$

where K is the spring stiffness coefficient and n_i is the number of neighboring mass points that are linked to the ith mass point (Kang *et al.*, 2000).

5.3 Semi-implicit method

The use of semi-implicit method in fabric simulation was first proposed by Baraff and Witkin (1998). This method is more convergence and less time consuming than implicit method. Difference between semi-implicit with implicit method is how to calculate velocity and force approximation at next time. In this method, force at next time is defined as in the following equation:

$$F_i(t+h) = F_i(t) + \frac{\partial F_i}{\partial P_i}(t) \times \Delta P_i + \frac{\partial F_i}{\partial P_i}(t) \times \Delta V_i \tag{37}$$

In semi-implicit method, mass point position, velocity and acceleration are calculated as in the following equation:

$$\begin{aligned} a_i(t+h) &= \frac{1}{m} F_i(t+h) \\ V_i(t+h) &= h \times H \times F_i(t+h) \\ P_i(t+h) &= h \times V_i(t+h) \end{aligned} \tag{38}$$

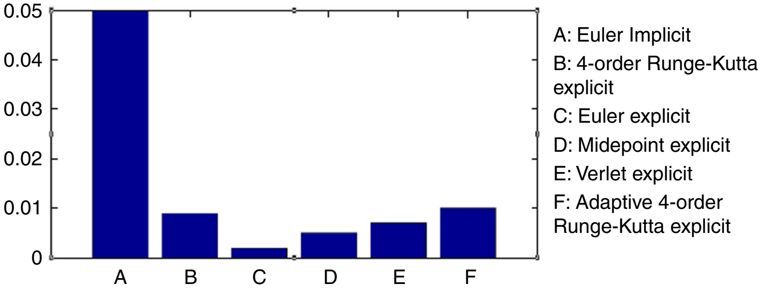
5.4 Comparison of the above-mentioned six kinds of algorithms of numerical integration

The common numerical integration in fabric simulation are Euler, Midpoint, Verlet, four-order Runge-Kutta and Adaptive Runge-Kutta, which are all based on explicit method and Euler implicit method.

Value of time step and the stability region of the every method have a direct relation. If the method has a large stability region, a large time step can be select, which can improve the efficiency in the repeated calculation. On the contrary, in order to avoid the system diverges, a smaller time step should choose.

Figure 27 shows maximum time step that six numerical integration algorithms are stable in it. In order to comparison of various methods, the basic properties of the fabric model such as mesh type, number of mass points and super elasticity method were the same. Only the algorithm changed.

It is observed from Figure 27 that explicit methods require smaller step time than implicit methods to avoid diverge. So explicit method is not stable when the time steps of the simulation are very large.



Notes: Simulation parameters: mesh type: rectangular, number of mass points: 2,500, super elasticity method: position correction method with ordering

Figure 27. Maximum time step for six methods of numerical integration

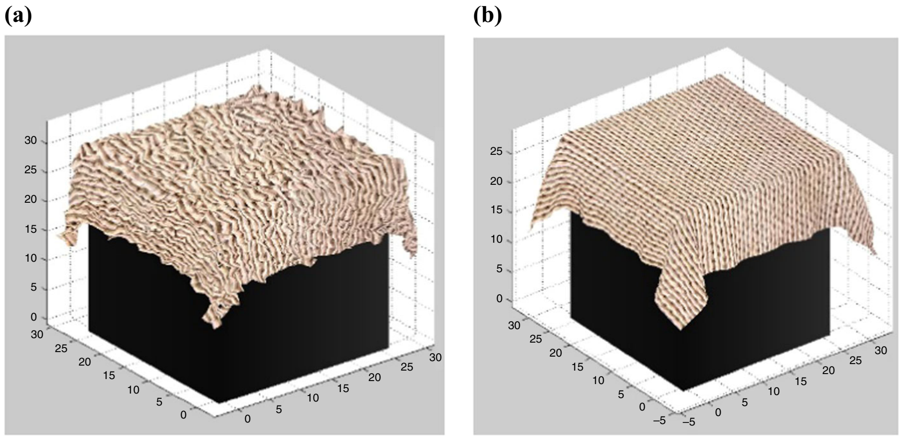
By considering results of Figure 27, at the same time, repetition of algorithm operation in explicit method is more than implicit method. Although the implicit method need small step time, it also suffers from computational overhead because it must solve a large linear system. However, complexity and calculated amount in implicit method is more than the same order of explicit. Considering to the real time of simulation, explicit is generally used in fabric simulation.

When the time steps gets bigger of maximum values which are presented in Figure 27 for each algorithm, the fabric takes a chaotic shape, as demonstrated in Figure 28.

Results of comparison mentioned numerical algorithm in this part are similar to proposed results by some researchers (Young *et al.*, 2001; Baraff and Witkin, 1998; Wang *et al.*, 2009; Fuhrmann *et al.*, 2003; Zhenfang and Bing, 2012; Kang *et al.*, 2000; Zhang and Yuen, 2000).

6. Collision

When an object starts to move in any kind of environment, there are chances that will be bumped into something. If nothing is done about this in a simulation, the object will



Notes: (a) Large time step; (b) small time step

Figure 28. Large vs small time steps of a fabric simulation using Euler explicit algorithm

penetrate and then pass through other objects. In fabric simulation, not only collisions between fabric and surface should be considered, but it is also necessary to consider self-collisions between different parts of fabric due to the delicate nature and hence ease of deformation of fabric. In collision handling, there are two issues that must be discussed: collision detection and collision response.

6.1 Collision detection

One of the simplest illustrative situations to consider for collision detection and response is that of a particle traveling at constant velocity toward a stationary (Figure 29). To detect collision of the particle with the plane, its planar equation should be determined:

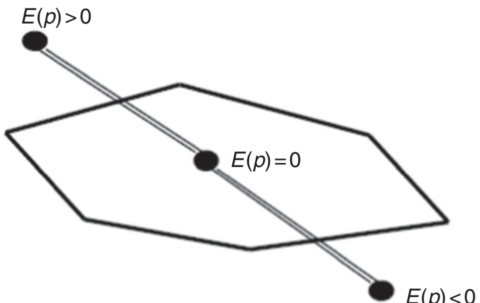
$$E(P) = aP_x + bP_y + cP_z + d = 0 \tag{39}$$

where $E(P)$ is the planar function, a, b, c, d are coefficients of the planar equation and p the particle position in space. The planar equation can be formed so that for points above the plane the planar equation evaluates to a positive value, $E(P) > 0$; for points below the plane, $E(P) < 0$. At each time step, the particle is tested to see if it is still above the plane, $E(P(t_i)) > 0$. As long as this evaluates to a positive value, there is no collision. For the first time of t which $E(P(t_i)) < 0$ indicates that the particle has collided with the plane at interval $[t_i, t_{i\ -1}]$ (Chen *et al.*, 1998).

In complex model, collision ability of all mass point at fabric model with other object in environment and self-collision among fabric model should be investigated. These collisions can be divided into point-triangle collisions and edge-edge collisions. The point-triangle collision occurs when a node of mesh collides with a triangle of other mesh. In edge-edge collision, one edge of a triangle collides with an edge of other triangle.

6.1.1 Point-triangle collision. By considering Figure 30, suppose P_4 is initially at A, and following the direction of its velocity, V_4 reaches B at time t, then a collision has occurred during this time step if there is a t for which B is coplanar with triangle P_1 P_2 P_3 (following the directions V_1, V_2 and V_3), the roots of the Equation (40) shows the time when the four points will be coplanar (Chen *et al.*, 1998):

$$(P_{21} + tV_{21}) \times (P_{31} + tV_{31}) \cdot (P_{41} + tV_{41}) = 0 \tag{40}$$



Source: Chen *et al.* (1998)

Figure 29.
Collision particle with the plane

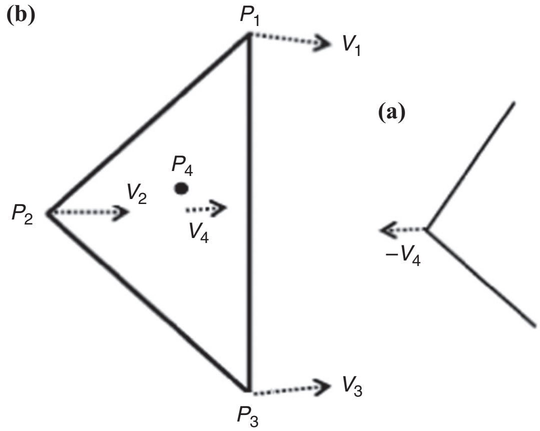
where:

$$P_{ij} = P_j - P_i$$

$$P_{ij} = V_j - V_i \quad (41)$$

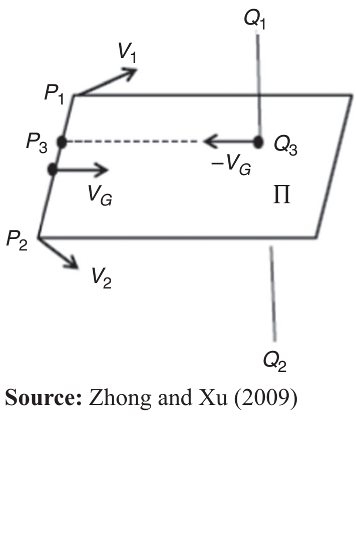
6.1.2 Edge-edge collision. In the edge-edge collision (edge P_1P_2 and edge Q_1Q_2 in Figure 31), velocity of centroid of edge P_1P_2 as the moving direction toward the triangle edge Q_1Q_2 (V_G) is determined. The possible colliding point between P_1P_2 and Q_1Q_2 on a plane Π can be found by V_G. Assume that Q_1Q_2 intersects with the moving plane Π at point Q_3. The incident ray (Q_3, $-V_G$) will intersect with P_1P_2 at P_3, which is the potential collision point on edge P_1P_2. If the distance between Q_3 and P_3 is smaller than the tolerant distance ζ, a collision will occur (Zhong and Xu, 2009).

