University of Iowa

Iowa Research Online

Fall 2013

Cloth simulation by isogeometric analysis

Chao Zheng
University of Iowa

Recommended Citation
Zheng, Chao. "Cloth simulation by isogeometric analysis." PhD (Doctor of Philosophy) thesis, University of Iowa, 2013.
https://doi.org/10.17077/etd.lqbmhc06

CLOTH SIMULATION BY ISOGEOMETRIC ANALYSIS

by

Chao Zheng

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

December 2013

Thesis Supervisor: Professor Jia Lu

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Chao Zheng

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Mechanical Engineering at the December 2013 graduation.

Thesis Committee: _____
                  Jia Lu, Thesis Supervisor

                  _____
                  Kyung K Choi

                  _____
                  Sharif Rahman

                  _____
                  ShaoPing Xiao

                  _____
                  Colby Swan

                  _____
                  Timothy Marler

ACKNOWLEDGEMENTS

ABSTRACT

This thesis aims to develop a design-oriented simulation approach for cloth analysis. Our approach is built on the framework of NURBS-based isogeometric analysis, which utilizes NURBS as the basis functions of analysis. NURBS is a class of parametric geometry to represent curves and surfaces in computer-aided design (CAD) programs. Recently, NURBS geometry has been used directly in analysis. The overall goal of this thesis is to develop a computation infrastructure that enables cloth analysis directly on NURBS geometry.

The advantage of NURBS in the context of cloth modeling lies in the geometric smoothness. Using NURBS, it's easy to construct surfaces with C^1 or higher order of continuity. Compared to C^0 finite element geometry, the NURBS geometry is more effective in capturing wrinkles and folders of cloth, which are characteristics of cloth motion. The NURBS geometry enables the use of rotation-free Kirchhoff-Love shell. The rotation-free shell model not only saves freedoms, but also makes the contact/impact treatment much easier.

The major contribution of this work is the development of a NURBS cloth modeling approach. The mechanical model of cloth and its implementation with NURBS geometry will be presented in detail. Proper constitutive laws are employed for fabric materials. Since NURBS geometry from CAD typically contains multiple patches and trimmed patches, a certain treatment is proposed so that geometry can be used directly in analysis.

Another contribution of this work relates to a contact/impact algorithm. Contact problems in cloth simulation have been a bottleneck of continuum-based approach. Since the general contact method doesn't work well for cloth simulation, a special contact treatment is developed. The present contact model distinguishes three types of contact interactions. The first is the persistent contact force. This force in essence is the traditional penalty force, but is applied when a contact pair is within a separation tolerance instead of being penetrated. This essentially smears the abrupt contact reaction into a relatively smooth force defined only a small thickness. The second is trajectory impact, which deals with the reaction when impact occurs in a time step. The treatment ensures that a point stays on a same side of the surface it impacts on. The third is self-intersection. Intersection resorting force is introduced when the initial configuration has self-intersections, or when the trajectory impact force fails to eliminate all the collisions. We proposed a new method, the method of area minimization, to handle intersections.

The contact models have been integrated into an operator-split integration algorithm. A notable feature of this integration is that the contact/impact response is singled out from the momentum equation.

This work also proposes a continuum-based strain limiting scheme. Because the in-plane stiffness of cloth is much higher than the bending stiffness, numerical difficulty is encountered in either implicit or explicit time integration. The strain limiting is a numerical technique that formulates the in-plane response as a constraint problem to allow the use of lower in-plane stiffness.

A number of examples are presented to show the performance of the proposed approach. In the wrinkling study, the simulated wrinkle pattern looks similar with the experimental results. In the contact study, it is found that the current method can accurately recover a constant contact pressure field (press patch test), can handle contacts of multi-layer folds and produce realistic draping effect. The intersection resolution method is illustrated to be robust to various kinds of intersections. The fast projection method can enlarge time steps while limiting the in-plane strain.

The current method is also applied to the analysis of a soft armor. Beginning from CAD models the armor was put on the human body by a try-on simulation. In multi-layer models, the intersection resolution method is used to resolve the intersections between layers. Subsequently, cloth dynamics are simulated for different human motions. Mechanical indexes such as the extra torque caused by the armors, pressure force on the body, and stress in the armor are predicted. Parametric studies are performed to investigate the change in mechanical metrics under altered design parameters.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

1.1   Introduction to Cloth Simulation

Cloth simulation finds applications in the movie industry, garment design, and electronic commerce. In the movie industry, cloth simulation is essential to producing animated characters. The technology enables the production of visually astonishing movies such as *Ratatouille*, *Kung-Fu Panda*, *Wall-E*, and *Tangled* [34]. Virtual characters such as Gollum, from *Lord of the Rings*, Jar-Jar and Yoda from *Star Wars*, Davy Jones from *Pirates of the Caribbean*, and teddy bear from *Ted*, have become more and more common in mainstream live-action movies [34]. In the garment industry, virtual prototyping is becoming popular [46, 87]. Designers can "sew" cloth patterns together, put them on a virtual model, and watch the try-on effect before they make a real prototype. Successful cloth design software includes Optitex [54], Marvelous Designers [42], etc. In the field of electronic commerce, the idea of using cloth simulation for online virtual fitting rooms has been proposed for many years [23]. Although there isn't online fitting rooms using real-time cloth simulation by far, we expect the technology will become popular in the future.

In spite of the broad applications, cloth simulation had not been widely used until the recent decade. This is mainly because cloth simulation is difficult to realize in finite element method. A piece of cloth tends to buckle under small compression or shear force. In order to simulate the wrinkling and folding processes of cloth, we have

to deal with buckling instability and post-buckling analysis. For an implicit method, one has to solve nearly singular linear systems, which takes longer time to solve. The explicit method does not entail solving linear system, but it requires very small time steps. In most practical applications, cloth simulation is expected to be fast (several minutes or even several seconds). Conventional simulation methods typically don't meet the efficiency requirement.

Another difficulty of cloth simulation is the complicated contact-impact process. The difficulty of contact-impact in cloth simulation manifests on two fronts: a piece of cloth can contact an external body or itself on either side, and it can change the contact sides over time, making it difficult to delineate the contact side using a pre-selected orientation. Also, in the context of the traditional penalty method, it is difficult to select suitable penalty parameters. Given that the lateral stiffness of cloth is very small, it is difficult to maintain numerical stability because a small lateral force may result in an excessive perturbation. Delineating complicated contact also requires to delicate contact detecting algorithms which also present challenge to computation.

Another challenge of cloth simulation is how to effectively enforce realistic strain on cloth. The bending stiffness of fabric material is far lower than in-plane stiffness, and consequently, the in-plane strain in most cases is negligibly small, invisible to the naked eye. Mathematically, this gives rise to stiff differential equations. However, using high physical in-plane stiffness brings in several numerical issues. For explicit methods, the time step is determined by the highest stiffness in the system;

a higher in-plane stiffness would force a small time step which is not necessary for simulating the flexure response. The implicit method of Barraf [3] may allow large time steps, but this method brings in fictional damp, which is proportional to in-plane stiffness. Thus, in early simulations the physical in-plane stiffness was replaced by artificial low values. The main problem of lowering the in-plane stiffness is that the cloth may stretch excessively. How to use lower stiffness while eliminating overstretching efficiently is an important research topic.

Many novel ideas have been proposed to address the issues in cloth simulation. Below, we will briefly summarize the major advances in the following subject areas: (1) models of cloth; (2) methods of time integration; (3) contact-impact algorithm; (4) strain limiting.

1.1.1   Models of Cloth

Models of cloth can be divided into two categories: the spring-based model and the continuum-based model [88]. Spring-based model treats a piece of cloth as a spring network connecting a set of mass points. Breen et al. [12, 13, 39] probably were the first to simulate cloth by a spring-based model. Their original motivation was to obtain better approximation to the testing data of wool, therefore their energy function was a set of highly nonlinear springs. Provot [68] used a simpler linear spring system, and his stiffness of spring was much lower than the experimental one. This model was widely used in the graphics field because it is easy to implement, and it provides fast simulation with reasonable results. Choi and Ko [19] proposed an

immediate buckling spring model, which solved the instability problem of implicit methods. Bridson et al. [16] proposed an explicit model of bending spring, which gives better simulation results. Baraff et al. [3] and Grinspun et al. [36] introduced a discrete shell model, which considers the in-plane deformation in a continuum manner and the bending deformation in spring manner. A disadvantage of spring-based model is that it is difficult to accurately reproduce the mechanical behavior of cloth. Studies [12, 13] have attempted to determine spring parameters based on the stress-strain and curvature-moment curves obtained from mechanical tests [45]. However, parameters so obtained are mesh dependent and thus not easily transferable across networks of different topologies.

Another family of models falls into the continuum approach, which treats the cloth as continuum and uses the continuum mechanics law to describe the deformation. This approach has many advantages over the spring-based model. For example, the material properties of the continuum approach is independent of the topology of the grid. Another feature [43] of the continuum approach is that they can leverage the existing research on material and geometric nonlinearity. Many continuum cloth models have been proposed. Collier et al. [22] developed a shell element with finite rotation and small strain to model the draping process of cotton and achieved good agreement with experimental results. Gan et al. [32], Chen et al. [17], and Eischen et al. [28] utilized a degenerated solid shell to simulate fabric draping. Man and Swan et al. [60, 58, 59] used the continuum degenerated shell model to simulate the soft armors. Kim [50] applied Simo's geometrically exact shell [73, 74, 75] to model cloth.

In addition to shell theories, Ascough et al. [2] modeled the cloth as a network of beams, and Teng et al. [78] used the finite-volume method to simulate cloth. These studies showed that, despite the coarse microstructure of fabric materials, continuum theories can effectively capture the characteristics of cloth deformation. However, the continuum simulations has not gained popularity, possibly due to the following reasons: (1) they need more degrees of freedoms; (2) computation efficiency has been low, lower than spring-based model ;(3) the contact/impact treatment has been a bottle-neck [43].

1.1.2 Time Integration

Two integration schemes have dominated cloth simulation. One is called the semi-implicit scheme, which was first introduced to the cloth simulation community by Baraff and Witkin [3]. The semi-implicit scheme is based on the backward Euler method, but in solving the nonlinear equivalent equation with Newton-Raphson method, the iteration is halted after the first iterative step. The semi-implicit scheme is efficient because full Newton-Raphson iteration is avoided while large time step is used. The drawback of Baraff's scheme is that unrealistic damping significantly degrades the realism. The additional damping stems from two sources. The first is the numerical error caused by the lack of nonlinear iteration. For instance, when a spring rigidly rotates an angle, the stress should be zero. But in the semi-implicit, the stress predicted by the first time of iteration is nozero, and this prediction will reduce the rotation velocity. The second additional damping is the artificially big damping force

introduced to avoid post-buckling instability. Baraff assumed the internal damping force is in the same direction as the internal force, and its magnitude is proportional to the strain rate. A high flexure damping coefficient helps to maintain the numeral stability in cloth buckling. Choi and Ko [19] proposed an immediate buckling spring model, which eliminates artificial damping. Choi and Ko [19] also introduced a 2nd order backward Euler method to reduce the numerical damping. Oh et al. [64] modified the standard Euler method in order to reduce numerical damping.

Another popular time integration scheme is the Velocity Verlet method. The Velocity Verlet method was first applied to cloth simulation by Bridson et al. [16]. Velocity Verlet method is an implicit/explicit scheme. Its main idea is to solve the velocity implicitly while updating the displacement term explicitly. The time increment of Velocity Verlet method can be larger than that of pure explicit method. When the damping coefficient matrix is simplified to a diagonal matrix, there is no need to solve linear systems.

1.1.3   Contact/Impact Scheme

The treatment of contact/impact has been a bottleneck in cloth simulation. According to how penetration is dealt with, there has been two main types of contact/impact algorithms: the coupled approach and the decoupled approach. The coupled approach considers the penetration conditions as constraints of cloth movement [3, 34] and couples the contact condition with the solution of the dynamic motion. This naturally calls for an implicit setting because the configuration at the

end of time step is yet to be found. Thus an iteration of trial-and-test is necessary to ensure penetration-free at the end of a time step [34]. The decoupled approach [15] solves the momentum equation first without considering contact; if contact occurs, the non-penetration condition will be enforced by applying the impulse explicitly.

The contact/impact response has been described mainly in two different ways: the proximity model and the trajectory collision model (also known as continuous contact model). The proximity model introduces a contact force when two pieces of cloth in current configuration are very close [62, 3, 19]. This model is efficient, but penetration could be missed when two points impact at high speed. The trajectory collision model [62, 86] introduces an impact force when the trajectories of two primitives impact during the current time step. Some researchers switched between these two models [76]. Bridson [15] proposed a two-phase scheme to take advantage of both models. The persistent contact model is applied first, which is expected to stop most trajectory intersections. The impact model is then applied, resolving the remaining trajectory penetrations. This sequential treatment fits naturally into an operator-split integration framework. This scheme was shown to be efficient and robust for complex problems.

In Bridson's scheme, the combination of proximity and trajectory collision models alone is still not sufficient to eliminate all penetrations - new penetrations could be generated numerically. Bridson introduced a third filter: the rigid impact zone (RIZ). In all the locations where the trajectory collision fails, the connected nodes are considered as a RIZ, and all the nodes inside a RIZ are imposed to be

a rigid body. A drawback of RIZ approach is that it also stops tangential sliding. Harmon [37] proposed an implicit way to treat the missed collisions. Harmon consider each penetration as an equation, and all the penetrations form a linear system. By solving this linear system, which may be costly, all the penetrations could be resolved. However, these history-dependent contact models will fail after the non-simulated objects impose the cloth intersects [4].

In addition, both Bridson's and Harmon's models are unable to resolve initial intersections. In simulating multi-layer garments, different layers of garments may be designed and modeled in different spaces, and thus initial penetrations are frequently encountered. As alluded above, for cloth simulation it is infeasible to use a predefined surface orientation to delineate contact side. The approach by Bridson and others essentially use the geometries at the current time step to enforce contact condition: a contact point should stay on a same side of a contact surface. A pitfall of this approach is that, one a point ends on the wrong side, the algorithm will try to keep it there. Baraff [4] firstly proposed a history-free contact scheme, which allows intersections to occur, and then resolves them either kinematically or kinetically. Joining the element-element intersection segments, the two regions inside the same intersection contour form a contact pair. Baraff's scheme only works when both sides of intersection contours are closed. Wicke [92] later extended baraff's scheme to arbitrary intersection types with the aid of 2D parametric domain.

Another way to resolve arbitrary kinds of intersections is the contour length minimization method proposed by [89]. This methods considers the gradient of in-

tersection contour length as the direction of resorting force, and thus it works for both closed or unclosed contours. The contour minimization method was believed to be more robust because an edge-polygon intersection missing from collision detection will not break the algorithm. The major problem of Volino's scheme is that it breaks down when contour length is a local minimum or a reduction of the contour length does not necessarily mean a reduction in the intersection area. A simple example of local minimization is given when one rectangles inserts into another one.

This thesis proposes a method of intersection resolution that minimize the intersection area. Minimizing the intersection area will leads to the elimination of intersections, and local minimization status is nonexist.

1.1.4 Strain Limiting

Strain limiting means artificially reducing the in-plane stiffness while imposing certain limits on the allowable strain range. In other word, the in-plane deformation is formulated as a constraint problem. Provot [68] proposed an explicit method of strain limiting that restores the overstretched springs by adjusting the particle position directly. Because adjusting position of one spring may result in overstretching another spring, an iteration was required. Both Jacobi iteration and Gauss-Seidel iteration [15] were utilized, but neither can guarantee convergence.

The implicit method of strain limiting was also proposed [39, 38]. These approaches were mainly based on augmented Lagrange method. Goldenthal et al.[35] proposed a fast projection scheme, which solve the constraint equation more effec-

tively. The implicit method requires solving a linear system for each iteration, but it can converge very fast.

The fast projection method used by Goldenthal et al. [35] constrained the spring length in a weft- and warp- aligned quadrilateral grid. The method was limited to this special grid configuration. To obtain a grid-independent strain limiting scheme, one has to resort to the continuum models. Study on continuum-based strain limiting has been rare. Thomaszewski et al. [84] proposed an explicit strain limiting for triangular elements. This report appears to be the only work on continuum strain limiting.

1.1.5 Other Related Work

There are many other important studies related to the cloth simulation, though they are not the focus of this thesis. Oh et al. [65, 53] proposed a physically faithful multigrid method, which could largely improve the simulation efficiency. Parallel computing of cloth simulation was developed by Thomaszewski and Blochinger [83], Selle et al. [71], and Kim et al. [48]. Bhat et al. [11] suggested to use videos to evaluate the material parameters. Wang et al. [91] developed an example-based wrinkle synthesis method to generate fine-scale details with coarse mesh. Kaldor et al. [44] simulated knitted cloth in yarn level.

It is worth to note that the cloth modeling is a branch of physically-based animation in computer graphics. Since the Lasseter's discussion [52] and Terzopoulos et. al's seminal paper [80], significant contributions were made in simulating various

objects, such as the simulation of rods [10], hairs [70], skins [61], fracture [77], fluid [14] and so on. A detailed survey can be found in [33] and [63].

1.2 Introduction to Isogeometric Analysis

Isogeometric analysis is an advance of finite element; the essential idea is to use the same basis in analysis as used in computer-aided design (CAD). directly in analysis[40]. The aim of isogeometric analysis is to bridge the gap between CAD programs and analysis, hence reducing the cost of FEM model generation. Hughes et al. [40] were the first to propose the concept and framework of isogeometric analysis. Cottrell et al. and Bazilevs et al. [7] investigated the approximation property, stability, and error estimates for h-refinement. Cottrell et al. [24] applied isogeometric analysis to structural vibrations and gave an accuracy study later [25]. Elguedj et al. [29] proposed B-bar and F-bar projection methods for nearly incompressible material in small or large deformation. An efficient quadrature scheme was proposed by Hughes et al.[41]. Bazilevs [6] et al. used T-spline for analysis, which allows for locally adaptive refinement. Kim proposed [49] an analysis scheme for trimmed-NURBS, a more complex form of NURBS surfaces. Isogeometric analysis has been applied to biomechanics, e.g., a patient-specific aneuysm model was introduced by Bazilevs et al.[8]. A NURBS muscle model was introduced by Zhou and Lu [95]. Lu [55] and Temizer et al.[27] proposed a contact scheme for NURBS-based isogeometric analysis, and the accuracy of contact pressure was largely improved by the higher-order geometry. NURBS-based shell elements were introduced by Kiendl et al. and

Benson et al., for both Kirchhoff-Love shell [47] and Reissner-Mindlin shell [9]. A NURBS-based continuum rod element was proposed by Lu [56, 57], which proved to be much more efficient then the corresponding FE model. Isogeometric analysis was also applied to topology optimization by Wall et al.[90], cho el al.[18] and Seo et al. [72], and it was found that the shape sensitivity was enhanced.

1.3   Goals of Present Work

The goal of this dissertation is to develop a continuum-based cloth model with high simulation efficiency and high ability of treating contact. The software derived from this study should primarily be a design tool that aids garment or armor design, but could also be used in animation. Specifically, we will pursue the following research tasks:

(1) developing a NURBS-based isogeometric shell model and employing appropriate constitutive models for fabric material. The model should allow for trimmed NURBS and multi-patch NURBS that are commonly used in garment design.

(2) developing a robust and efficient contact-impact algorithm capable of dealing with complicated contact conditions including two-sided contact and self-contact.

(3) developing a continuum-based strain limiting algorithm to allow for large time step while keeping in-plane strain small;

(4) testing and verifying the formulation using benchmark examples and garment-level simulation.

(5) establishing a protocol that takes NURBS armor designs and pipelines

them to analysis with minimal user interference.

1.4 Organization of the Thesis

The outline of this paper is as follows. Chapter 2 introduces the isogeometric model of cloth. Chapter 3 introduces the time integration and the contact model. A new area minimization method for intersection resolution is proposed in Chapter 4, and a continuum-based fast projection method is proposed in Chapter 5. In chapter 6, the current simulation method is applied to the design of soft armor. Chapter 7 presents some conclusions and limitations of the current study.

CHAPTER 2
ISOGEOMETRIC ANALYSIS OF CLOTH

We focus on *thin cloth* that can be effectively described by Kirchhoff-Love shell theory. By thinness, one typically means that the thickness h is less than 1/20 of the characteristic length in the lateral direction. The Kirchhoff-Love theory assumes that the transverse shear is negligible and thus the shell kinetics is described by in-plane stretching and lateral bending. In analysis, the Kirchhoff-Love element requires C^1 continuity between elements, which is difficult to achieve in Lagrangian- or Hermite-based finite element interpolations. NURBS geometry maintains a C^1 or higher order continuities by construction, and thus naturally supporting Kirchhoff-Love analysis [47]. One of the advantages of NURBS Kirchhoff-Love element, in comparison to the Reissner-Mindlin formulation in finite element geometry, e.g. [73, 74], is that only the surface displacement is involved; no rotational degree-of-freedoms (DOFs) are needed. Another attraction of the NURBS geometry, which possesses global smoothness, is more suited for describing wrinkles or folds, which are the characteristics of cloth motion. We expect that NURBS can better capture smooth motions and do so with fewer DOFs compared to finite element method.

In a sense, the use of smooth geometry in cloth simulation is not novel. Thomaszewski et al. [82] proposed the use of subdivision shells [21, 20]. B-splines and NURBS have been used in physical modeling of deformable bodies in the Computer Graphics and CAD communities [81, 79, 69], although they may not be specifically for fabrics.

In the practical application of isogeometric analysis, several kinds of special geometries may be encountered. The first is the multi-patch model, in which one model contains more than one NURBS patches. The relation of control points to impose C^1 continuity across patches is derived in this chapter. The second is the trimmed NURBS. A novel scheme to handle the trimmed NURBS is also developed in this chapter.

2.1   NURBS Geometry

NURBS is a parametric geometry used to represent curves and surfaces in CAD. A NURBS curve is defined by a knot vector $\boldsymbol{\xi}$, a set of control points $\{\mathbf{Q}_i\}$ and their weights $\{w_i\}$, and a degree p. Complete coverage is provided by Piegl and Tiller [26, 5, 67, 31].

$$\mathcal{C}(\xi) = \frac{\sum_{i=1}^{n} B_{i,p}(\xi) w_i \mathbf{Q}_i}{\sum_{i=1}^{n} B_{i,p}(\xi) w_i} \tag{2.1}$$

or simply $\mathcal{C}(\xi) = \sum_{i=1}^{n} R_i(\xi) \mathbf{Q}_i$ with $R_i = \frac{B_{i,p}(\xi) w_i}{\sum_{j=1}^{n} B_{j,p}(\xi) w_i}$. In Eq. (2.1), $\{B_{i,p}\}$ are the B-spline basis functions, which are completely defined by the knot vector and the degree. The knot vector is a non-decreasing sequences of coordinates in the parametric space, denoted as $\boldsymbol{\xi} = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$. The B-spline basis functions follow a recursive relation

$$B_{i,p} = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1} + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1} \tag{2.2}$$

where for zero degree,

$$B_{i,0} = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The knot vector specifies the divisions of the curve in the parametric space. Each non-empty segment $[\xi_1, \xi_{i+1}]$ defines a Bezier segment. The interior knots can repeat upto p times; with each repetition the degree of continuity at that knot reduces by one. The first and last knots can repeat $p + 1$ times, in which case the B-spline is interpolatory at the two ends. B-splines and NURBS enjoy many properties desirable for geometric description and analysis. For a complete coverage, see [67].

A NURBS surface is constructed by taking the tensor-product of two NURBS curves, giving the parametric form

$$\mathcal{S}(\xi^1, \xi^2) = \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B_{i,p}(\xi^1) B_{j,q}(\xi^2) w_{ij} \mathbf{Q}_{ij}}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B_{i,p}(\xi^1) B_{j,q}(\xi^2) w_{ij}} \tag{2.4}$$

The control points $\mathbf{Q}_{ij}$ form a $n_1 \times n_2$ net. The rectangle $[\xi_1^1, \xi_{n_1+p+}^1] \times [\xi_1^2, \xi_{n_2+q+1}^1]$ defines the parametric domain of the NURBS surface, where each non-empty subdomain $[\xi_i^1, \xi_{i+1}^1] \times [\xi_j^2, \xi_{j+1}^1]$ specifies a Bezier element. Eq. (2.4) can be expressed in the following form:

$$\mathcal{S}(\xi^1, \xi^2) = \sum_{I=1}^{n} N_I(\xi^1, \xi^2) \mathbf{Q}_I, \quad n = n_1 \times n_2 \tag{2.5}$$

where the controls points and their interpolation functions are stored in vector form.

2.2   NURBS Kirchhoff-Love Shell

We focus on *thin cloth* that can be effectively described by Kirchhoff-Love shell theory. By thinness, one typically means that the thickness h is less than $1/20$ of the characteristic length in the lateral direction. The current model can be applied to most common fabric and soft armors, but it doesn't work for very thick cloth or armors.

The Kirchhoff-Love theory assumes that the transverse shear is negligible and thus the shell kinetics is described by in-plane stretching and lateral bending. In analysis, the Kirchhoff-Love element requires C^1 continuity between elements, which is difficult to achieve in Lagrangian- or Hermite-based finite element interpolations. NURBS geometry maintains C^1 or higher order continuities by construction, thus naturally supporting Kirchhoff-Love analysis [47]. The advantage of NURBS Kirchhoff-Love element, in comparison to the Reissner-Mindlin formulation in finite element geometry, e.g. [73, 74], is that only the surface displacement is involved; no rotational DOFs are needed.