6.1.3 Collision detection optimization. Collision handling, and especially collision detection, is the most time consuming part in fabric simulation. Indeed, the collision detection between a fabric model with N mass points and an object with M nodes has a



Source: Zhong and Xu (2009)

Figure 30. Point-triangle collision



Source: Zhong and Xu (2009)

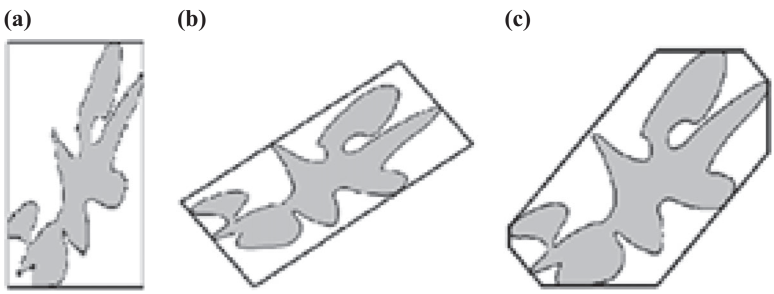
Figure 31. Edge-edge collision

$O(M \times N)$ complexity and the self-collision detection has a $O(N^2)$ complexity. So, there is a need to reduce calculation complexity. Researchers have used several optimization algorithms to collision detection in fabric simulation based on mass spring model. These methods are shown in Table VIII.

Bounding boxes hierarchy. Bounding boxes hierarchy was first proposed by Volino and Thalmann (1994). The core idea of this method is wrapping up the complex geometric object with a bounding box. Then dissect every box into two smaller boxes. Do the same things to the dissected fabric that enclosed till the bounding box includes only a basic geometrical unit (triangle). So the bounding boxes hierarchy is built. In the bounding boxes hierarchy algorithm, first carry on intersection test between two bounding boxes. Only when the bounding boxes have intersected will further intersection calculation of the wrapped geometric object be processed. Therefore, the collision detection algorithm can be significantly improved by parsing the bounding box tree while eliminating rapidly collisions tests between elements that belong to two zones whose bounding boxes do not intersect. There are many kinds of bounding boxes (Provot, 1997). The mostly used bounding boxes are axis aligned bounding box (AABB), oriented bounding box (OBB) and discrete orientation polytopes (k-DOPs). Figure 32 presented object wrapping using these three methods.

S. No.	Optimization method	References
1.	*Bounding boxes hierarchy*	
	AABB	Vassilev (2000), Bayraktar *et al.* (2007), Bridson *et al.* (2002), Zhibin and Zhanli (2006), Durupinar and Gudukbay (2007), Zhong and Xu (2009), Liu *et al.* (2012a, b), Fuhrmann *et al.* (2003), Huang *et al.* (2014), Zhang and Yuen (2001), Guimaraes and Silva (1991), Zhou *et al.* (2005), Liu *et al.* (2012a, b), Zhenfang and Bing (2012), Durupinar (2004), Li *et al.* (2014), Hu *et al.* (2013)
	K-DOP	Hasler *et al.* (2007), Goldenthal (2010)
2.	Image space	Vassilev (2001) and Vassilev and Spanlang (2000)
3.	Uniform grid	Shou *et al.* (2013)
4.	Minimal enclosure	Ji *et al.* (2006a, b)
5.	Voxel based	Oh *et al.* (2004), Cho *et al.* (2010), Zhang and Yuen (2001), Choi and Ko (2002), Meyer *et al.* (2001)
6.	Surface curvature	Zhibin and Zhanli (2006)

Table VIII. Methods of collision detection optimization



Notes: (a) ABB; (b) OBB; (c) K-DOP
Source: Klosowski *et al.* (1998)

Figure 32. Object wrapping using algorithms of collision detection optimization

AABB: AABB is the most widely used collision detection algorithm by researchers in fabric simulation. Advantage of this method than the other methods is quickly updating an AABB tree as a model is deforming. An AABB tree is constructed top-down, by recursive subdivision that is shown as Figure 33. At each recursion step, the smallest AABB of the set of primitives is computed. This process continues until each subset contains one element in fabric model (Huang *et al.*, 2014).

K-DOPs: the major difference between this method with AABB is that in the AABB trees, higher order k-DOPs are used to object wrapping, instead of axis aligned bounding boxes. The most common k-DOPs are displayed in Figure 34 (Hasler *et al.*, 2007). As discovered by Mezger *et al.* (2002) 14-DOPs have the highest performance when used for bounding fabric.

Image space. This method use depth buffer and normal buffer of the graphics acceleration hardware to optimize collision detection. In image space method, the body is painted with the color representing the appropriate body normal. This is done by setting the color of each vertex (R, G, B) to the value of the vertex normal (x, y, z). The rendering processes generate two depth buffers and two normal buffers for the front and the back of the scanned body that are shown in Figures 35 and 36. To check for a

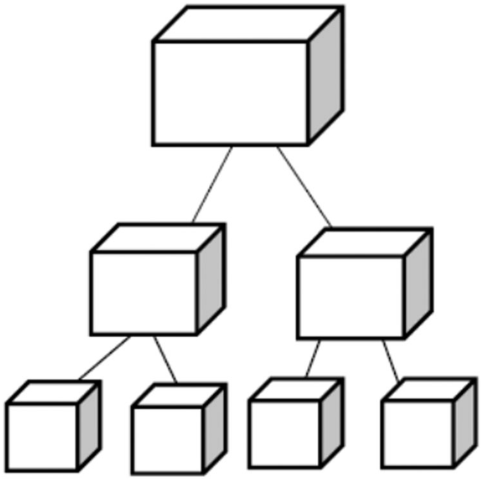
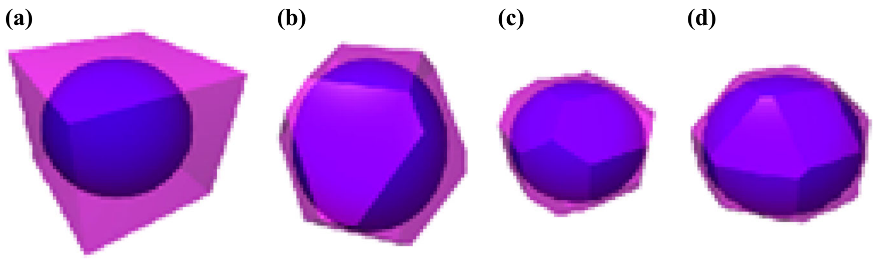
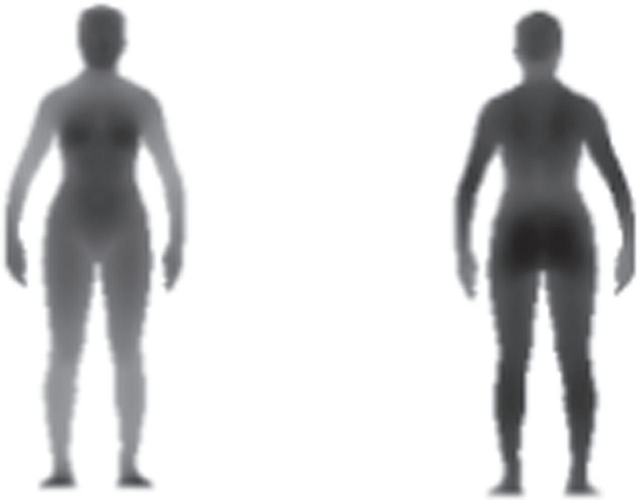


Figure 33.
Collision detection optimization (AABB method)



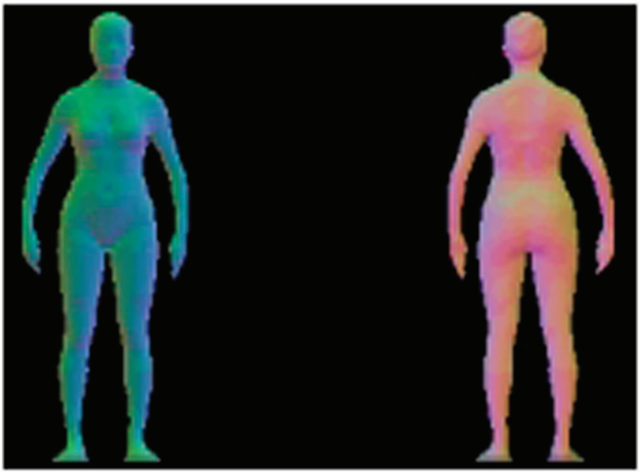
Notes: (a) 6; (b) 14; (c) 18; (d) 26-DOPs
Source: Mezger *et al.* (2002)

Figure 34.
The most common discrete-oriented polytopes



Source: Vassilev and Spanlang (2000)

Figure 35.
Depth buffer



Source: Vassilev and Spanlang (2000)

Figure 36.
Normal buffer

collision we convert the appropriate vertex coordinates of the fabric to an index in the depth and normal buffer. A check for collision is accomplished by simply comparing the *z*-value of the vertex coordinate with the corresponding value in the depth buffer (Vassilev and Spanlang, 2000).

Vassilev and Spanlang (2000) compared image space and bounding boxes hierarchy method. Vassilev study showed that image space is much faster, because of not only its time per iteration is shorter, but also produces a smaller number of iterations (Vassilev and Spanlang, 2000).

Minimal enclosure. Feng Ji *et al.* (2006a, b) introduced minimal enclosure to facilitate fast and reliable detection of fabric – body and fabric – fabric collisions. For collision detection between fabric and body, each body part is first considered as a group of l closed loops, and each loop is constructed by connecting n points. Then each loop is enclosed by a column, and every two vertically adjacent columns are connected by a TDC that are shown in Figure 37. Therefore, the $l \times n$ body points are replaced by l columns and $l-1$ TDC. Before the detection, each fabric particle would be examined against $l \times n \times 2$ body triangles, whereas now it is examined against only $2l-1$ body triangles. Also Feng Ji mentioned two special cases for self-collision detection: some of fabric parts are impossible to collide, so they are excluded in collision checking; fabric triangle created with neighboring triangles initially contact or intersect each other, thus collisions or penetrations between these triangles are also excluded in the checking process (Ji *et al.*, 2006a, b).