2.2.1   Kinematics

The NURBS formulation below follows Kiendl et. al. [47]. We use the same set of NURBS basis functions to parameterize the reference and current configurations of the cloth surface:

$$
\begin{aligned}
\mathbf{X} &= \sum_{I=1}^{n} N_I(\xi^1, \xi^2)\mathbf{Q}_I \\
\mathbf{x} &= \sum_{I=1}^{n} N_I(\xi^1, \xi^2)\mathbf{q}_I
\end{aligned}
\tag{2.6}
$$

Here $\mathbf{Q}_I$'s are the reference coordinates of control points, and $\mathbf{q}_I$'s the current coordinates. The knot parameters $\boldsymbol{\xi} = (\xi^1, \xi^2)$ serves as the convected coordinates whereby a fixed pair $\boldsymbol{\xi}$ represents a same material point throughout the deformation. These two coordinates induce two convected surface basis vectors $\mathbf{a}_1 = \mathbf{x}_{,\xi^1}$, $\mathbf{a}_2 = \mathbf{x}_{,\xi^2}$ spanning the tangent plane at every point on the surface. A line element in the current configuration is thus represented as $d\mathbf{x} = \mathbf{a}_1 d\xi^1 + \mathbf{a}_2 d\xi^2$. In the reference configu-

ration, the surface bases are denoted by $\{\mathbf{A}_1, \mathbf{A}_2\}$, and a line element is given by

$d\mathbf{X} = \mathbf{A}_1 d\xi^1 + \mathbf{A}_2 d\xi^2$. The basis vectors are illustrated in Fig. 2.1.
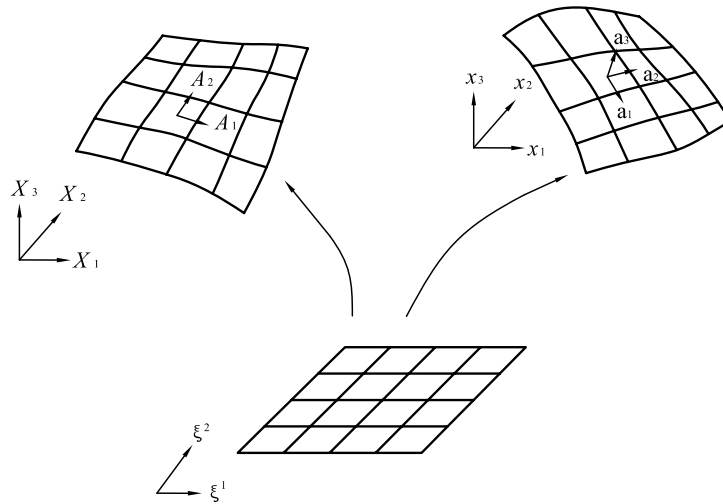
Figure 2.1: Illustration of shell kinematics

The unit normal $\mathbf{a}_3$ (in the current configuration) is

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\parallel \mathbf{a}_1 \times \mathbf{a}_2 \parallel} \tag{2.7}$$

With respect to the convected basis vectors, the surface deformation tensor $\mathbf{C}$ and

the Green-Lagrangian strain $\mathbf{E}$ take the form

$$C_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta, \quad \alpha, \beta = 1, 2$$

$$E_{\alpha\beta} = \frac{1}{2}(\mathbf{a}_\alpha \cdot \mathbf{a}_\beta - \mathbf{A}_\alpha \cdot \mathbf{A}_\beta) \quad \alpha, \beta = 1, 2 \tag{2.8}$$

The surface curvature tensor $\boldsymbol{\kappa}$ is defined by

$$\kappa_{\alpha\beta} = -\frac{\partial \mathbf{a}_3}{\partial \xi^\alpha} \cdot \mathbf{a}_\beta = \mathbf{a}_3 \cdot \frac{\partial^2 \mathbf{x}}{\partial \xi^\alpha \partial \xi^\beta}, \quad \alpha, \beta = 1, 2 \tag{2.9}$$

It is convenient to use a local orthonormal basis to perform the element computations presented later. To this end we introduce a pair of orthonormal vectors $\{\bar{\mathbf{E}}_1, \bar{\mathbf{E}}_2\}$ point-wise in the tangent plane spanned by $\{\mathbf{A}_1, \mathbf{A}_2\}$. With respect to the new bases, we express a line element as $d\mathbf{X} = \bar{\mathbf{E}}_1 d\bar{X}_1 + \bar{\mathbf{E}}_2 d\bar{X}_2$. Locally, $(d\bar{X}_1, d\bar{X}_2)$ are related to $(d\xi^1, d\xi^2)$ via

$$\begin{bmatrix} d\bar{X}_1 \\ d\bar{X}_2 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{E}}_1 \cdot \mathbf{A}_1 & \bar{\mathbf{E}}_1 \cdot \mathbf{A}_2 \\ \bar{\mathbf{E}}_2 \cdot \mathbf{A}_1 & \bar{\mathbf{E}}_2 \cdot \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} d\xi^1 \\ d\xi^2 \end{bmatrix} := \mathbf{J} d\boldsymbol{\xi} \qquad (2.10)$$

Derivatives of a basis function N with respect to physical coordinates follow the chain rule:

$$\begin{bmatrix} N_{,\bar{X}_1} \\ N_{,\bar{X}_2} \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} N_{,\xi^1} \\ N_{,\xi^2} \end{bmatrix} \qquad (2.11)$$

In the current configuration, the basis $(\bar{\mathbf{a}}_2, \bar{\mathbf{a}}_2)$ convected from the physical basis $\{\bar{\mathbf{E}}_1, \bar{\mathbf{E}}_2\}$ are

$$\bar{\mathbf{a}}_\alpha := \frac{\partial \mathbf{x}}{\partial \bar{X}_1} = \sum_I N_{I,\bar{X}^\alpha} \mathbf{q}_I, \qquad \alpha = 1, 2 \qquad (2.12)$$

With respect to the physical basis, the Green-Lagrangian strain assumes the form $\bar{E}_{\alpha\beta} = \frac{1}{2}(\bar{\mathbf{a}}_\alpha \cdot \bar{\mathbf{a}}_\beta - \delta_{\alpha\beta})$. The physical components of the curvature tensor can be obtained by the transformation $\bar{\kappa}_{\alpha\beta} = \frac{\partial \xi^\delta}{\partial \bar{X}_\alpha} \kappa_{\delta\gamma} \frac{\partial \xi^\gamma}{\partial \bar{X}_\beta}$.

The local bases $\{\bar{\mathbf{E}}_i\}$ can be selected in many ways, but we recommend aligning $\{\bar{\mathbf{E}}_1\}$ and $\{\bar{\mathbf{E}}_2\}$ to the weft and warp directions, respectively.

2.2.2 Constitutive Law

Cloth response is typically inelastic, exhibiting anisotropic properties and a small to moderate amount of hysteresis [66, 51, 1]. Since large rotation is involved, the use of finite strain is necessary. Because in-plane strain of fabrics is usually small

($< 2\%$), we assume a linear anisotropic relation between the (in-plane) Piola-Kirchhoff stress $\mathbf{S}$ and the Green-Lagrangian strain $\mathbf{E}$:

$$\mathbf{S} = \mathbb{D}\mathbf{E}, \qquad \text{In Voigt form} \qquad \mathbb{D} = \begin{bmatrix} d_1 & d_3 & 0 \\ d_3 & d_2 & 0 \\ 0 & 0 & d_4 \end{bmatrix} \qquad (2.13)$$

An early objection to shell models of fabric materials was that they produce rubber-like behaviors. In part, this is because the consistent bending equation ($\mathbf{m} = \frac{h^3}{12}\mathbb{D}\boldsymbol{\kappa}$) over-estimates the moment. Among other reasons, the actual moment of inertial of fabric sheets should be smaller than the nominal value $\frac{h^3}{12}$. The issue can be resolved by abandoning the consistent bending and using an independent bending function instead. As a step in this direction, we assume the bending momentum and curvature in the weft and warp directions have the following relation [1] and ignore the energy of bending shear.

$$m_I = \begin{cases} \text{sign}(\kappa_I)2B_0\sqrt{\kappa_0|\kappa_I|}, & |\kappa_I| \leq \kappa_0 & (2.14) \\[2mm] \text{sign}(\kappa_I)(B_0|\kappa_I| + B_0\kappa_0), & |\kappa_I| > \kappa_0 & (2.14') \end{cases}$$

in which I can be replaced by 'weft' or 'warp' and B_0, κ_0 are material constants. The bending curve is depicted in Figure 2.2. The shape mimics the ascending portion of a typical fabric bending in Kawabata test [45]. The bending moment in general depends on curvature change; here, the reference curvature is taken to be zero.

2.2.3 Element Equation

External forces acting on a piece of cloth normally include a body force $\rho\mathbf{b}$; damping force $\mathbf{f}_d$ (per unit surface area) which is in general a function of the cloth velocity; and traction forces $\bar{t}$ prescribed on the boundary edge. Following the textile
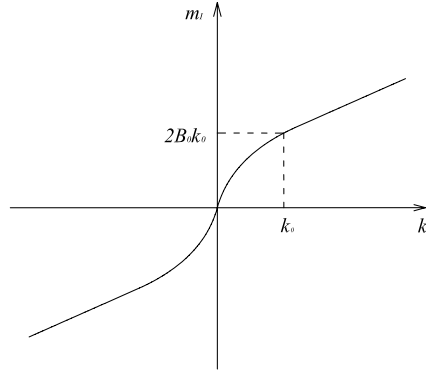
Figure 2.2: Moment-curvature curve in principal space

community convention, we use the surface density ρ to describe the mass distribution; $\rho = \rho_0 h$ where ρ_0 is the 3D density and h is the cloth thickness. The weak form of dynamics equilibrium equation is given by

$$\int_\Omega \rho \delta \mathbf{u}^T \mathbf{a} dA + \int_\Omega (h \delta \mathbf{E}^T \mathbf{S} + \delta \boldsymbol{\kappa}^T \mathbf{m}) dA = \int_\Omega (\delta \mathbf{u}^T (\rho \mathbf{b} + \mathbf{f}_d) dA + \int_{\Gamma_t} (\delta \mathbf{u}^T \bar{\mathbf{t}}) h ds \quad (2.15)$$

In NURBS representation, $\mathbf{u} = \mathbf{x} - \mathbf{X} = \sum_I N_I (\mathbf{q}_I - \mathbf{Q}_I)$, and thus

$$\delta \mathbf{u} = \mathbf{N} \delta \mathbf{q}, \qquad \mathbf{N} = [N_1 \mathbf{I}\ N_2 \mathbf{I} \cdots] \quad (2.16)$$

where $\mathbf{I}$ is the 3×3 identity matrix. The variation of Green-Lagrangian strain, in Voigt form $\delta \mathbf{E} = (\delta E_{11},\ \delta E_{22},\ 2\delta E_{12})$, is derived to be

$$\delta \mathbf{E} = \mathbf{B}_m \delta \mathbf{q}, \qquad \mathbf{B}_m = \begin{pmatrix} \mathbf{B}_{11}^{m1} & \mathbf{B}_{11}^{m2} & ... \\ \mathbf{B}_{22}^{m1} & \mathbf{B}_{22}^{m2} & ... \\ 2\mathbf{B}_{12}^{m1} & 2\mathbf{B}_{12}^{m2} & ... \end{pmatrix} \quad (2.17)$$

where

$$(\mathbf{B}_{\alpha\beta}^{mI})^T = \frac{\partial E_{\alpha\beta}}{\partial \mathbf{q}_I} = \frac{1}{2}(N_{I,\beta} \mathbf{a}_\alpha + N_{I,\alpha} \mathbf{a}_\beta) \quad (2.18)$$

Similarly, from the definition of curvature we can derive

$$\delta \boldsymbol{\kappa} = \mathbf{B}_b \delta \mathbf{q}, \qquad \mathbf{B}_b = \begin{pmatrix} \mathbf{B}_{11}^{b1} & \mathbf{B}_{11}^{b2} & ... \\ \mathbf{B}_{22}^{b1} & \mathbf{B}_{22}^{b2} & ... \\ 2\mathbf{B}_{12}^{b1} & 2\mathbf{B}_{12}^{b2} & ... \end{pmatrix} \quad (2.19)$$

where

$$(\mathbf{B}_{\alpha\beta}^{bI})^T = \frac{\partial \kappa_{\alpha\beta}}{\partial \mathbf{q}_I} = N_{I,\alpha\beta}\mathbf{a}_3 + \frac{1}{j}\left[N_{I,1}(\mathbf{a}_2 \times \mathbf{x}_{,\alpha\beta}) + N_{I,2}(\mathbf{x}_{,\alpha\beta} \times \mathbf{a}_1)\right] - \tag{2.20}$$
$$(\mathbf{x}_{,\alpha\beta} \cdot \mathbf{a}_3)(N_{I,1}\mathbf{a}^1 + N_{I,2}\mathbf{a}^2)$$

In the above, $j = ||\mathbf{a}_1 \times \mathbf{a}_2||$ is the area stretch, and $(\mathbf{a}^1, \mathbf{a}^2)$ are the bases dual to $(\mathbf{a}_1, \mathbf{a}_2)$, satisfying $\mathbf{a}^\alpha \cdot \mathbf{a}_\beta = \delta_\beta^\alpha$ and $\mathbf{a}^\alpha \cdot \mathbf{a}^3 = 0$.