Uniform grid. Shou *et al.* (2013), proposed a new method to optimize collision detection based on uniform grid that remarkably reduces the computation cost, this method included three steps as follows: first, constructing a 3D uniform grid for each body model that the length of grid edge is determined by the average length of all triangular edges in body. As in Figure 38, for each triangular surface, grids occupied this triangle can be calculated and then store the reference of it into grids that it occupied. Second, for each triangle edge in fabric model, in each interval time, the triangle goes through grids can be determined. Then, via the references out of all triangles in those grids can be gated, and

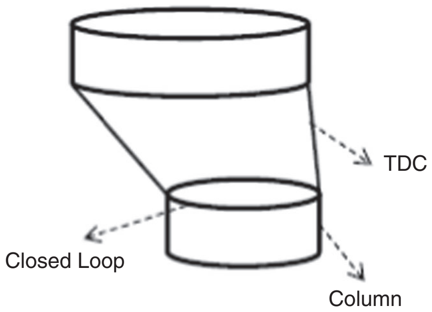
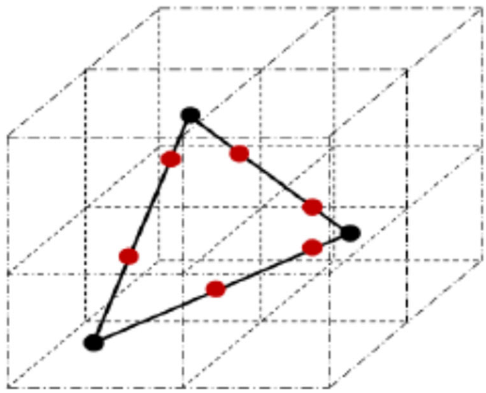


Figure 37.
Collision detection optimization (minimal enclosure method)



Source: Shou *et al.* (2013)

Figure 38.
3D uniform mesh for body model

test which triangles would intersect with the moving edge. Fourth, get the triangle which intersected point is the near the starting point of the moving edge is the triangle as the result of collision detection (Shou *et al.*, 2013).

Voxel based. In this method, body model divide into uniform sub-volumes called voxels that are arranged in a regular and rectilinear form. Each voxel contains several triangles of the body model that as shown in Figure 39. Therefore, the potential collision fabric triangles for an edge are limited in the voxels associating with the edge. The voxel size is very important to the voxel-based method. To increase the accuracy of collision detection, a smaller voxel size should be selected. On the other hand, each voxel should contain as few triangles as possible. As the voxel size becomes smaller, a triangle will penetrate through more voxels. It will bring about lots of computational cost. Obviously, very small voxels are not suitable for dynamic simulation (Zhang and Yuen, 2000).

Surface curvature. This method has been implemented in the case of self-collision detection. This optimization is based on the following property: when a given zone has a low curvature it cannot self-intersect, so all the zones it includes do not intersect with each other. The curvature of a zone will be evaluated by the set of normal vectors of the triangle belonging to the zone (Figure 40). In this method, a cone which includes these normal vectors with angle α at its vertex is computed, if $\alpha < \pi$, the zone cannot self-intersect (Provot, 1997).

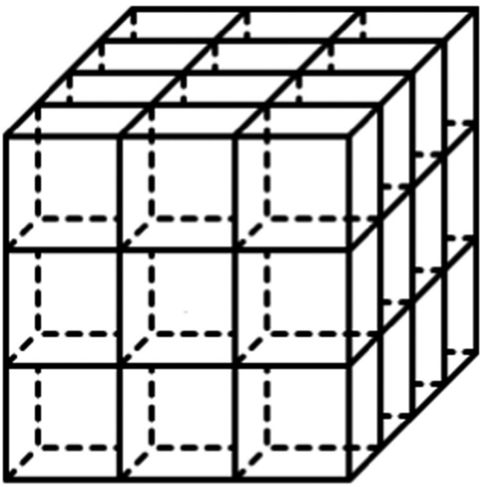
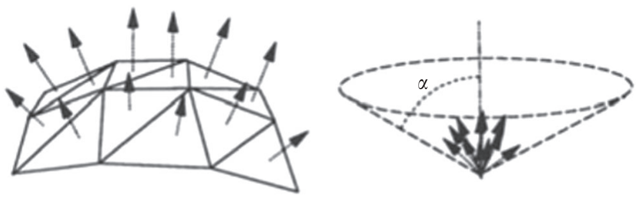


Figure 39. Collision detection optimization (Voxel-based method)



Source: Provot (1997)

Figure 40. Cone including normal to triangles of a zone of the fabric surface

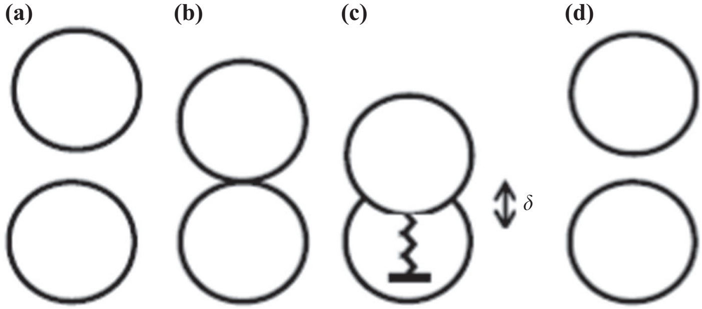
6.2 Collision response

After a self-collision or a collision between fabric and body is detected, a reaction is needed to prevent the penetration and collision among them. This reaction is called collision response. For the collision response step in the fabric simulation, several methods were proposed by researchers. These methods can be classified as physical and geometrical approaches. Collision response schemes that use the repulsive force between the objects are called physical methods. Approaches which modify positions and velocity of colliding object are referred to as geometric response (Liu *et al.*, 2012b). In addition to physical and geometrical methods, some researchers are considered friction force as collision response. The collision responses are considered by researchers for fabric simulation based on mass spring model, are shown in Table IX.

6.2.1 Repulsive force. This method uses a repulsive force in the normal direction of particle motion to return the initial state. There are two methods that use repulsive force to collision response in fabric modeling based on mass spring as: penalty method and inverse dynamic. In the penalty method, a spring with a zero rest length is momentarily attached from the penetration point that it imparts a restoring force (Figure 41).

Table IX. The collision responses are considered by researchers for fabric simulation based on mass spring model

S. No.	Collision response	References
1.	Repulsive force	Baraff and Witkin (1998), Terzopoulos *et al.* (1987), Provot (1995), Louchet *et al.* (1995), Oh *et al.* (2004, 2006), Guimaraes and Silva (1991), Zhou *et al.* (2005), Choi and Ko (2002), Desbrun *et al.* (1999), Chen *et al.* (1998), Carignan *et al.* (1992)
2.	Modify positions and velocity	Vassilev (2000, 2001), Zhong and Wang (1998), Vassilev and Spanlang (2000), Bayraktar *et al.* (2007), Bayraktar (2002), Bridson *et al.* (2002), Ji *et al.* (2006a, b), Durupinar and Gudukbay (2007), Liu *et al.* (2012a, b), Fuhrmann *et al.* (2003), Zhang *et al.* (2012), Bridson *et al.* (2003), Meyer *et al.* (2001), Zhenfang and Bing (2012), Vassilev *et al.* (2001), Ye *et al.* (2009), Tang *et al.* (2013), Chittaro and Corvaglia (2003)
3.	Friction force	Baraff and Witkin (1998), Zhong and Wang (1998), Vassilev and Spanlang (2000), Vassilev (2000), Bayraktar *et al.* (2007), Bridson *et al.* (2002), Dochev and Vassilev (2003), Shou *et al.* (2013), Selle *et al.* (2009), Fuhrmann *et al.* (2003), Oh *et al.* (2004, 2006), Li *et al.* (2013), Rusinko and Swan (2012), Guimaraes and Silva (1991), Choi and Ko (2002), Chen *et al.* (2012), Bridson *et al.* (2003), Meyer *et al.* (2001), Desbrun *et al.* (1999), Bender *et al.* (2009), Smith (2011), Ye *et al.* (2009)



Notes: (a) Before collision; (b) moment of collision; (c) particles penetration; (d) after collision

Figure 41. Penalty method for collision response

The spring imparts a force on the point in the direction of the penetrated surface normal and with a magnitude according to Hooke's law in the following equation:

$$F_c = -K\delta \tag{42}$$

where F_c is the collision force, K the spring stiffness coefficient and δ the value of particles penetration.

Though this method is easy to implement, is not ideal. Choosing the optimal value for the spring stiffness coefficient is difficult. If the spring stiffness coefficient is too weak, then the collision will not be corrected immediately and value of particles penetration will be much. If the spring stiffness coefficient is too strong, high contact force will be applied on the colliding particles; then the colliding surfaces will be thrown apart in an unrealistic manner (Chen *et al.*, 1998). Implementing the penalty method produces the particles motion according to Figure 42.

It is observed from Figure 33 that penalty method is fully elastic and do not consider energy dissipation. A damper is paralleled with the spring to dissipate energy (Figure 43). In the case, F_c is computed using the following equation (Yang and Shang, 2013):

$$F_c = -K\delta - DV_\delta \tag{43}$$

where D is the damper coefficient and V_δ the velocity of particles penetration.

Inverse dynamic is another method that uses repulsive force to collision response. In this method, collision is inelastic. Such collisions between two particles are characterized by the fact that their velocity after they collide equals the velocity of their

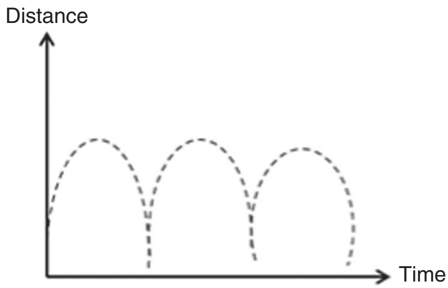
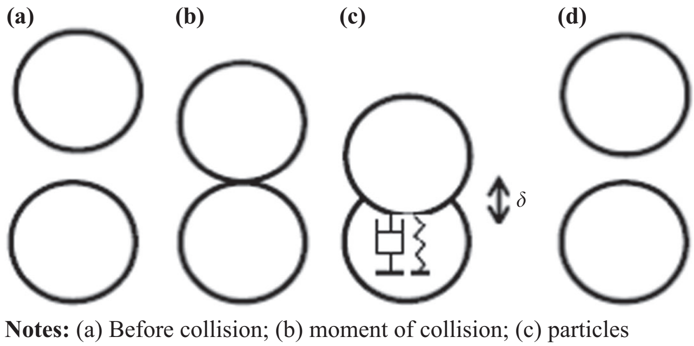


Figure 42.
Particle motion in penalty method



Notes: (a) Before collision; (b) moment of collision; (c) particles penetration; (d) after collision

Figure 43.
Penalty method modifying for collision response

mass centers of before their collision in the following equation:

$$V_i(t+h) = V_c \tag{44}$$

where V_c is the mass centers velocity of two particles before they collide and $V_i(t+h)$ the particle speed after collision that i stand for 0 and 1 according to colliding particle. V_c is calculated as in the following equation:

$$V_c = \frac{m_1 V_1(t) + m_2 V_2(t)}{m_1 + m_2} \tag{45}$$

where m_1 and m_2 are masses of colliding particles, V_1 and V_2 are velocity of colliding particles.

Assuming that forces are constant in the time step, velocity after the collision will be as in the following equation:

$$V_i(t+h) = V_i(t) + \frac{(F_i + F_{c_i})\Delta t}{m_i} \tag{46}$$

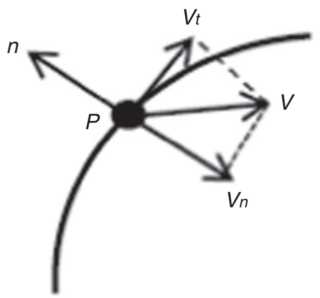
where Δt is the time step, F the resultant of external and internal forces on particle and F_c the unknown force which when added to F makes $V_c = V_i \times F_c$ is determined by considering the following equations (Carignan *et al.*, 1992):

$$F_{c_i} = \frac{m_1 m_2 (V_2(t) - V_1(t))}{(m_1 + m_2)\Delta t} - F_i \tag{47}$$

Disadvantage of repulsive force is that collision response will be adopted after penetration. In fact, at first penetration is occurred and then system has a reaction to it.

6.2.2 Modify positions and velocity. In this method, after collision detection, position and velocity of colliding object are corrected. Several methods are presented by researchers to modify position and velocity. Bridson *et al.* (2002), Zhong and Xu (2009), Ye *et al.* (2009) and Liu *et al.* (2012a, b), used momentum conserving law to correct velocity after collision. This method is appropriate when two colliding object are moving.

Vassilev (2000) used friction and reflection coefficients to correct velocity after collision that is shown in Figure 44. P is colliding point, V is velocity of particle before collision and n is surface normal at the colliding point. Then velocity after collision can



Source: Vassilev (2000)

Figure 44. Modify velocity for collision response

be computed as in the following equation:

$$V(t+h) = C_{fric}V_t(t) - C_{refl}V_n(t) \tag{48}$$

where $V_t(t)$ and $V_n(t)$ tangent and normal components of V, C_{fric} and C_{refl} are a friction and a reflection coefficients, which depend on the material of the colliding objects (Vassilev, 2000).

This method is appropriate when an element of fabric collide with static surface. Durupinar (2004) has modified the presented equation by Vassilev as in the following equation:

$$V(t+h) = C_{fric}V_t - C_{refl}V_n + V_{human} \tag{49}$$

where V_{human} is body velocity.

In the Durupinar method, new positions of colliding particles are corrected as in the following equation:

$$P = \Pi + N \tag{50}$$

where P is the final position of particle, Π the projection of the particle on the body triangle and N the normal vector of the collision plane (Durupinar and Gudukbay, 2007).

Fuhrmann *et al.* (2003) used position correction method for collision response. If a particle is detected to be closer to surface than a given threshold, it is set back in the direction of normal. Particle position after collision is computed by the following equation:

$$\begin{aligned} P(t+h) &= P_n(t) + P_t(t) \\ P_n(t) &= n\cdot(\xi - d) \\ P_t(t) &= -C_{fric}d_t \\ d_t &= d - n(d\cdot n) \end{aligned} \tag{51}$$

where $P(t+h)$ is the particle position after collision, $P_t(t)$ and $P_n(t)$ the tangent and normal components of position, d the distance of the particle to the surface, n the normal vector of the collision surface, ξ the threshold for collision and C_{fric} the friction coefficient (Fuhrmann *et al.*, 2003).

Tang *et al.* computed collision response by modifying position and velocity. Initially impulse forces for all the particles that are involved in collision are determined according to the following equation:

$$I = k(v\cdot n)n \tag{52}$$

where I is the impulse on particle, v the relative velocity between particles pair, n the normal vector on the contact surface and k the stiffness factor. Then, these impulses were used to update the velocities and positions of the particles that can be expressed as:

$$v(t+h) = v(t) + \frac{I}{m} \tag{53}$$

$$p(t+h) = p(t) + v(t+h)\Delta t \tag{54}$$

where $v(t+h)$ is the particle velocity after collision, $v(t)$ the particle velocity before collision, $p(t+h)$ the particle position after collision, $v(t)$ the particle position before collision (Tang *et al.*, 2013).

6.2.3 Friction force. When an element in fabric collides to other element in fabric or object in environment, friction force is applied on element opposite to the travel direction.

This force is proportional to the relative velocity between the two elements and their material. The friction force is calculated as in the following equation:

$$F_s = \mu_s V_N \tag{55}$$

where F_s is the friction force, μ_s the friction coefficient and V_N the normal components of velocity.

6.3 Comparison of the above-mentioned two kinds of collision response

Collision detection and response are very important for realistic fabric simulation. Tow common methods of collision response include penalty and modify of positions-velocity methods. In the penalty method, collision response is very sensitive to selecting spring stiffness. If spring stiffness is small, collision response not to be corrected immediately and fabric will penetrate into object (Figure 45(a)). If the spring stiffness is great, then the colliding surfaces will be thrown apart in an unrealistic manner (Figure 45(b)).

Figure 46 shows results of fabric simulation with two methods of collision response.

It is observed from Figure 47 that there is no noticeable difference in fabric appearance between two methods. Therefore, because of sensitivity of penalty method to selection spring stiffness, it is suggested that modify of positions-velocity method is used in fabric simulation. However, modify of positions – velocity methods takes larger computational time than penalty method. Difference of time used in two methods is about 13 percent in presented work.

7. Experimental example of fabric and cloth simulation

In this section, some of the empirical samples presented by researchers about fabric and cloth simulation are mentioned. Provot (1995) simulated behavior of a fabric hanging by two adjacent corners that is presented in Figure 47. Simulation parameters used by Provot are listed in Table X.

Villard and Borouchaki (2005) simulated behavior of a hanging piece of fabric by two fixed corner, fabric fall over four little spheres and fabric fall over a ball that is

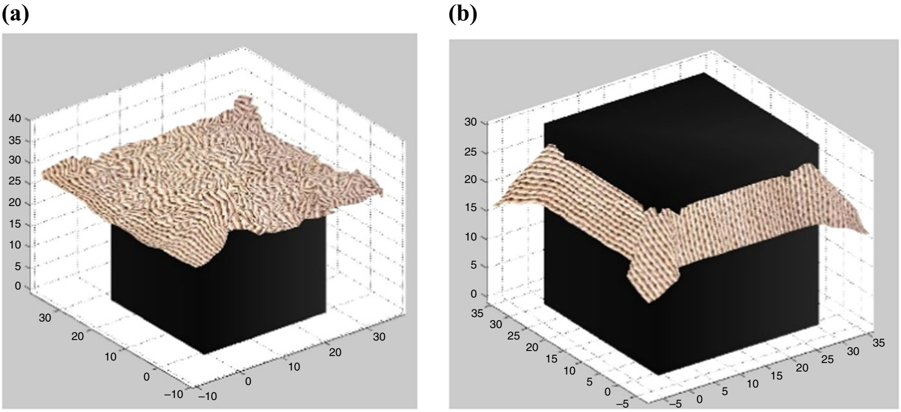
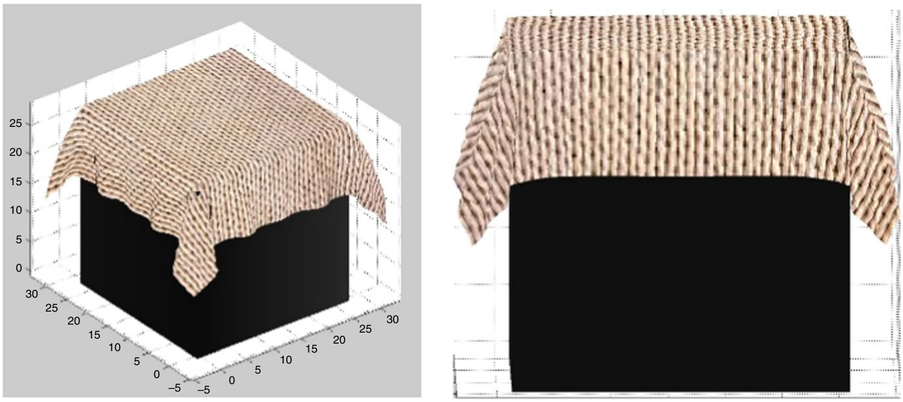


Figure 45. Fabric simulation with penalty method

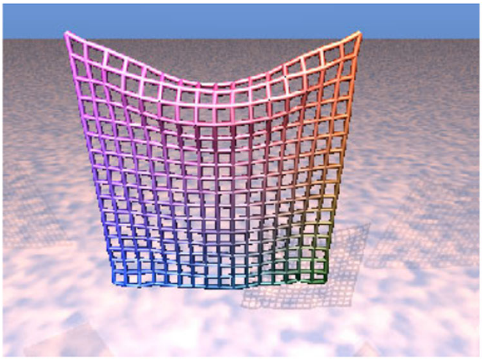
Notes: (a) Small spring stiffness; (b) large spring stiffness (Simulation parameters: mesh type: rectangular, number of mass points: 2,500, super elasticity method: position correction method with ordering, numerical integration: implicit method)

(a) (b)



Notes: (a) Penalty method; (b) modify of positions-velocity methods (simulation parameters are same as Figure 45)

Figure 46. Comparison of two common methods of collision response in fabric simulation



Source: Provot (1995)

Figure 47. Simulation behavior of fabric hanging by two adjacent corners

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Hook's law
External	Gravity – damping – wind
Super elastic problem	Position correction
Numerical integration	Euler explicit
Collision	
Collision detection	–
Collision response	Inverse dynamic

Table X. Simulation parameters used by Provot shown in Figure 48. Villard presented a new method based on adaptive meshing allowing to reduce the number of mesh elements and computational time. Simulation parameters are considered by Villard as shown in Table XI (Villard and Borouchaki, 2005).