Substituting Eq. (2.16), Eq. (2.17) and Eq. (2.19) into Eq. (2.15) yields the discrete dynamic equation

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{f}_{damp} + \mathbf{f}_{int} = \mathbf{f}_{ext} \tag{2.21}$$

where

$$\begin{aligned}
\mathbf{M} &= \int_\Omega \mathbf{N}^T\mathbf{N}\rho \, dA \\
\mathbf{f}_{int} &= \int_\Omega \left(h\mathbf{B}_m^T\mathbf{S} + \mathbf{B}_b^T\mathbf{m}\right) dA \\
\mathbf{f}_{ext} &= \int_\Omega \mathbf{N}^T\rho\mathbf{b} \, dA + \int_{\Gamma_t} \mathbf{N}^T\bar{\mathbf{t}}h \, ds \\
\mathbf{f}_{damp} &= \int_\Omega \mathbf{N}^T\mathbf{f}_d \, dA
\end{aligned} \tag{2.22}$$

For low speed air drag we assume that $\mathbf{f}_d = -\eta\dot{\mathbf{x}}$ where η is a viscosity constant; in this case, $\mathbf{f}_{damp} = -\frac{\eta}{\rho}\mathbf{M}\dot{\mathbf{q}} := -\mathbf{D}\dot{\mathbf{q}}$.

For numerical integration, we use Gauss quadrature, and each quadrature domain is a cell from the knot grid of a NURBS patch.

2.3   Multi-patch Model

A tensor product NURBS surface is topologically rectangular, having four edges. To describe complicated shapes, two methods are commonly used: trimmed
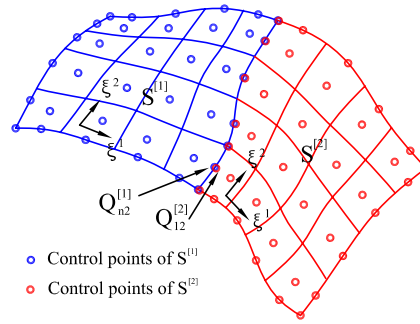
Figure 2.3: Illustration of G^1 multi-patch model

NURBS and multi-patch NURBS. Treatment of trimmed NURBS in isogeometric analysis has been discussed in some recent papers [49]. We will present trimmed NURBS in shell analysis in the next section. Here we focus on multi-patch NURBS and discuss the enforcement of C^1 continuity across patch boundaries. By a patch, we refer to a NURBS unit defined by a tensor product 2D knot domain and the corresponding control points. A multi-patch model by definition consists of multiple patches that match along patch boundaries. To use the Kirchhoff-Love element, we must ensure C^1 continuity in physical derivative across the common edge of adjacent patches. Here, we show that physical C^1 continuity can be ensured by satisfying G^1-continuous, a weaker continuity condition [30].

With reference to Figure 2.3, consider two NURBS patches $\mathcal{S}^{[1]}$ and $\mathcal{S}^{[2]}$ that connect on a common edge. We restrict the discussion to cases where the two patches have the same NURBS parameterization in ξ^2 direction (the edge direction) and matching control points on the edge, that is

$$\mathbf{Q}_{nj}^{[1]} = \mathbf{Q}_{1j}^{[2]}, \qquad j = 1, 2, ... n_2 \tag{2.23}$$

Knots in ξ^1 are specified individually in each patch, with $\xi^1 = b$ being the end knot of the left patch and $\xi^1 = a$ the beginning knot of the right patch. We assume that degrees in the ξ^1 domain are the same in the two patches, and knots a and b repeat exactly $(p+1)$-times so that the end-point interpolation property holds. In this case, the NURBS basis functions from both sides match on the common edge:

$$N_{nj}^{[1]}(b, \xi^2) = N_{1j}^{[2]}(a, \xi^2) := R_j(\xi^2), \quad j = 1, 2, ...n_2 \tag{2.24}$$

and that the common edge is determined by either the last column of control points in the patch $S^{[1]}$, or the first column of control points of $S^{[2]}$:

$$\mathcal{S}^{[1]}(b, \xi^2) = \sum_{j=1}^{n_2} R_j(\xi^2)\mathbf{Q}_{nj}^{[1]}, \quad \mathcal{S}^{[2]}(a, \xi^2) = \sum_{j=1}^{n_2} R_j(\xi^2)\mathbf{Q}_{0j}^{[2]} \tag{2.25}$$

Since the control points match, we conclude that

$$\mathcal{S}^{[2]}(a, \xi^2) = \mathcal{S}^{[1]}(b, \xi^2), \qquad \frac{\partial \mathcal{S}^{[2]}(a, \xi^2)}{\partial \xi^2} = \frac{\partial \mathcal{S}^{[1]}(b, \xi^2)}{\partial \xi^2} \tag{2.26}$$

On the common edge, the derivative with respect to ξ^1, in $S^{[1]}$, depends on the last two columns of control points

$$\frac{\partial \mathcal{S}^{[1]}(b, \xi^2)}{\partial \xi^1} = \frac{p}{b - \xi_n} \frac{w_{n-1}}{w_n} \sum_{j=1}^{n_2} R_j(\xi^2)(\mathbf{Q}_{nj}^{[1]} - \mathbf{Q}_{(n-1)j}^{[1]}) \tag{2.27}$$

whereas in $S^{[2]}$,

$$\frac{\partial \mathcal{S}^{[2]}(a, \xi^2)}{\partial \xi^1} = \frac{p}{\bar{\xi}_{p+1} - a} \frac{w_2}{w_1} \sum_{j=1}^{n_2} R_j(\xi^2)(\mathbf{Q}_{2j}^{[2]} - \mathbf{Q}_{1j}^{[2]}) \tag{2.28}$$

G^1 continuity in ξ^1 requires that [30]

$$\frac{\partial \mathcal{S}^{[2]}(a, \xi^2)}{\partial \xi^1} = c\frac{\partial \mathcal{S}^{[1]}(b, \xi^2)}{\partial \xi^1} \tag{2.29}$$

where c is a non-zero constant. This condition states that the tangents in ξ^1-direction at two sides of the edge are parallel, but not necessarily the same magnitude. The condition (2.29) is satisfied if

$$\mathbf{Q}_{2j}^{[2]} - \mathbf{Q}_{1j}^{[2]} = c'(\mathbf{Q}_{nj}^{[1]} - \mathbf{Q}_{(n-1)j}^{[1]}), \qquad j = 1, 2, ... n_2 \tag{2.30}$$

where c' is a constant derived from c and some knots and weights. Invoking the condition (2.23), we conclude that the two patches are G^1-continuous if

$$\mathbf{Q}_{nj}^{[1]} = \mathbf{Q}_{1j}^{[2]} = \frac{\mathbf{Q}_{2j}^{[2]} + c'\mathbf{Q}_{(n-1)j}^{[1]}}{1 + c'}, \qquad j = 1, 2, ... n_2 \tag{2.31}$$

An important implication of G^1-continuity is that it implies C^1-continuity in physical derivative. To see this, we first write the derivative conditions in the matrix form

$$\left[\frac{\partial \mathbf{x}^{[2]}}{\partial \xi^1} \ \frac{\partial \mathbf{x}^{[2]}}{\partial \xi^2}\right] = \left[\frac{\partial \mathbf{x}^{[1]}}{\partial \xi^1} \ \frac{\partial \mathbf{x}^{[1]}}{\partial \xi^2}\right] \left[\begin{array}{cc} c & 0 \\ 0 & 1 \end{array}\right] \tag{2.32}$$

where $\frac{\partial \mathbf{x}^{[2]}}{\partial \xi^2}$ etc. are understood as a column vector and take the value of the common edge. By construction (cf the local physical basis vectors $\{\bar{\mathbf{E}}_1, \bar{\mathbf{E}}_2\}$ on both sides match along the edge. Invoking the definition of $\mathbf{J}$ in Eq. (2.10), it is clear that,

$$\mathbf{J}^{[2]} = \mathbf{J}^{[1]} \left[\begin{array}{cc} c & 0 \\ 0 & 1 \end{array}\right] \tag{2.33}$$

By chain rule, it is straightforward to show that

$$\left[\frac{\partial \mathbf{x}^{[2]}}{\partial \bar{X}_1} \ \frac{\partial \mathbf{x}^{[2]}}{\partial \bar{X}_2}\right] = \left[\frac{\partial \mathbf{x}^{[1]}}{\partial \bar{X}_1} \ \frac{\partial \mathbf{x}^{[1]}}{\partial \bar{X}_2}\right] \tag{2.34}$$

which indicates that C^1 continuity is ensured in physical derivatives.

Carrying out the same analysis on the deformed configuration, it can be concluded that physical C^1-continuity is enforced if

$$\mathbf{q}_{1j}^{[2]} = \mathbf{q}_{nj}^{[1]} = \frac{\mathbf{q}_{2j}^{[2]} + c'\mathbf{q}_{(n-1)j}^{[1]}}{1 + c'}, \quad j = 1, 2, ...n_2 \tag{2.35}$$

In implementation, this condition is treated as a constant on the three columns of control points. The middle column is replaced by a linear combination of $\mathbf{q}_{2j}^{[2]}$ and $\mathbf{q}_{(n-1)j}^{[1]}$.

2.4   Trimmed NURBS

The tensor-product NURBS can only represent surfaces with four-edge topology. Trimmed NURBS can be used to represent an arbitrary surface. A trimmed NURBS is defined by a standard tensor product NURBS surface with some closed curves on the 2D parametric space that defines the boundary of the surface, as shown in Fig. 2.4. A boundary curve is expressed as follows:

$$\mathcal{C}(t) = \begin{pmatrix} \xi^1(t) \\ \xi^2(t) \end{pmatrix} \tag{2.36}$$

In simulation, we need to tessellate the parametric space to subdomains in triangles or quadrilaterals for numerical integration, as shown in Fig. 2.5. We tessellate the trimmed NURBS in three steps: (1) We tessellate the tensor product NURBS and the boundary curves. (2) We compute intersection between each NURBS cell and the boundary curves. And if intersection exist, we cut the cell into two polygons and discarding the one outside of boundary curve. (3) We split each polygon into triangles. It should be mentioned that these triangles are used for integration, so their shape doesn't influence the quality of shape function and simulation accuracy.
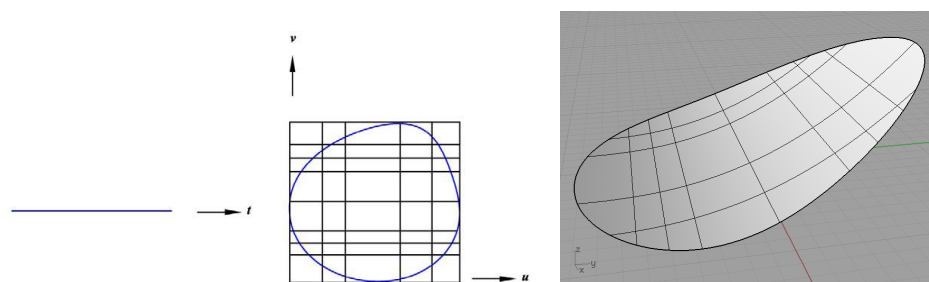
Figure 2.4: Illustration of trimmed NURBS

(a)   (b)
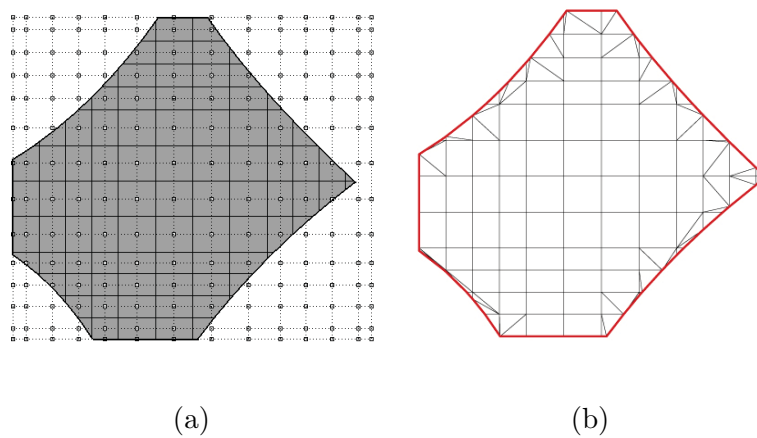
Figure 2.5: Illustration of tessellating trimmed NURBS. (a) A trimmed NURBS in CAD program; (b) A trimmed NURBS that is ready for analysis
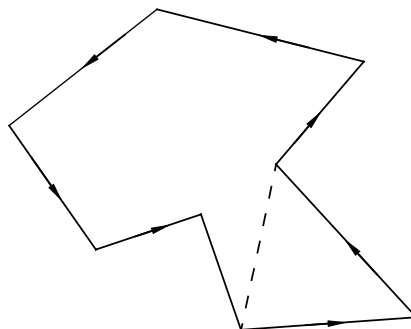
Figure 2.6: Illustration of polygon to triangles

Here we recommend a good but not unique way to split polygon: for all the convex corners of the polygon, we calculate the minimum angle α of the triangle enclosed by the corner. Then corner with the largest α is cut off, and a new polygon is generated, as shown in Fig. 2.6. The same operation is repeated on the new polygon again until the polygon degenerates to a triangle.