Feng Ji *et al.* (2006a, b) simulated a rectangular fabric draping freely in a gentle wind that is illustrated in Figure 49. Five types of woven and knitted fabrics were tested and simulated. In order to describe the dynamic internal deformation of the fabric during its draping under self-weight, the increasing rate of the summed area of all triangles

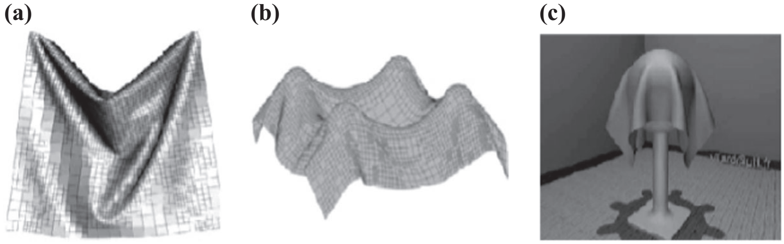


Figure 48. Simulation of fabric draping

Notes: (a) Hanging fabric by two fixed corner; (b) fabric fall over four little spheres; (c) fabric fall over a ball

Source: Villard and Borouchaki (2005)

Table XI. Simulation parameters used by Villard *et al.*

Mesh	Adaptive mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Hook's law
External	Gravity – air resistance – wind
Super elastic problem	Position correction
Numerical integration	Euler explicit
Collision	
Collision detection	–
Collision response	Inverse dynamic

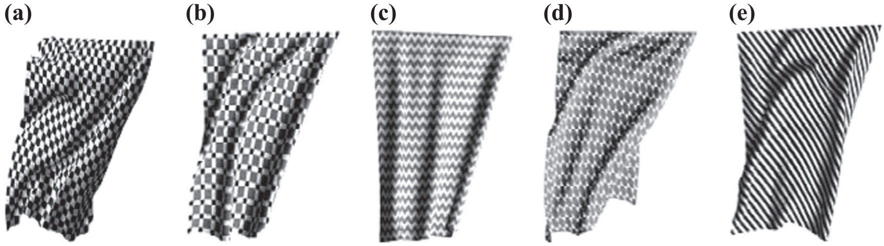


Figure 49. Simulation of dynamic draping behavior of fabrics hanging in the wind

Notes: (a) Plain woven fabric; (b) twills woven fabric; (c) interlock knitting fabric; (d) fancy weft knitting; (e) tricot stitch fabric

Source: Ji *et al*. (2006a, b)

composing the fabric sheet was calculated as a function of time to represent the dynamic face strain of the fabric, and the average strain of all structural springs was calculated to represent the dynamic 1D strain of the fabric. Simulation parameters used by Feng Ji *et al.* (2006a, b) are listed in Table XII.

Horiba *et al.* (2007) proposed the interactive fabric simulation considering airflow. As a result, Horiba could simulate the fabric behavior in airflow on real time (see Figure 50). Simulation parameters presented by Horiba are listed in Table XIII.

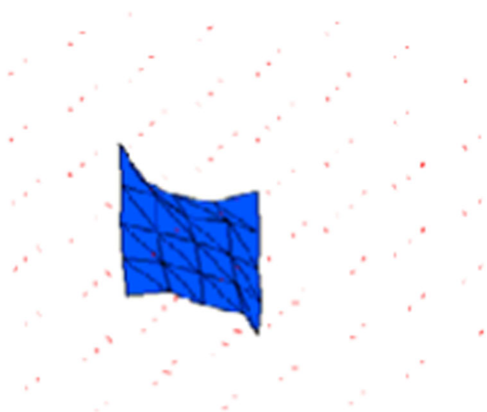
Haiyan and Zhaofeng (2008) simulated fabric draping behavior by four adjacent corners that can be show in Figure 51. Haiyan used simulation parameters according to Table XIV (Haiyan and Zhaofeng, 2008).

Liu *et al.* (2012a), simulated draping behavior of skirt that is shown in Figure 52. Liu used a new method for bend behavior of fabric in mass spring model. In this method, bend force measured by curvature angle among two adjacent triangles which share the same edge and change rate of angle. Simulation parameters used by Liu are presented in Table XV (Liu *et al.*, 2012a, b).

Zhang *et al.* (2012) simulated static draping of the square fabric on a round table that is shown in Figure 53. In their study, three kinds of plain woven fabric which are different in weight and thickness were chosen. Also, Zhang simulated behavior of skirt

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Using Kawabata test
Shear	Using Kawabata test
Bend	Using Kawabata test
External	Gravity – air resistance – wind
Super elastic problem	–
Numerical integration	Euler explicit
Collision	
Collision detection	–
Collision response	–

Table XII. Simulation parameters used by Feng Ji *et al.*

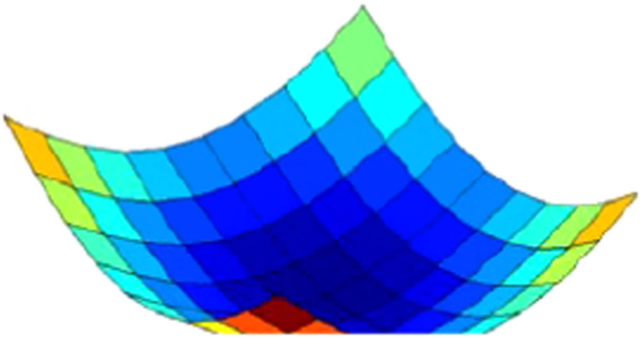


Source: Horiba *et al.* (2007)

Figure 50. Simulation behavior fabric in airflow

Table XIII. Simulation parameters used by Horiba *et al.*

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Hook's law
External	Gravity – air resistance – wind
Super elastic problem	–
Numerical integration	Euler explicit
Collision	
Collision detection	–
Collision response	–



Source: Haiyan and Zhaofeng (2008)

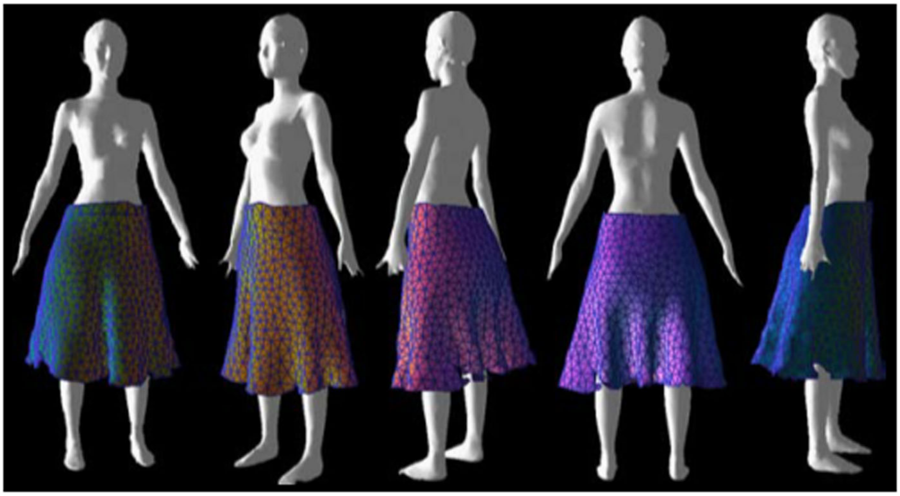
Figure 51. Simulation fabric draping behavior by four adjacent corner

Table XIV. Simulation parameters used by Haiyan *et al.*

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Hook's law
External	Gravity – damping
Super elastic problem	–
Numerical integration	Euler explicit
Collision	
Collision detection	–
Collision response	–

draping constructed by these fabrics (see Figure 54). The suitable end-use skirts of three kinds of fabric were analyzed according to the skirt draping shape. Zhang used simulation parameters according to Table XVI (Zhang *et al.*, 2012).

Chen *et al.* (2012) proposed a technique that simulates wet clothing for virtual human in the rain that is shown in Figure 55. Also, they simulated wet fabric on movable ball.

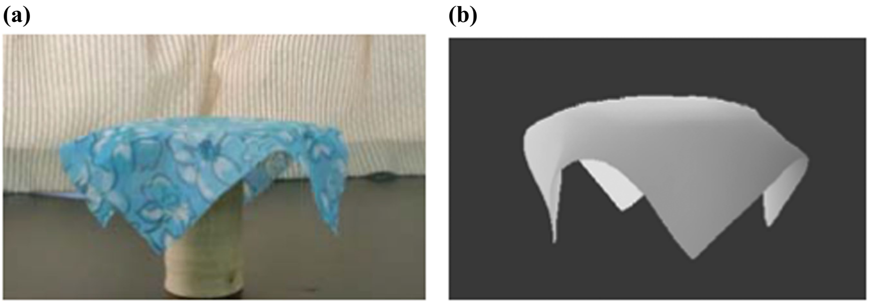


Source: Liu *et al*. (2012a, b)

Figure 52. Simulation draping behavior of skirt

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Using a model based on curvature and change rate of the angle
External	Gravity – damping
Super elastic problem	–
Numerical integration	Euler implicit
Collision	
Collision detection	AABB
Collision response	–

Table XV. Simulation parameters used by Liu *et al.*



Notes: (a) Real fabric; (b) simulated fabric

Source: Zhang *et al*. (2012)

Figure 53. Simulation of static draping of the square fabric on a round table

Notes: (a) Side view; (b) above view
Source: Zhang *et al*. (2012)

Figure 54.
Simulation of skirt draping behavior

Table XVI.
Simulation parameters used by Zhang *et al.*

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Using Kawabata test
Shear	Using Kawabata test
Bend	Using Kawabata test
External	Gravity – air resistance
Super elastic problem	–
Numerical integration	Euler explicit
Collision	
Collision detection	AABB
Collision response	Modify positions and velocity

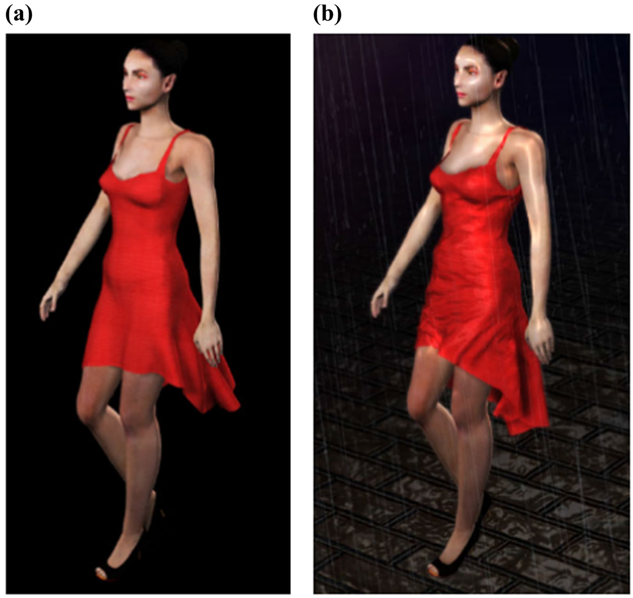
Water is gradually dropping on the fabric until that piece of fabric is fully saturated (see in Figure 56). For this purpose, masses points in fabric model are considered as time dependent. Simulation parameters used by Chen are listed in Table XVII (Chen *et al.*, 2012).