Because the domain outside the trimming boundary is discarded in integration and rendering (if the boundary is defined as clock-wise, then the inside part is discarded), among the control points of the original tensor-product NURBS, a part of them may have no influence on the simulation results (we call them unused control points). In order to avoid singularity, we don't assign freedoms to these unused control points, and at the end of each time step, we update the displacement of unused control points to be the same as its closest used control points. The problem is that some control points have very little influence, say having mass of 10^{-12}. Thus we create a criteria for unused control points:

$$A_I = \int_\Omega N_I d\Omega < \varepsilon \qquad (2.37)$$

where Ω represents the domain inside the trimming boundary, and ε is a user-defined tolerance. In our program, ε is set to be 10^{-6} times the element size.

2.5   Examples

Because we have not introduced our contact/impact yet, we only present some simple examples in this chapter. The first example is used to show the ability of NURBS to capture wrinkles. The second involves multi-patch model and mesh sen-

sitivity is studied. The third involves in trimmed NURBS. It should be noted that the examples here are not using quasi-static analysis, but using dynamic integration, which will be introduced in the following chapter.

2.5.1 Wrinkling in Shear

For a piece of cloth of dimension $0.38\,m \times 0.128\,m$, all the edges of the cloth are fixed, and the upper edge is translated by $3mm$ in horizontal direction, as shown in Fig. 2.7. The cloth is represented by a second order NURBS patch with 3332 control points. The displacement will form a parallelogram of approximately uniform wrinkles at 45° to the edges. The half wavelength of the wrinkle ca be predicted by [93].

$$\lambda = \sqrt{\frac{\pi H h}{\sqrt{3(1-\nu^2)\gamma}}} \qquad (2.38)$$

where $\gamma = \delta/H$ and h is the cloth thickness.

Elastic properties of the cloth are taken to be $E = 3500\,MPa$, $\nu = 0.31$, and linear bending model is used with bending stiffness $E_b = Eh^3/12$. The fabric thickness is $h = 0.1\,mm$, and the density is $\rho = 1500\,kg/m^3$. The damping coefficient is 2.0.

The simulation results are reported in Fig. 2.8, where the wrinkle angles are about 47°. The analytical wavelength is $25\,mm$, and the numerical wavelengths are given in Table 2.1. So the wavelengths obtained by simulation are close the analytical solution, but they are also influenced by the initial random perturbations.
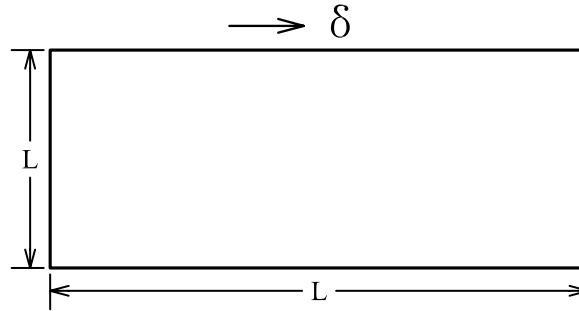
Figure 2.7: Wrinkling in shear

Figure 2.8: Wrinkling under shear force

Table 2.1: Wave lengths

wave #	wave length (mm)
1	28
2	26
3	26
4	26
5	24
6	25
7	25
8	27
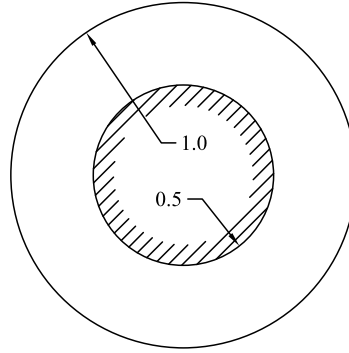
Figure 2.9: Initial configuration of clothdrapping over a pedestal

2.5.2 Cloth Draping over a Pedestal

This example simulates a piece of round shape cloth with radius 1.0m draping over a round table with radius 0.5m. Only the cloth part out of the table is modeled and its inner boundary is fixed, as shown in Fig. 2.9. The cloth is subject to gravity $g = 9.8m/s^2$ in the vertical direction. The membrane constitutive model uses Saint Venant-Kirchhoff material with Young's module $E = 70000Pa$, Poisson's ratio $\nu = 0.3$. The bending parameters $B_0 = 1.8 \times 10^{-3} \, N \cdot m$, $\kappa_0 = 30m^{-1}$. The surface density $\rho_0 = 0.144kg/m^2$ and damping constant $\gamma = 3\rho$. Fabric thickness $h = 0.001177m$ and persistent contact layer thickness $\delta = 0.01m$. The friction coefficient $\mu = 0.2$.

The cloth is represented by a second order NURBS patch and three different mesh sizes are used for simulation. The simulation results are shown in Fig. 2.10. The wrinkle number of last two models are same, showing that the number of wrinkle is mesh size independent. The wrinkle number of the first model is one less than the others, implying that the results converge as mesh refined.
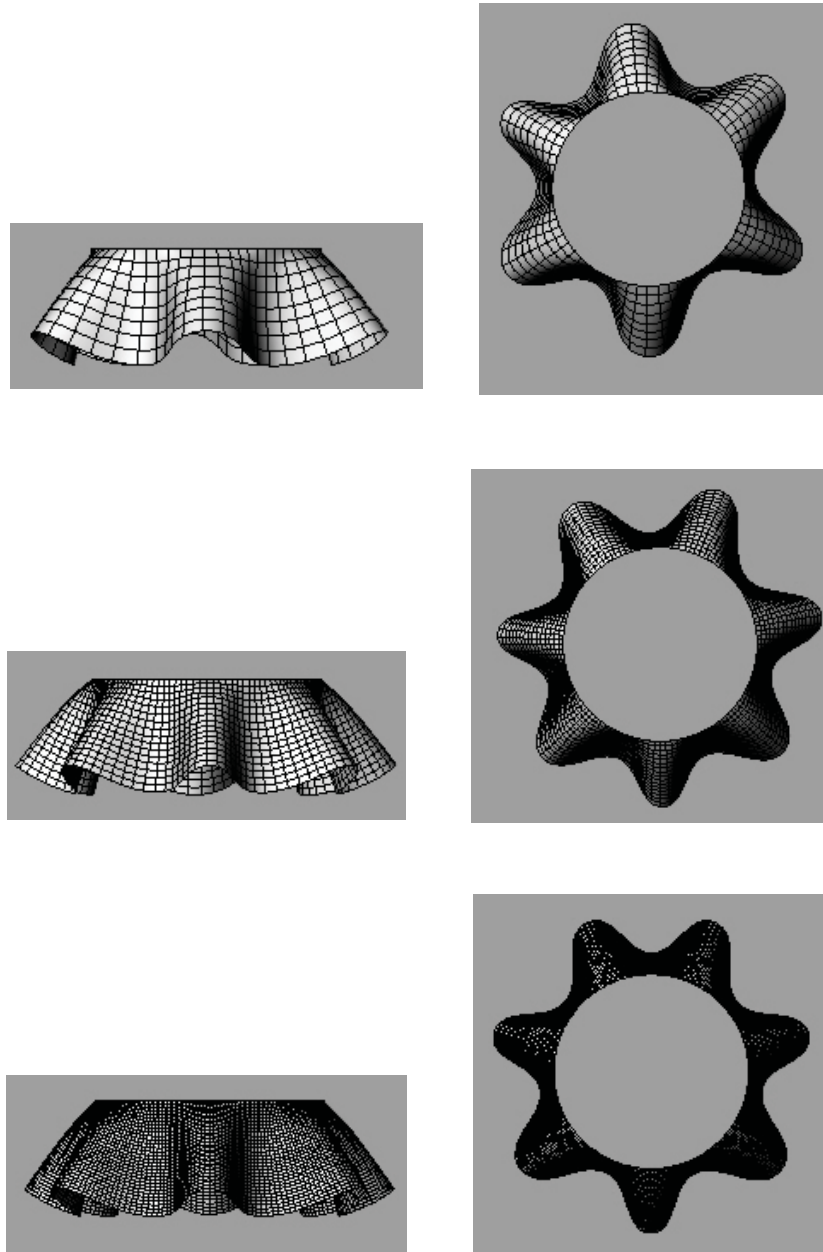
Figure 2.10: Simulation results of cloth draping over a pedestal.
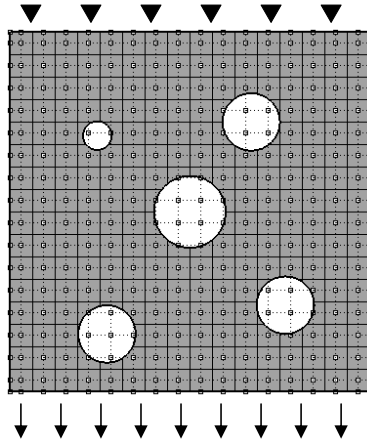
Figure 2.11: A trimmed-NURBS

2.5.3   Trimmed NURBS

A piece of cloth of dimension $1\,m \times 1\,m$ contains four randomly located holes, as shown in Fig. 2.11. The top edge of the cloth is fixed in vertical direction and the bottom edge of the cloth is subject to a vertical push p=1.1177 N/m. Material properties of the cloth are taken to be $E = 10000\,Pa$, $\nu = 0.3$, $B_0 = 3.3 \times 10^{-3}\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The fabric thickness is $h = 0.001177\,m$, and the surface density is $\rho = 0.117\,kg/m^2$. A linear damping force with damping constant $\eta = 0.351$. The cloth is described by a second order NURBS patch with 324 control points. The time increment Δt at the beginning of simulation is about $0.0025\,s$, and it is adjusted automatically as the cloth deforms. Snapshots at $t = 0\,s$ $t = 1\,s$ are presented in Fig. 2.12. This example shows that the trimmed NURBS allows us to describe a complicated shape with coarse mesh.

Undeformed   at t=1 s
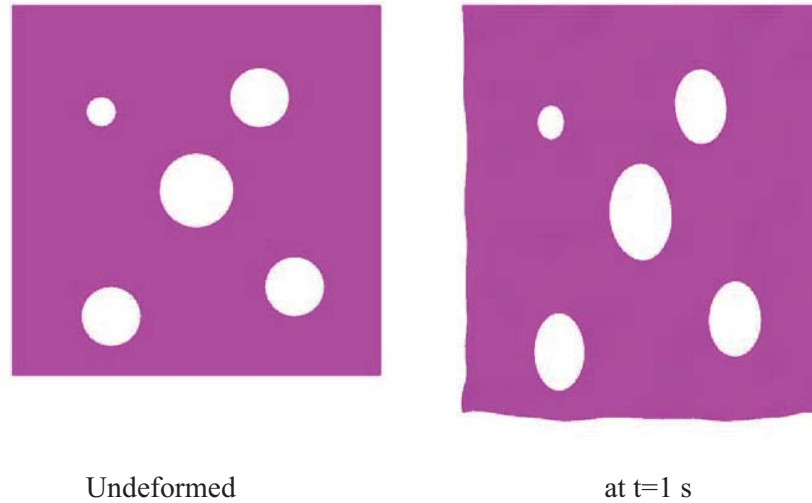
Figure 2.12: Simulation of a trimmed-NURBS

2.6   Conclusions

A NURBS-based isogeometric shell model of cloth is proposed. Leveraging the geometric smoothness of NURBS, the shell element utilizes only displacement-degree-freedom (DOFs) and thus saves a half of DOFs compared to the standard shell element. Continuum constitutive laws are used. To better represent the material behavior of cloth, the in-plane and the bending response are considered separately, and anisotropic constitutive law is employed. Compared with spring-based models, our continuum shell as utilized here can easily represent anisotropic property when the fabric weft and warp are not aligned with grid lines. Multi-patch models are also admitted in analysis, and the way to enforce C^1 continuity is discussed. The treatment of trimmed NURBS is also considered. The multi-patch and trimmed NURBS together ensure that typical NURBS designs from a cloth CAD software can

be admitted. The results of the numerical examples indicate that the smoothness of

NURBS geometry helps to simulate wrinkles.