Huang *et al.* (2014) simulated fabric that falls under the effect of gravity and fabric colliding with a ball that are presented in Figure 57. Huang simplified the Provot's mesh to improve the efficiency of computing. Result showed that it does not affect the model significantly whether it is with two shear springs or one shear spring. Therefore, one of the shear springs was eliminated to simplify the model. Huang used simulation parameters according to Table XVIII (Huang *et al.*, 2014).

Some of the commercial packages that used mass spring modeling for fabric simulation are 3D Max, Maya, Marvelous designer and Optima software

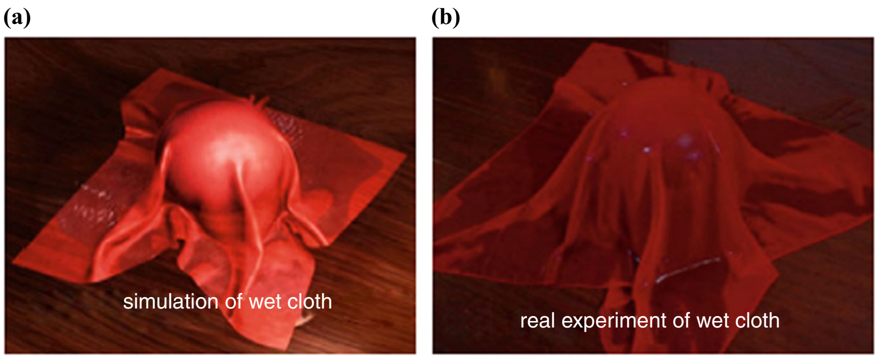
7.1 Challenges and limitations in fabric simulation

7.1.2 Challenges. One of the researchers concerns in fabric modeling problem is difference between theoretical and experimental results. So in the mass spring model,



Notes: (a) Dry; (b) wet
Source: Chen *et al*. (2012)

Figure 55. Simulation of cloth behavior



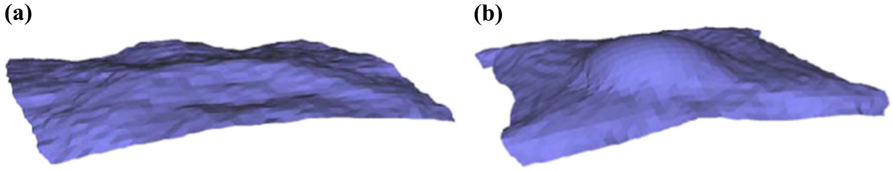
Notes: (a) Real fabric; (b) simulated fabric
Source: Chen *et al*. (2012)

Figure 56. Simulation of wet fabric on movable ball

it is required to set the model parameters describing deformation behavior. In this regard, a few optimized-based approaches have been carried out to recover the mass spring parameters in fabric simulation by correcting the model parameters according to the experimental result. For instance, Louchet *et al.* (1995) used genetic algorithm to optimize the mass spring model parameters in fabric simulation. The model parameters consist of spring stiffness, elongation rate and natural length of spring in stretch, bend and shear cases. They showed validity of the optimized model by recovering the model parameters in case of hanging a simulated fabric from two corners (Louchet *et al.*, 1995).

Table XVII.
Simulations parameters used by Chen *et al.*

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Using Kawabata test
Shear	Using Kawabata test
Bend	Using Kawabata test
External	Gravity – damping – water
Super elastic problem	–
Numerical integration	Euler implicit
Collision	
Collision detection	–
Collision response	Friction



Notes: (a) Draping under the effect of gravity; (b) colliding fabric with a ball
Source: Huang *et al*. (2014)

Figure 57.
Simulation fabric behavior

Table XVIII.
Simulation parameters used by Huang *et al.*

Mesh	Rectangular mesh
Forces	
Internal	
Stretch	Hook's law
Shear	Hook's law
Bend	Hook's law
External	Gravity – damping – wind
Super elastic problem	–
Numerical integration	Euler implicit
Collision	
Collision detection	AABB
Collision response	–

Bianchi *et al.* (2003) proposed a solution to specification model parameters base on genetic algorithm. Their focus was determination of mesh topology in 2D simulation. They used finite elements model (FEM) to obtain the topology of a mass spring model. Their work results demonstrated that genetic algorithm is able to recover the topology of mass spring model and spring connections were successfully identified (Bianchi *et al.*, 2003). In the subsequent work in 2004, they extended their method to 3D model. Furthermore, they introduced a new approach to simultaneously optimize mesh topology and spring stiffness values. Linear elastic FEM deformation computations were used as reference for the model confirmation (Bianchi *et al.*, 2004). Han *et al.* (2009)

considered a range of parameter values for the mass spring model (bending stiffness, stretch stiffness and shear stiffness) for fabric simulation and in order to achieve highest compliance, they determined appropriate values for these three model parameters by using trial and error method (Han *et al.*, 2009). Mongus *et al.* (2012) used the genetic algorithm to find the best values for stretch and shear spring stiffness coefficients in mass spring modeling for fabric simulation. Optimization was done through error minimization between the model and experimental results. They used two indexes in objective function to compare simulated and real fabric behavior: DC and distribution of folds. Different textiles may produce the same DC but they differ in the number, amplitude and distribution of folds. Therefore, they used fast Fourier transformation to measure these properties (Mongus *et al.*, 2012).

Another challenge in fabric simulation is increasing speed and real-time simulation. So researchers are looking for new techniques to increase speed in numerical integration, super elasticity, collision detection and response methods (Vassilev, 2001; Shou *et al.*, 2013; Li *et al.*, 2013; Huang *et al.*, 2014; Hu *et al.*, 2013).

7.1.3 Limitations. Fabric simulation is the result of the combination of various methods that have dramatically evolved during the decade. However, there still exits some limitation. One of the limitations in fabric simulation is the lack of using fabric properties such as material, weave structure, density and so on in mass spring modeling. The fundamental work should aim to develop a new fabric model to simulate real physical and mechanical behavior of fabric. In this regard, mesh structure can be presented based on fabric structure. This issue is more important in the case of knitted fabrics, due to the difference in structure of fabrics. But researchers have less attention to this topic. And they considered mesh structure in knitted fabric as same as woven fabric.

8. Conclusion

In the edge of digital, fabric simulation technology has been considered into many fields. 3D fabric simulation is complex and its implementation requires knowledge in different fields such as textile engineering, computer engineering and mechanical engineering. Several methods have been presented for fabric simulation such as physical and geometrical models. Mass spring model, the typical physically based method, is one of the methods for fabric simulation which widely considered by researchers. This method is a fast and flexible technique with high ability to simulate fabric behavior in real time with different environmental conditions. Mass spring model has more accuracy than geometrical models and also it is faster than other physical modeling. The aim of this paper is comparison and survey on research's conducted in fabric simulation using mass spring model. The paper reviews and compares presented mesh types in mass spring model, forces applied on model, super elastic effect and ways to avoid the super elasticity effect, numerical integration methods for solving equations, collision detection and its response. And finally some of the empirical samples presented by researchers about fabric simulation are mentioned.

References

Aileni, R.M., Ciocoiu, M. and Farimap, D. (2011), "3D garesim software application for virtual garment simulation", *The 7th International Scientific Conference Learning and Software for Education Bucharest, April 28-29*.

Aono, M. (1996), "A wrinkle propagation model for cloth", *Proceedings CG International: Computer Graphics Around the World, Tokyo*.

Ascough, J., Bez, H.E. and Bricis, A.M. (1996), "A simple beam element, large displacement model for the finite element simulation of cloth drape", *Journal Textile Institute*, Vol. 87 No. 1, pp. 152-165.

Asmaa, A.E. and Elsayed, E.H. (2011), "Speeding up cloth simulation by linearizing the bending function of the physical mass-spring model", *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, May 16-19*.

Baraff, D. and Witkin, A. (1998), "Large steps in cloth simulation", *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, July 19-24*, pp. 43-54.

Bayraktar, S. (2002), "Simulating cloth behavior by using mass spring network", A thesis for the degree of master science, September.

Bayraktar, S., Gudukbay, U.O. and Ozguc, B. (2007), "Practical and realistic animation of cloth", *3DVT Conference, May 7-9*.

Bender, J., Bayer, D. and Diziol, R. (2009), "Dynamic simulation of inextensible cloth", *International Journal on Computer Science and Information Systems*, Vol. 4 No. 2, pp. 86-102.

Bianchi, G., Harders, M. and Szekely, G. (2003), "Mesh topology identification for mass-spring models", *Medical Image Computing and Computer-Assisted Intervention*, Vol. 2878 No. 11, pp. 50-58.

Bianchi, G., Solenthaler, B., Szekely, G. and Harders, M. (2004), "Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations", *Medical Image Computing and Computer-Assisted Intervention*, Vol. 3217 No. 9, pp. 293-301.