CHAPTER 3
CONTACT/IMPACT SCHEME

3.1 Time Integration

3.1.1 Without Contact and Impact

The dynamic simulation in this thesis is completed by the Verlocity Verlet scheme, which was first applied to cloth simulation by [16]. The flowchart for time integration is as follows:

1. Update positions at t^{n+1} explicitly by

$$\begin{cases} \dot{\boldsymbol{q}}^{n+\frac{1}{2}} = \dot{\boldsymbol{q}}^n + \dfrac{1}{2}\ddot{\boldsymbol{q}}^n \Delta t, \\[2mm] \boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \dot{\boldsymbol{q}}^{n+\frac{1}{2}} \Delta t; \end{cases} \tag{3.1}$$

2. Update velocity and acceleration at t^{n+1} from

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{q}}^{n+1} + \boldsymbol{C}\dot{\boldsymbol{q}}^{n+1} + \boldsymbol{f}_{int}(\boldsymbol{q}^{n+1}) = \boldsymbol{f}_{ext}(t^{n+1}), \\[2mm] \dot{\boldsymbol{q}}^{n+1} = \dot{\boldsymbol{q}}^{n+\frac{1}{2}} + \dfrac{\Delta t}{2}\ddot{\boldsymbol{q}}^{n+1}. \end{cases} \tag{3.2}$$

This scheme in essence is the Newmark method with $\beta = 0$ and $\gamma = 1/2$. The time increment is taken to be $\alpha \Delta x \sqrt{\rho/E}$, where $E = \max\{d_1, d_2\}$ is the maximum in-plane stiffness, $\alpha < 1$ is a user parameter, and Δx is the minimum distance between two NURBS control points. A diagonal mass and damping matrix is used in the second step to avoid the solution of coupled linear equations.

The second line of Equation (3.1) indicates that the physical meaning of $\dot{\mathbf{q}}^{n+\frac{1}{2}}$ is the average velocity during the current time step. Since the position and velocity

of every continuum point are determined by $\mathbf{q}$ and $\dot{\mathbf{q}}$, it can be inferred that

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+\frac{1}{2}}. \tag{3.3}$$

3.1.2 With Contact and Impact

When contact or impact occurs, we introduce the contact/impact force $\mathbf{f}_c$ and write the weak form of the momentum equation as

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{damp} + \mathbf{f}_c. \tag{3.4}$$

The integral of Eq.(3.4) over time interval $[t_n, t_{n+1}]$ is given by

$$\dot{\mathbf{q}}(t_{n+1}) - \dot{\mathbf{q}}(t_n) = \mathbf{M}^{-1} \int_{t_n}^{t_{n+1}} (\mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{damp} + \mathbf{f}_c) d\tau. \tag{3.5}$$

Now we define $\mathbf{i}_c = \int_{t_n}^{t_{n+1}} \mathbf{f}_c d\tau$, which represents the contact/impact impulse on the time interval $[t_n, t_{n+1}]$, then

$$\dot{\mathbf{q}}(t_{n+1}) - \dot{\mathbf{q}}(t_n) = \mathbf{M}^{-1} \int_{t_n}^{t_{n+1}} (\mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{damp}) d\tau + \mathbf{M}^{-1}\mathbf{i}_c. \tag{3.6}$$

In the right-hand side of Eq.(3.6), the first term will be evaluated by Newmark method with $\gamma = 1/2$, while the second term will be determined by the contact/impact model directly (see Section 3.2.2). Then we have

$$\dot{\mathbf{q}}^{n+1} = \dot{\mathbf{q}}^n + \frac{1}{2}\Delta t \mathbf{M}^{-1}(\mathbf{f}_{ext}^n - \mathbf{f}_{int}^n + \mathbf{f}_{damp}^n) + \frac{1}{2}\Delta t \mathbf{M}^{-1}(\mathbf{f}_{ext}^{n+1} - \mathbf{f}_{int}^{n+1} + \mathbf{f}_{damp}^{n+1}) + \mathbf{M}^{-1}\mathbf{i}_c, \tag{3.7}$$

where $\Delta t = t_{n+1} - t_n$.

The right-hand side of Eq.(3.7) can be taken apart into three parts. The first part is denoted as $\tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}}$, written in

$$\tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}} = \dot{\mathbf{q}}^n + \frac{1}{2}\Delta t \mathbf{M}^{-1}(\mathbf{f}_{ext}^n - \mathbf{f}_{int}^n + \mathbf{f}_{damp}^n). \tag{3.8}$$

$\tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}}$ is considered as the average velocity obtained from the momentum equation without contact/impact force. This operation is the same as the first step of standard Velocity Verlet, but the results are labeled by a superimposed tilde, representing that the contact has not yet been accounted. Without contact accounted, the positions at t_{n+1} can be advanced by $\tilde{\mathbf{q}}^{n+1} = \mathbf{q}^n + \tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}}\Delta t$.

The term $\mathbf{M}^{-1}\mathbf{i}_c$ is considered as the second part of the right-hand side of Eq.(3.7). Let $\dot{\mathbf{q}}^{n+\frac{1}{2}}$ denotes the sum of the first two parts, given by

$$\dot{\mathbf{q}}^{n+\frac{1}{2}} = \tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}} + \mathbf{M}^{-1}\mathbf{i}_c\,. \tag{3.9}$$

Again, we update positions at t_{n+1} by $\dot{\mathbf{q}}^{n+\frac{1}{2}}$, written in $\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t\dot{\mathbf{q}}^{n+\frac{1}{2}}$, or in $\mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n+1} + \mathbf{M}^{-1}\mathbf{i}_c\Delta t$.

From Eq.(3.7), it is straightforward to find out the third part, and have the following relation:

$$\dot{\mathbf{q}}^{n+1} = \dot{\mathbf{q}}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t\mathbf{M}^{-1}(\mathbf{f}_{ext}^{n+1} - \mathbf{f}_{int}^{n+1} + \mathbf{f}_{damp}^{n+1})\,. \tag{3.10}$$

This operation is exactly the same as the second step of stand Velocity Verlet. Note that this operation doesn't change the final positions $\mathbf{q}^{n+1}$.

In summary, the Velocity Verlet method with contact and impact is given by: for time step counter $n = 1, 2...$

1.1 Update candidate positions at t^{n+1} explicitly by

$$\begin{cases} \tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}} = \dot{\mathbf{q}}^n + \frac{1}{2}\ddot{\mathbf{q}}^n\Delta t \\ \tilde{\mathbf{q}}^{n+1} = \mathbf{q}^n + \tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}}\Delta t \end{cases} \tag{3.11}$$

1.2 Update positions at t^{n+1} and average velocity by applying contact/impact impulse

$$\begin{cases} \dot{\mathbf{q}}^{n+\frac{1}{2}} = \tilde{\dot{\mathbf{q}}}^{n+\frac{1}{2}} + \mathbf{M}^{-1}\mathbf{i}_c \\[2mm] \mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n+1} + \mathbf{M}^{-1}\mathbf{i}_c\Delta t \end{cases} \tag{3.12}$$

The computation of $\mathbf{i}_c$ is discussed in Section 3.2.2.

2. Update velocity and acceleration at t^{n+1} from

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}}^{n+1} + \mathbf{D}\dot{\mathbf{q}}^{n+1} + \mathbf{f}_{int}(\mathbf{q}^{n+1}) = \mathbf{f}_{ext}(t^{n+1}) \\[2mm] \dot{\mathbf{q}}^{n+1} = \dot{\mathbf{q}}^{n+\frac{1}{2}} + \dfrac{\Delta t}{2}\ddot{\mathbf{q}}^{n+1} \end{cases} \tag{3.13}$$

This scheme was first proposed by Bridson et al. [15]. At the absence of compact/impact response, this algorithm reduces to the standard Velocity Verlet method. An important feature of the algorithm is that the contact/impact response is dealt with in the first substep. The update in the second substep is carried at fixed configuration, leaving intact the non-penetration state from step one.

Once the update is completed in the NURBS coordinates $\mathbf{q}$, the physical coordinates $\mathbf{x}$ follows from the mapping Eq.(2.6). It can be inferred from Eq.(3.11) and Eq.(3.12), that

$$\begin{cases} \mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\mathbf{v}^{n+\frac{1}{2}} \\[2mm] \tilde{\mathbf{x}}^{n+1} = \mathbf{x}^n + \Delta t\tilde{\mathbf{v}}^{n+\frac{1}{2}} \end{cases} \tag{3.14}$$

This implies that for any point on the cloth, once $\mathbf{x}^{n+1}$ and $\mathbf{x}^n$ are known, we can calculate $\mathbf{v}^{n+1/2}$ directly, and vice versa.

In order to establish a robust and efficient contact/impact scheme, we break contact/impact response into three kinds forces. The first kind is the persistent contact force, which is introduced when the distance between two primitives (including

nodes, elements, element edges) is smaller than a certain tolerance. The persistent force maintains a small distance between two primitives and try to avoid penetrations by keeping the contact pair within the tolerance to expelling them slightly part.

The second kind is impact response. This is called up when the trajectories of two primitives collide. The impact treatment keep the primitives on the correct side.

While impact treatment succeeds to prevent a penetration, it may cause collisions somewhere else, especially in the multilayer cases. Even if the impact treatment can stop all the penetrations during simulation, it can't eliminate self-intersections in the initial configurations. In these cases, we use the the third type of interaction, the intersection resorting force, to resolve the tangled cloth.

The three kinds of contact/impact force are introduced one by one in the following sections.

3.2   Persistent Contact Force

3.2.1   Contact Detection

For the persistent contact, we calculate the smallest distance between a grid point and a grid cell at the beginning of each time step. The normal vector of the grid cell is considered as the contact normal, noting to flip the normal vector when the point is on the back side of the cell.
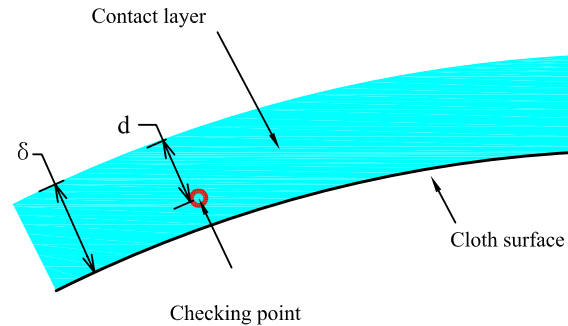
Figure 3.1: Illustration of persistent contact region

3.2.2   Contact Response

3.2.2.1   Normal Contact

Given a point $\mathbf{x}_A$, the normal distance to the target surface B is defined by $g = (\mathbf{x}_A - \mathbf{x}_B) \cdot \mathbf{n}$, where $\mathbf{x}_B$ is the closet projection point and $\mathbf{n}$ is the unit normal of B at that point, pointing to $\mathbf{x}_A$. Note that the direction of the normal is not pre-set, but defined stepwise by the relative position at the beginning of the time step. By construction, $g^n = (\mathbf{x}_A^n - \mathbf{x}_B^n) \cdot \mathbf{n} \geq 0$. In contact update, the point $\mathbf{x}_A$ is required to remain on the same side of the surface B during the time step. The rate of change of the normal displacement before contact is given by $\tilde{v}_{rn} = \mathbf{n} \cdot (\tilde{\mathbf{v}}_A^{n+1/2} - \tilde{\mathbf{v}}_B^{n+1/2})$. The contact condition is characterized as

$$
\begin{array}{lll}
g^n \leq \delta, \ \tilde{v}_{rn} \geq 0 & \text{persistent contact, separating} & \\
g^n \leq \delta, \ \tilde{v}_{rn} < 0 & \text{persistent contact, approaching} & (3.15) \\
g^n > \delta & \text{no contact} &
\end{array}
$$

It is worth noting that the contact condition is characterized by the configuration $\mathbf{x}^n$, not the predictor $\tilde{\mathbf{x}}^{n+1}$. In the spirit of explicit formulation, we use the surface normal $\mathbf{n}$ at t^n to compute the normal distance at t^{n+1}, namely $g^{n+1} = (\mathbf{x}_A^{n+1} - \mathbf{x}_B^{n+1}) \cdot \mathbf{n}$.

Introduce also $v_{rn} = \mathbf{n} \cdot (\mathbf{v}_A^{n+1/2} - \mathbf{v}_B^{n+1/2})$, the relative normal velocity after contact/impact. By multiplying $\mathbf{n}$ on both sides of Eq.(3.14), we have

$$\tilde{g}_{n+1} = g^n + \tilde{v}_{rn}\Delta t$$
$$g^{n+1} = g^n + v_{rn}\Delta t, \tag{3.16}$$

Approaching. We assume that at the end of time step the two points in contact will move at nearly the same velocity in the normal direction, that is $v_{rn} \approx 0$. Thus, $g^{n+1} \approx g^n$, according to $(3.16)_2$. For numerical purposes, we take g^{n+1} to be slightly greater than g^n by setting $g^{n+1} = g^n + kd$, where k is a small number, $0 < k < 1$. It follows that

$$v_{rn} = \frac{g^{n+1} - g^n}{\Delta t} = \frac{kd}{\Delta t} \tag{3.17}$$

Note that the slave point remains on one side of the target surface during the time step, and thus the impenetration condition is observed.