Breen, D.E., House, D.H. and Wozny, M.J. (1994), "A particle-based model for simulating the draping behavior of woven cloth", *Textile Research Journal*, Vol. 64 No. 11, pp. 663-685.

Bridson, R., Fedkiw, R.P. and Anderson, J. (2002), "Robust treatment of collisions, contact, and friction for cloth animation", *ACM Transactions on Graphics*, Vol. 21 No. 3, pp. 594-603.

Bridson, R., Marino, S. and Fedkiw, R. (2003), "Simulation of clothing with folds and wrinkles", *Symposium on Computer Animation, New York, NY, January*, pp. 28-36.

Carignan, M., Yang, Y., Thalrnann, N.M. and Thalrnanr, D. (1992), "Dressing animated synthetic actors with complex deformable clothes", *Computer Graphics, Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, Vol. 26 No. 2, pp. 99-104.

Chapra, S. and Canle, R. (2010), *Numerical Methods for Engineers*, Mc Graw-Hill, New York, NY.

Charfi, H., Gagalowicz, A. and Brun, A. (2006), "Measurement of viscosity damping parameters of fabric related to a non-linear textile model", *Textile Research Journal*, Vol. 76 No. 10, pp. 787-798.

Chen, B. and Govindaraj, M. (1995), "A physically based model of fabric drape using flexible shell theory", *Textile Research Journal*, Vol. 65 No. 6, pp. 324-330.

Chen, Y., Thalmann, N.M. and Allen, B.F. (2012), "Physical simulation of wet clothing for virtual humans", *Visual Computer*, Vol. 28 Nos 6-8, pp. 765-774.

Chen, Y., Zhu, Q. and Kaufman, A. (1998), "Physically based animation", *Proceedings of Computer Animation, Philadelphia, PA, June 8-10*.

Chittaro, L. and Corvaglia, D. (2003), "3D virtual clothing: from garment design to web3D visualization and simulation", *Proceedings of the Eighth International Conference on 3D Web Technology, New York, January 7-9*, pp. 73-85.

Cho, C.S., Park, J.Y., Boeing, A. and Hingston, P. (2010), "An implementation of a garment-fitting simulation system using laser scanned 3D body data", *Computers in Industry*, Vol. 61 No. 6, pp. 550-558.

Choi, K.J. and Ko, H.S. (2002), "Stable but responsive cloth", *ACM Transactions on Graphics*, Vol. 21 No. 3, pp. 604-611.

Connor, C. and Stevens, K. (2003), "Modeling cloth using mass spring systems", *Applied Soft Computing*, Vol. 12 No. 5, pp. 266-273.

Dai, X., Li, Y. and Zhang, X. (2003), "Simulation anisotropic woven fabric deformation with a new particle model", *Textile Research Journal*, Vol. 73 No. 12, pp. 1091-1099.

Dai, X.Q., Furukawa, T., Mitsui, S., Takatera, M. and Shimizu, Y. (2001), "Drape formation based on geometric constraints and its application to skirt modeling", *International Journal of Clothing Science and Technology*, Vol. 13 No. 1, pp. 23-37.

Desbrun, M., Schroder, P. and Barr, A. (1999), "Interactive animation of structured deformable objects", *Proceedings of Graphics Interface, Kingston, June 4-7.*

Dochev, V. and Vassilev, T. (2003), "Efficient super-elasticity handling in mass-spring systems", *International Conference on Computer Systems and Technologies*, pp. 483-490.

Durupinar, F. and Gudukbay, U. (2007), "A virtual garment design and simulation system", *11th International Conference Information Visualization, Zurich, July 4-6.*

Durupinar, F. (2004), "A 3D garment design and simulation system", A thesis for the degree of master of science.

Eberhardt, B., Weber, A. and Strasser, W. (1996), "A fast, flexible particle-system model for cloth draping", *IEEE Computer Graphics & Applications*, Vol. 16 No. 5, pp. 52-59.

Etzmuss, O., Keckeisen, M. and Strasser, W. (2003), "A fast finite element solution for cloth modeling", *Proceedings of the Pacific Conference on Computer Graphics and Applications, Alberta, October 8-10.*

Feynman, C.R. (1986), "Modeling the appearance of cloth", A thesis for the degree of master science, Department of Electrical Engineering and Computer Science, Cambridge.

Fuhrmann, A., Gro, C. and Luckas, V. (2003), "Interactive animation of cloth including self collision detection", *Proceedings of Winter School of Computer Graphics*, Vol. 11 No. 1, pp. 141-148.

Gan, L., Ly, N.G. and Steven, G.P. (1995), "A study of fabric deformation using nonlinear finite elements", *Textile Research Journal*, Vol. 65 No. 11, pp. 660-668.

Goldenthal, A. (2010), "Implicit treatment of constraints for cloth simulation", Thesis submitted for the degree of Doctor of Philosophy.

Guimaraes, M. and Silva, D. (1991), "Cloth animation with collision detection", *Proceedings of the IFIP WG 5.10 Working Conference, Tokyo, April 8-12.*

Haiyan, Y. and Zhaofeng, G. (2008), "An improved mass-spring model to simulate draping cloth", *International Conference on Intelligent Computation Technology and Automation, October 20-22, Hunan.*

Han, F., Stylios, G.K. and Watt, H. (2009), *3D Modelling, Simulation and Visualisation Techniques for Drape Textiles and Garments*, Woodhead Publishing.

Hasler, N., Rosenhahn, B., Asbach, M., Ohm, J.R. and Seidel, H.P. (2007), "An analysis-by-synthesis approach to tracking of textiles", *Motion and Video Computing*, pp. 1-8.

Hinds, B.K. and McCartney, J. (1990), "Interactive garment design", *Visual Computer*, Vol. 6 No. 2, pp. 53-61.

Horiba, H., Kitahata, S., Inui, S. and Hashimoto, M. (2007), "Study on interactive cloth simulation considering airflow", *Journal of Information and Computing Science*, Vol. 2 No. 6, pp. 103-110.

Hu, J. and Chen, S. (2000), "Numerical drape behavior of circular fabric sheets over circular pedestals", *Textile Research Journal*, Vol. 70 No. 7, pp. 593-603.

Hu, J., Huang, W., Yu, K., Huang, M. and Li, J. (2013), "Cloth simulation with a modified implicit method based on a simplified mass-spring model", *Applied Mechanics and Materials*, Vols 373-375 No. 8, pp. 1920-1926.

Huang, W., Hu, J., Yu, K., Wang, Y. and Jiang, M. (2014), "Cloth simulation based on simplified mass-spring model", *Journal of Electrical Engineering*, Vol. 12 No. 5, pp. 3811-3817.

Hutchinson, D., Preston, M. and Hewitt, T. (1996), "Adaptive refinement for mass spring simulations", *Eurographics Workshop on Animation and Simulation, Poitiers, August 31-35*.

Ji, F., Li, R. and Qiu, Y. (2006a), "Simulate the dynamic draping behavior of woven and knitted fabrics", *Journal of Industrial Textiles*, Vol. 35 No. 3, pp. 201-215.

Ji, F., Li, R. and Qiu, Y. (2006b), "Three-dimensional garment simulation based on a mass-spring system", *Textile Research Journal*, Vol. 76 No. 12, pp. 12-17.

Kang, Y.M. and Cho, H.G. (2002a), "Bilayered approximate integration for rapid and plausible animation of virtual cloth with realistic wrinkles", *Proceedings of the Computer Animation*, pp. 203-211.

Kang, Y.M. and Cho, H.G. (2002b), "Complex deformable objects in virtual reality", *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 49-56.

Kang, Y.M., Choi, J.H. and Cho, H.G. (2000), "Fast and stable animation of cloth with an approximated implicit method", *Proceedings of Computer Graphics International, Geneva, June 19-24*.

Kang, Y.M., Choi, J.H., Cho, H.G. and Lee, D.H. (2014), "Efficient and realistic animation of cloth with approximate implicit method and wrinkle", *Visual Computer*, Vol. 17 No. 12, pp. 1-12.

Klosowski, J., Held, M., Mitchell, J., Sowizral, H. and Zikan, K. (1998), "Efficient collision detection using bounding volume hierarchies of k-DOPs", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4 No. 1, pp. 21-36.

Li, F., Chen, X., Wang, L. and Zhao, Q. (2014), "Canopy frame interactions for umbrella simulation", *Computers & Graphics*, Vol. 38 No. 2, pp. 320-327.

Li, Y., Chern, L., Kim, J.D. and Li, X. (2013), "Numerical method of fabric dynamics using front tracking and spring model", *Computer Physics Communications*, Vol. 14 No. 5, pp. 1-24.

Liu, H., Zhong, Y. and Wang, S. (2012a), "Combination of implicit integration and collision response for cloth simulation", *Journal of Fiber Bioengineering & Informatics*, Vol. 5 No. 1, pp. 59-65.

Liu, H., Zhong, Y. and Wang, S. (2012b), "Converting to reusable garment model", *International Journal of Clothing Science and Technology*, Vol. 24 No. 5, pp. 308-316.

Liu, Y. (2010), "3D garment real-time simulation in character animation", *Advanced Materials Research*, Vols 108-111 No. 5, pp. 753-758.

Louchet, J., Rovot, X. and Crochemore, D. (1995), "Evolutionary identification of cloth animation models", *Proceedings of the Eurographics Workshop in Maastricht, September 2-3*, pp. 44-54.

Mesit, J., Guha, R. and Chaudhry, S. (2007), "3D soft body simulation using mass-spring system with internal pressure force and simplified implicit integration", *Journal of Computers*, Vol. 2 No. 8, pp. 34-43.

Meyer, M., Debunne, G., Desbrun, M. and Barr, A.H. (2001), "Interactive animation of cloth-like objects for virtual reality", *The Journal of Visualization and Computer Animation*, Vol. 12 No. 1, pp. 1-12.

Mezger, J., Kimmerle, S. and Etzmu, O. (2002), "Progress in collision detection and response techniques for cloth animation", *Proceedings of 10th Pacific Conference on Computer Graphics and Applications*, pp. 444-445.

Mongus, D., Repnik, B., Mernik, M. and Zalik, B. (2012), "A hybrid evolutionary algorithm for tuning a cloth-simulation model", *Applied Soft Computing*, Vol. 12 No. 8, pp. 266-273.