Separating. In this situation the repulsive force will accelerate the separation, pushing the point further away from the surface. We assume that $g^{n+1} = \tilde{g}^{n+1} + kd$. It follows that

$$v_{rn} = \frac{g^{n+1} - g^n}{\Delta t} = \frac{kd}{\Delta t} + \tilde{v}_{rn} \tag{3.18}$$

3.2.2.2 Tangential Contact

At the presence of friction, two pieces of cloth in contact/impact may also exchanges momentum in the tangential direction. The relative tangential velocity is given by $\mathbf{v}_{rt} = \mathbf{v}_r - v_{rn}\mathbf{n}$. If the frictional impulse is not sufficient to stop tangent movement, the bodies will slide relative to each other. According to (3.27) and

(3.30), the normal impulse, in magnitude, is $\frac{\rho_A\rho_B}{\rho_A+\rho_B}|\Delta v_{rn}|$ where Δv_{rn} is given by the contact/impact law. Thus, if the friction force follows the Coulomb law, the frictional impulse, in magnitude, is $\mu\frac{\rho_A\rho_B}{\rho_A+\rho_B}|\Delta v_{rn}|$, where μ is the frictional constant. It follows that

$$
\begin{aligned}
\Delta v_{At} &:= v_{At} - \tilde{v}_{At} = -\mu\frac{\rho_B}{\rho_A + \rho_B}\Delta v_{rn} \\
\Delta v_{Bt} &:= v_{Bt} - \tilde{v}_{Bt} = \mu\frac{\rho_A}{\rho_A + \rho_B}\Delta v_{rn},
\end{aligned}
\tag{3.19}
$$

from which we conclude that $\Delta v_{rt} = \Delta v_{At} - \Delta v_{Bt} = -\mu\Delta v_{rn}$.

On the other hand, preventing sliding requires a minimum frictional impulse of $\frac{\rho_A\rho_B}{\rho_A+\rho_B}|\tilde{v}_{rt}|$. Therefore, tangential sticking occurs if $|\tilde{v}_{rt}| < \mu|\Delta v_{rn}|$, in this case $v_{rt} = 0$. Putting them together, the tangential velocity change is given by

$$
\Delta v_{rt} = \begin{cases}
-\mu\Delta v_{rn} & \text{if } |\tilde{v}_{rt}| > \mu|\Delta v_{rn}| \qquad (3.20\text{a}) \\[2mm]
-\tilde{v}_{rt} & \text{if } |\tilde{v}_{rt}| \le \mu|\Delta v_{rn}| \qquad (3.20\text{b})
\end{cases}
$$

Finally, the change in relative velocity is given by

$$
\Delta\mathbf{v}_r = \Delta v_{rn}\mathbf{n} + \Delta v_{rt}\mathbf{t}
\tag{3.21}
$$

This result is substituted back to (3.30) to update the velocities of A and B.

3.3 Trajectory Impact Force

3.3.1 Impact Detection

For the trajectory impact, we test the intersections between the trajectories of a grid point and a grid cell. The trajectory impact test between a point and a triangle is achieved in the following way. First, we find a time t at which the point is co-planar with the triangle: therein the current time step. First, we find all the
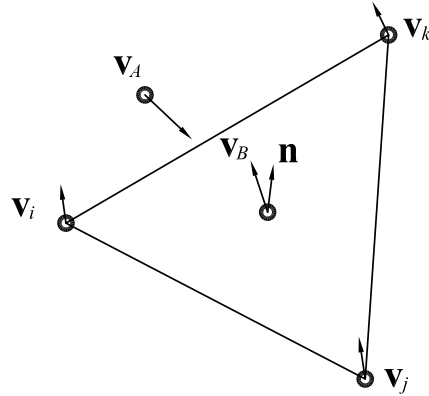
Figure 3.2: Impact between a point and a triangle

coplanar time by solving the following cubic polynomial equation

$$
\det \begin{vmatrix} 1 & 1 & 1 & 1 \\ \mathbf{x}_A^n + \tilde{\mathbf{v}}_A^{n+\frac{1}{2}}\tau & \mathbf{x}_i^n + \tilde{\mathbf{v}}_i^{n+\frac{1}{2}}\tau & \mathbf{x}_j^n + \tilde{\mathbf{v}}_j^{n+\frac{1}{2}}\tau & \mathbf{x}_k^n + \tilde{\mathbf{v}}_k^{n+\frac{1}{2}}\tau \end{vmatrix} = 0 \qquad (3.22)
$$

where $\tau = t - t^n$. This is a cubic equation for which closed form roots are available. If a time $t \in [t^n, t^{n+1}]$ is found, we check whether the point A is inside the triangle. If so, the solution of t is deemed as the impact time. If multiple impact times exist, the smallest one is used. The algorithm is illustrated in Figure 3.2.

3.3.2 Impact Response

3.3.2.1 Normal Impact

Impact is determined by the penetration test. The basic assumption for impact response is that the normal distance g^{n+1} is zero after impact. Numerically, we introduce a small separation, taken to be either $k\delta$ or g^n, whichever is smaller. From

this condition we can derive

$$
v_{rn} = \begin{cases} 0 & \text{if } g_n < k\delta \\[2ex] \frac{k\delta - g^n}{\Delta t} & \text{if } g^n \geq k\delta \end{cases}
\tag{3.23}
$$

The first case corresponds to a pure plastic impact. For the second case, we note that v_{rn} is negative, thus $|v_{rn}| = \frac{g^n - k\delta}{\Delta t}$. Since $\tilde{g}^{n+1} \leq 0$, $g^n - k\delta < g^n - \tilde{g}^{n+1}$. Divided by Δt, we conclude that $|v_{rn}| < |\tilde{v}_{rn}|$. This shows that the impact is not elastic.

3.3.2.2 Tangential Impact

The velocity change in tangential direction of impact is obtained in the same way as the contact friction. Finally, the change in relative velocity is given by

$$
\Delta \mathbf{v}_r = \Delta v_{rn} \mathbf{n} + \Delta v_{rt} \mathbf{t}
\tag{3.24}
$$

3.4 Intersection Resorting Force

When there is self-intersections in the current configuration, the resorting forces are introduced. Suppose a point A on an edge is coincident with a point B on an element at time t^n, we use a resorting force to introduce a relative velocity $\Delta \mathbf{v}_r$ so that the intersections will reduce globally at t^{n+1}. There are two measurements for the amount of global intersection: intersection contour length and intersection area. First, we should make $\Delta \mathbf{v}_r$ on the same direction as the gradient of intersection length or area for optimizing the intersection reduction. Also, we set the norm of $\Delta \mathbf{v}_r$ as a function gradient norm, given by

$$
|\Delta \mathbf{v}_r| = h_0 \frac{|\mathbf{G}|}{\sqrt{|\mathbf{G}| + g_0^2}},
\tag{3.25}
$$

in which h_0 and g_0 are two user defined constants.

More details about intersection detection and resolution will be discussed in Chapter 4.

3.5   Distributing Contact Forces to Objects

3.5.1   Exchange of Linear Momentum in Lateral Impact

The dynamics of cloth or cloth-like material possesses some unique features that render the impact problem simpler than that of bulk materials. In most practical problems, a piece of cloth will be subjected to impulsive forces only, not impulsive torques. In this case, an impact action will induce a change in linear momentum, while the angular momentum change is negligible. Thus, if we consider only the action of impact force (cf. the contact corrector in Sub-problem 1), the impulse equation becomes

$$\rho\dot{\mathbf{v}} = \mathbf{f}_{impact} \tag{3.26}$$

The change of linear momentum due to impulse is given by

$$\rho\Delta\mathbf{v} = \mathbf{i} \tag{3.27}$$

where $\mathbf{i} = \int_{t^n}^{t^{n+1}} \mathbf{f}_{impact}dt$ is the total impulse over the time step. It is clear that, a piece of cloth, despite being a continuous system, follows an impulse equation closely resembling that of particles. The weak form of this equation reads $\mathbf{M}\Delta\dot{\mathbf{q}} = \mathbf{I}$. It turns out to be easier to solve the momentum exchange in strong form.

Consider two pieces of cloth (or two sub-regions of the same piece of cloth) in impact. Let $\tilde{\mathbf{v}}_A$ be the velocity field on cloth A before impact and $\mathbf{v}_A$ the velocity

field after impact. The difference $\Delta \mathbf{v}_A = \mathbf{v}_A - \tilde{\mathbf{v}}_A$ gives the velocity change due to impact. For a given point A, we determine the contact point B on the target surface using the methods described later, and in this way we define the contact velocity field on B. By conservation of linear momentum,

$$\rho_A \Delta \mathbf{v}_A + \rho_B \Delta \mathbf{v}_B = \mathbf{0} \tag{3.28}$$

Note that this is a *field equation* that holds over the common contact surface. For later use, we need to relate $\Delta \mathbf{v}_A$ and $\Delta \mathbf{v}_B$ to the change in relative velocity. To this end, let us denote by $\tilde{\mathbf{v}}_r = \tilde{\mathbf{v}}_A - \tilde{\mathbf{v}}_B$ the relative velocity before impact, and by $\mathbf{v}_r = \mathbf{v}_A - \mathbf{v}_B$ the relative velocity after impact, and introduce $\Delta \mathbf{v}_r := \mathbf{v}_r - \tilde{\mathbf{v}}_r$, the change in relative velocity. It is straightforward to check that

$$\Delta \mathbf{v}_r = \Delta \mathbf{v}_A - \Delta \mathbf{v}_B \tag{3.29}$$

Solving Eq. (3.28) and Eq. (3.29) together gives

$$\begin{aligned} \Delta \mathbf{v}_A &= \frac{\rho_B}{\rho_A + \rho_B} \Delta \mathbf{v}_r \\ \Delta \mathbf{v}_B &= -\frac{\rho_A}{\rho_A + \rho_B} \Delta \mathbf{v}_r \end{aligned} \tag{3.30}$$

These two equations will be used in the contact and impact model. Later, we will determine "impact laws" to related $\Delta \mathbf{v}_r$ to the incoming relative velocity $\Delta \tilde{\mathbf{v}}_r$ and thus determine the velocity changes in A and B during impact/contact. The equation (3.30) is derived in the context of cloth-cloth contact, however, it applies to cloth-rigid body contact as well. The latter is recovered by setting the density of the other body to infinity.

3.5.2 Determine Impulse on Control Points

Combining Eq.(3.24) and Eq.(3.30), we can compute the velocity change of each point when two points impact or contact. But Eq.(3.12) requires determining a set of impulses located on control points representing the effect of contact or impact. Consider a point A with interpolation form given by

$$\mathbf{x}_A = \sum_I N_I(\mathbf{x}_A)\mathbf{q}_I. \tag{3.31}$$

Suppose there is an impulse $\mathbf{i}$ located on point A which is projected to control points by

$$\mathbf{i}_{cI} = N_I(\mathbf{x}_A)\mathbf{i}, \tag{3.32}$$

where $\mathbf{i}_{cI}$ is the impulse applied on control point I.

If m_I represents the lumped mass of control point I, we can yield the velocity change of control point from the impulse-momentum theorem, given by

$$\Delta\dot{\mathbf{q}}_I = \frac{\mathbf{i}_{cI}}{m_I} = \frac{N_I(\mathbf{x}_A)\mathbf{i}}{m_I}. \tag{3.33}$$

It is straight forward to check that

$$\Delta\mathbf{v}_A = \sum_I N_I(\mathbf{x}_A)\Delta\dot{\mathbf{q}}_I. \tag{3.34}$$

Substituting Eq.(3.33) into Eq.(3.34) yields

$$\mathbf{i} = \frac{m_I\Delta\mathbf{v}_A}{\sum_I N_I^2(\mathbf{x}_A)}. \tag{3.35}$$

Substituting Eq.(3.35) into Eq.(3.32) yields

$$\mathbf{i}_{cI} = \frac{m_I N_I(\mathbf{x}_A)\Delta\mathbf{v}_A}{\sum_I N_I^2(\mathbf{x}_A)}. \tag{3.36}$$

3.6 Axis Aligned Bounding Box Tree

the computation cost of contact/impact detection can be sharply reduced by a bounding-box tree algorithm. In the contact/impact detection, the time cost is of order N^2 where N is the total number of primitives (the basic geometric unit of contact/impact detection). If we can assemble all the primitives to a set hierarchical boxes, then we can only test the primitives and save detecting time significantly.

One of the most commonly used hierarchical structures is an axis-aligned bounding box tree (AABB tree) [85], in which edges of the box are parallel to the Cartesian coordinate. In our scheme, we need to check the trajectory intersection or distance between a point and a triangle. Thus we consider each triangle as a primitive of our AABB tree. Some literature [15] also detected intersection between two edges, thus they needed another AABB tree for edges. However, we didn't do so.

3.6.1   Building the AABB Tree

In order to build the AABB tree, we put all the primitives into the first box, and recursively subdivid the boxes until each box contains no more than one primitive. Fig. 3.3 shows the flow chart for building an AABB tree in our program. It is a little bit different than the common version in literatures. We are only concerned about the center of the primitives in building topology of the AABB tree. The actual bounding of primitives is only used in a lazy refitting operation, which will be discussed in the next subsection. This changing can avoid an endless loop when the centers of two primitives coincide, and can make subdivision cutting more accurate.