Ng, H.N. and Grimsdale, R.L. (1996), "Computer graphics techniques for modeling cloth", *IEEE Computer Graphics & Applications*, Vol. 16 No. 5, pp. 28-41.

Oh, S., Ahn, J. and Wohn, K. (2004), "A new implicit integration method for low damped cloth simulation", *The 5th Korea-Israel Bi-National Conference on Geometric Modeling and Computer Graphics*, pp. 115-121.

Oh, S., Ahn, J. and Wohn, K. (2006), "Low damped cloth simulation", *Visual Computer*, Vol. 22 No. 2, pp. 70-79.

Oh, S.W., Kim, H.S. and Wohn, K.Y. (2002), "Collision handling for interactive garment simulation", *Proceedings VSMM, Kyungju*, pp. 239-251.

Ozgen, O., Kallmann, M., Ramirez, L. and Coimbra, C.F. (2010), "Underwater cloth simulation with fractional derivatives", *ACM Transactions on Graphics*, Vol. 29 No. 3, pp. 1-9.

Provot, X. (1995), "Deformation constraints in a mass-spring model to describe rigid cloth behavior", *Proceeding of Graphics Interface, Quebec*, pp. 147-155.

Provot, X. (1997), "Collision and self-collision handling in cloth model dedicated to design garments", *Proceedings of Computer Animation and Simulation*, pp. 177-189.

Qing, Z., Ying, D. and Guang, G. (2010), "A fast predictive-corrective mass spring method in cloth simulation", *International Conference on Computer Application and System Modeling, Taiyuan, October 22-24*.

Raymaekers, C., Vanacken, L., Cuppens, E. and Coninx, K. (2005), "A comparison of different techniques for haptic cloth rendering", *Proceedings Haptex Workshop on Haptic and Tactile Perception of Deformable Objects*, pp. 56-63.

Rusinko, J. and Swan, H. (2012), "Agent-based fabric modeling using differential equations", *CODEE Journal*, Vol. 25 No. 11, pp. 1-12.

Selle, A., Su, J., Irving, G. and Fedkiw, R. (2009), "Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15 No. 2, pp. 339-350.

Shou, Z., Yu, B., Chen, G., Cai, H. and Liu, Q. (2013), "Key designs in implementing online 3D virtual garment try-on system", *Sixth International Symposium on Computational Intelligence and Design, Hangzhou, October 28-29*.

Simnett, T. (2009), "An edge-based approach to adaptively refining a mesh for cloth deformation", *Theory and Practice of Computer Graphics*, pp. 77-84.

Smith, P. (2011), "Flag in the wind: using the fundamentals of cloth simulation to achieve a flag effect", *Thesis portfolio*.

Tang, M., Tong, R., Narain, R., Meng, C. and Manocha, D. (2013), "A GPU-based streaming algorithm for high-resolution cloth simulation", *Computer Graphic Forum*, Vol. 32 No. 7, pp. 21-30.

Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. (1987), "Elastically deformable models", *Computer Graphics*, Vol. 21 No. 4, pp. 205-214.

Thomaszewski, B., Pabst, S. and Strasser, W. (2009), "Continuum-based strain limiting", *Computer Graphics Forum Proceedings of Eurographics*, Vol. 28 No. 2, pp. 569-576.

Vassilev, T. (2001), "Efficient cloth model and collision detection for dressing virtual people", *Proceedings ACM/EG Games Technology Conference*, pp. 1-10.

Vassilev, T. and Rousev, R. (2008), "Algorithm and data structures for implementing a mass-spring deformable model on GPU", *Hayhn Tpyobeha Pycehcehckh Yhbepchtet*, Vol. 47 No. 5, pp. 102-109.

Vassilev, T. and Spanlang, B. (2014), "A mass spring model for real time deformable solids", *Proceedings of East-West Vision, Graz, September 12-13*.

Vassilev, T., Spanlang, B. and Chrysanthou, Y. (2001), "Fast cloth animation on walking avatars", *Computer Graphics Forum*, Vol. 20 No. 3, pp. 260-267.

Vassilev, T.I. (2000), "Dressing virtual people", *SCI'2000 Conference, Orlando, FL, July 23-26*.

Vassilev, T.I. and Spanlang, B. (2000), "Efficient cloth model for dressing animated virtual people", *Visual Computer*, Vol. 17 No. 4, pp. 147-157.

Villard, J. and Borouchaki, H. (2005), "Adaptive meshing for cloth animation", *Engineering with Computers*, Vol. 20 No. 4, pp. 333-341.

Volino, P. and Thalmann, N.M. (1994), "Efficient self collision detection on smoothly discretized surface animations using geometrical shape regularity", *Computer Graphics Forum*, Vol. 13 No. 3, pp. 155-166.

Volino, P. and Thalmann, N.M. (2002), "Avenues of research in dynamic clothing", *Proceedings of the Computer Animation, Geneva, June 19-21*.

Volino, P., Courchesne, M. and Thalmann, N.M. (1995), "Versatile and efficient techniques for simulating cloth and other deformable objects", *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, August 5-8*.

Wang, J., Hu, X. and Zhuang, Y. (2009), "The dynamic cloth simulation performance analysis based on the improved spring-mass model", *International Conference on Wireless Networks and Information Systems, Shanghai, December 28-29*.

Wang, X. and Devarajan, V. (2008), "Improved 2D mass-spring-damper model with unstructured triangular meshes", *Visual Computer*, Vol. 24 No. 1, pp. 57-75.

Weil, J. (1986), "The synthesis of cloth objects", *Computer Graphics*, Vol. 20 No. 4, pp. 359-376.

Wenhsiao, S. and Chen, R.Q. (2006), "A method of drawing cloth patterns with fabric behavior", *Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, April 16-18*.

Yang, J. and Shang, S. (2013), "Cloth modeling simulation based on mass spring model", *Applied Mechanics and Materials*, Vol. 310 No. 2, pp. 676-683.

Yang, Y. and Thalmann, N.M. (1993), "An improved algorithm for collision detection in cloth animation with human body", *Proceedings of Pacific GraphicsWorld Scientific Press, Vol. 1, Singapore*, pp. 237-251.

Ye, J., Webber, R.E. and Wang, Y. (2009), "A reduced unconstrained system for the cloth dynamics solver", *Visual Computer*, Vol. 25 No. 10, pp. 959-971.

Young, M.K., Jeong, H.C., Hwan, G.C. and Do, H.L. (2001), "An efficient animation of wrinkled cloth with approximate implicit integration", *Visual Computer*, Vol. 17 No. 3, pp. 147-157.

Yuan, Z., Ding, Y., Zhang, Y. and Zhao, J. (2010), "Real-time simulation of tissue cutting with CUDA based on GPGPU", *Advanced Materials Research*, Vols 121-122 No. 6, pp. 154-161.

Zhang, D. and Yuen, M. (2000), "Collision detection for clothed human animation", *Proceedings of the Pacific Conference on Computer Graphics and Applications, Hong Kong*, pp. 328-337.

Zhang, D. and Yuen, M. (2001), "Cloth simulation using multilevel meshes", *Computers & Graphics*, Vol. 25 No. 3, pp. 383-389.

Zhang, R., Wang, H., Ma, L. and Meng, N. (2012), "Modeling and simulation process for fabric and skirt based on KES testing", *The 7th International Conference on Computer Science & Education, Melbourne, July 14-17*.

Zhenfang, C. and Bing, H. (2012), "Research of fast cloth simulation based on mass-spring model", *National Conference on Information Technology and Computer Science*, pp. 323-327.

Zhengdong, L. and Shuyuan, S. (2011), "A mass-spring model for real time cloth deformation", *The Proceedings of the International Conference on Multimedia Technology, Hangzhou July 26-28*.

Zhibin, L. and Zhanli, L. (2006), "Dynamic cloth animation in virtual environments", *Tenth International Conference on Information Visualization, London, July 5-7*.

Zhong, Y. and Xu, B. (2009), "Three-dimensional garment dressing simulation", *Textile Research Journal*, Vol. 79 No. 9, pp. 792-803.

Zhong, Y.Q. and Wang, S.Y. (1998), "An efficient way of modeling woven fabric draping", *Computer Graphics*, Vol. 21 No. 3, pp. 205-214.

Zhong, Y.Q. and Wang, S.Y. (2001), "Cloth modeling based on particle system", *Journal of Dong Hua University* (English Edition), Vol. 18 No. 2, pp. 41-44.

Zhou, C., Jin, X., Wang, C.C.L. and Feng, J. (2008), "Plausible cloth animation using dynamic bending model", *Progress in Natural Science*, Vol. 18 No. 7, pp. 879-885.

Zhou, C., Zhu, H., Jin, X. and Feng, J. (2005), "Efficient and simple cloth animation", *Ninth International Conference on Computer Aided Design and Computer Graphics, December 7-10.*

Further reading

Goldsmith, W. (1960), *Impact: The Theory and Physical Behavior of Colliding Solids*, Edward Arnold Publishers Ltd, London.

About the authors

Vajiha Mozafary is a PhD Student in Textile Engineering at the Yazd University, Iran. She has received her Bachelor Degree in Textile Technology Engineering from the Yazd University of Technology, Iran, 2008. She then graduated her Master in the same filed from the Yazd University of Technology, Iran, in 2011. Her publications include two ISI index international journal papers, five national journal papers and three national conference papers.

Dr Pedram Payvandy is currently an Assistant Professor of Textile Engineering at the Yazd University, Iran. He has received his Bachelor Degree in Textile Technology Engineering from the Isfahan University of Technology, Iran, 2001. He then graduated his Master and his PhD in the same field from the Amirkabir University of Technology (Tehran Polytechnic), Iran, in 2003 and 2008, respectively. His research is mostly devoted to the implementation of heuristic optimization algorithms in textile applications. His publications include ten international journal papers, one book chapter paper in *Technical Textile Yarns* book published by Woodhead in 2010, and over 60 international and national conference papers. He has also two national patents in the "Interlacing and Twisting Metallic Filaments by Rotational Magnetic Field Systems." He has been awarded the Rieter award in 2004. Dr Pedram Payvandy is the corresponding author and can be contacted at: peivandi@yazd.ac.ir