It should be noted that in trajectory impact checking, we have to include the whole trajectory of the primitive into its bounding box.

3.6.2   Lazy Refitting

Both cloth and rigid bodies are moving in simulation, and any movement will change the bounding of primitives. Fortunately, the movement of cloth and rigid body keeps the mesh topology, so lazy refitting is a good choice. Lazy refitting means keeping the topology of the initial bounding box tree and only updating the bounding of every box when primitive moves.

We design two kinds of lazy refitting: lazy refitting of one primitive and lazy refitting of all primitives. The former is typically used for contact update, and the later one is typically used for position advance of time integration.

The lazy refitting of the AABB tree utilizes the property that the bounding of a box is the union of all its son boxes. So in lazy refitting of one primitive, the box containing this primitive is updated first, and then its parent box is updated recursively until it reaches the top level. The lazy refitting of all primitives is accomplished by updating boxes level by level from bottom to top.

3.6.3   Fast Intersection Detecting

Once a bounding box is built, we can perform fast intersection or distant testing between a point and all triangles. First, we calculate the bounding of the point, considered as a master box. In distant testing, the bounding of a point is a box with edge length 2δ centered at itself. In trajectory intersection testing, the
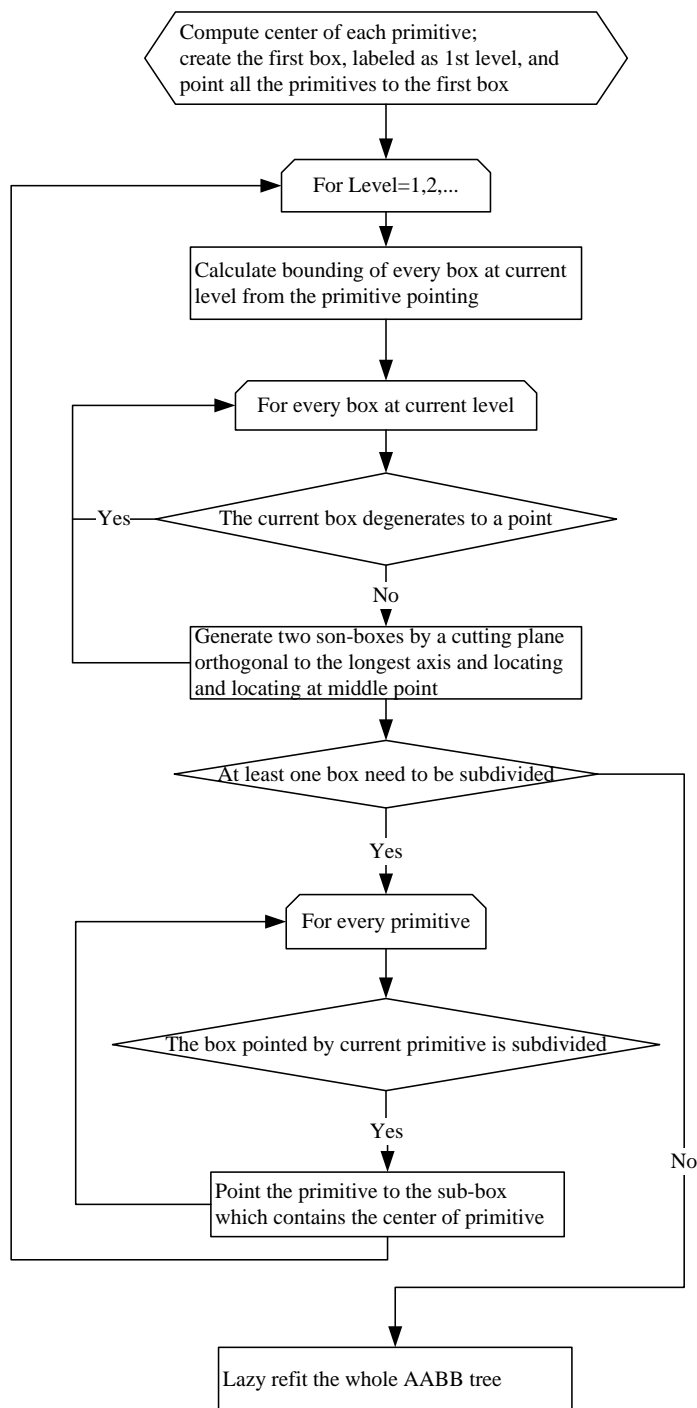
Figure 3.3: Flow chart of building AABB tree

bounding of a point is the minimum box containing its trajectory. Then we check the overlapping of the master box and the bounding boxes in the AABB tree (slave box). We test from top to bottom, and once a slave box proves to have no overlap, we discard all of the primitives inside it and its son-boxes. When a slave box with son-boxes proves to be overlapped, we further detect its son-boxes. When a slave box without son-boxes proves to be overlapped, we record all the primitives inside it as candidates, on which a real intersection or distant testing will be performed.

3.7   Examples

Several examples are presented in this section to demonstrate the performance of the contact treatment. The first tests on the capability of predicting quasi-static contact pressure. The second example is concerned with cloth draping, a benchmark problem for cloth simulation. The third example demonstrates the capability of complex self-contact handling. Finally a garment-level simulation is presented. All computations are conducted on a (single processor) laptop computer with 2.4GHz CPU and 3.0GB memory. A side advantage of NURBS modeling, although not essential for analysis, is that it can be readily rendered and textured in CAD programs. To demonstrate this point, we have textured the fabrics in Examples 2 to 4 while presenting the simulation results.

3.7.1   Contact Pressure

Since the contact/impact response laws are not experimentally characterized, it is imperative to examine whether such laws can at least predict a constant quasi-

static pressure field and a constant friction force. To check this, a simple contact/impact problem is simulated. A $0.6\,m \times 0.6\,m$ cloth is dropped to a slope, as shown in Figure 3.4. The cloth is initially parallel to the slope surface with a distance $0.1m$. The in-plane elastic parameters are $E = 30000\,Pa$, $\nu = 0.3$. The bending parameters are selected to be $B_0 = 3.3 \times 10^{-6}\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The 3D density $\rho_0 = 122.3\,kg/m^3$, and the Fabric thickness is $h = 0.001177\,m$, giving a surface density $\rho = 0.144\,kg/m^2$. The damping constant is taken to be $\eta = 0.432\,kg/(m^2 \cdot s)$. The contact layer thickness is $\delta = 0.001\,m$. For this problem, the minimum value of frictional coefficient for preventing sliding is $\mu_c = \tan 30°$. We consider two frictional coefficients, $\mu = 0.2$, and $\mu = \tan 30°$. The cloth is represented by a first order NURBS patch with 1 by 1 grid, and the four corners of the cloth are labeled as point 1 to point 4.
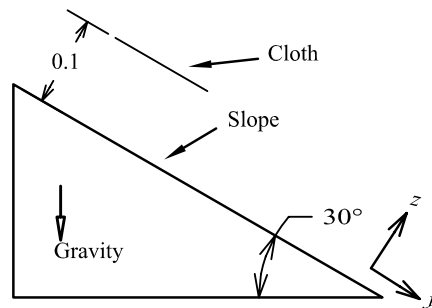

Figure 3.4: Schematics of the falling cloth problem

Figure 3.5(a) depicts the normal displacement vs. time. As shown, the cloth surface impacts the slope at about $t = 0.2\,s$, and the impact impulse stopped the cloth

from falling further down. After the initial impact update, a persistent contact begins to function. The contact algorithm moves the cloth asymptotically to a distance that equals to the contact layer thickness.

The contact pressure is computed by $p = \rho h \Delta v_n / \Delta t$. The analytical value for static contact is $p = \rho g \cos 30° = 1.2217\, Pa$. Figure 3.5(b) plots the pressure vs. time. After the pressure shock during and shortly after the impact step, the pressure converges to $p = 1.2216 Pa$, closing to the analytical value.

Figures 3.5(c) and (e) show the tangential displacement vs. time for the two cases. In case 1, the tangential displacement after the impact step increases quadratically with time, indicating that the cloth is sliding at a constant acceleration. In case 2, the tangential sliding is prevented after the impact step.

The friction force per unit area is computed from $F_f = \rho h \Delta v_t / \Delta t$. The analytical frictional forces of case 1 and case 2 are $F_f = 0.2443\, Pa$ and $F_f = 0.7053$, respectively. Figure 3.5(d) and (f) indicate that the frictional forces are accurately recovered.

3.7.2   Cloth Draping over a Sphere

The square cloth of dimension $0.4\, m \times 0.4\, m$ drapes under its own weight to a rigid sphere of radius $0.1\, m$, as shown in Fig. 3.6. Material properties of the cloth are taken to be $E = 30000\, Pa$, $\nu = 0.3$, $B_0 = 7 \times 10^{-6}\, N \cdot m$, $\kappa_0 = 30\, m^{-1}$. The fabric thickness is $h = 0.001177\, m$, and the surface density is $\rho = 0.144\, kg/m^2$. A linear damping force with damping constant $\eta = 0.288\, kg/(m^2 \cdot s)$ is included. The
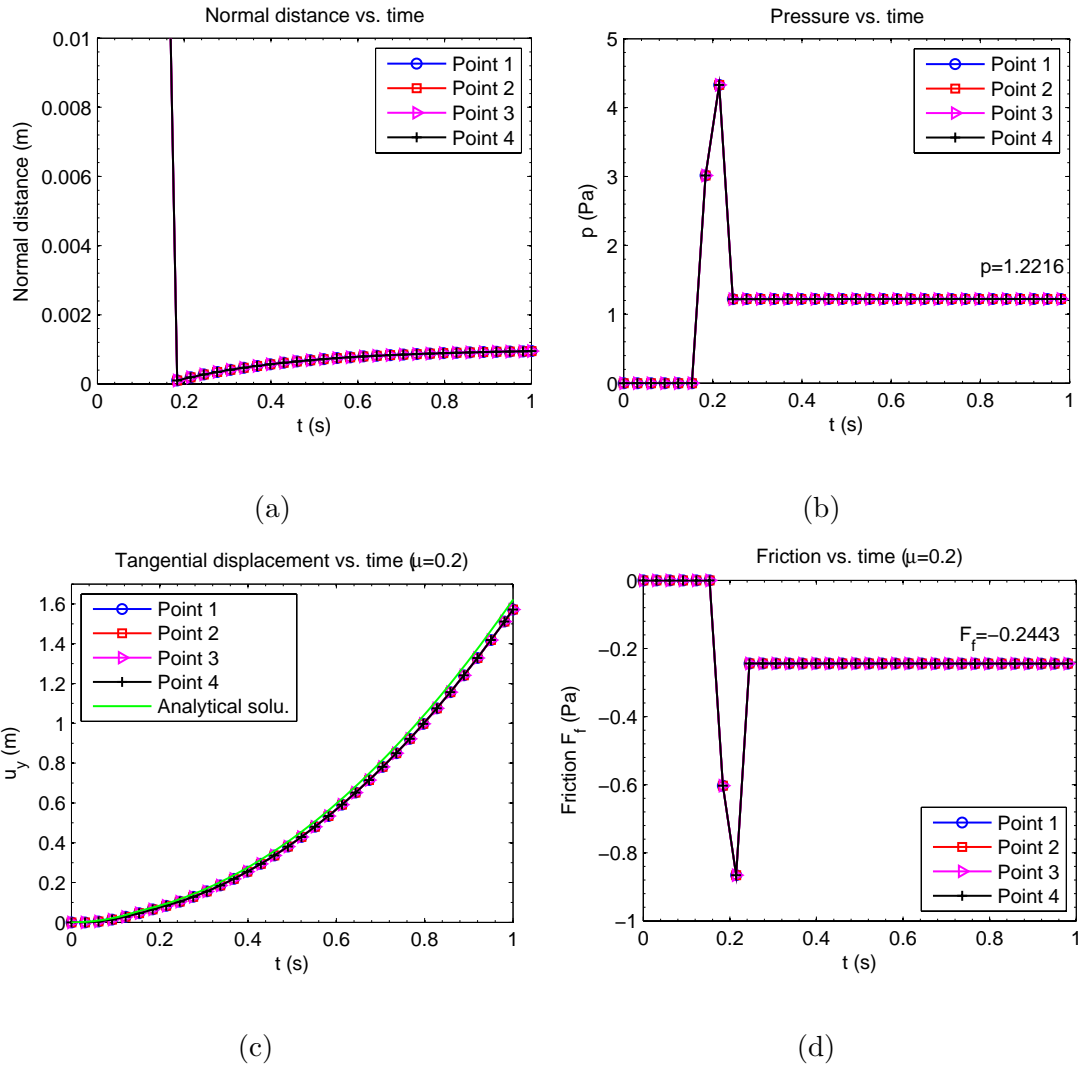
Figure 3.5: Flat cloth quasi-static contact. (a) Normal displacement; (b) Normal pressure; (c) Tangential displacement for $\mu = 0.2$; (d) Friction for $\mu = 0.2$; (e) Tangential displacement for $\mu = \tan 30°$; (f) Friction for $\mu = \tan 30°$
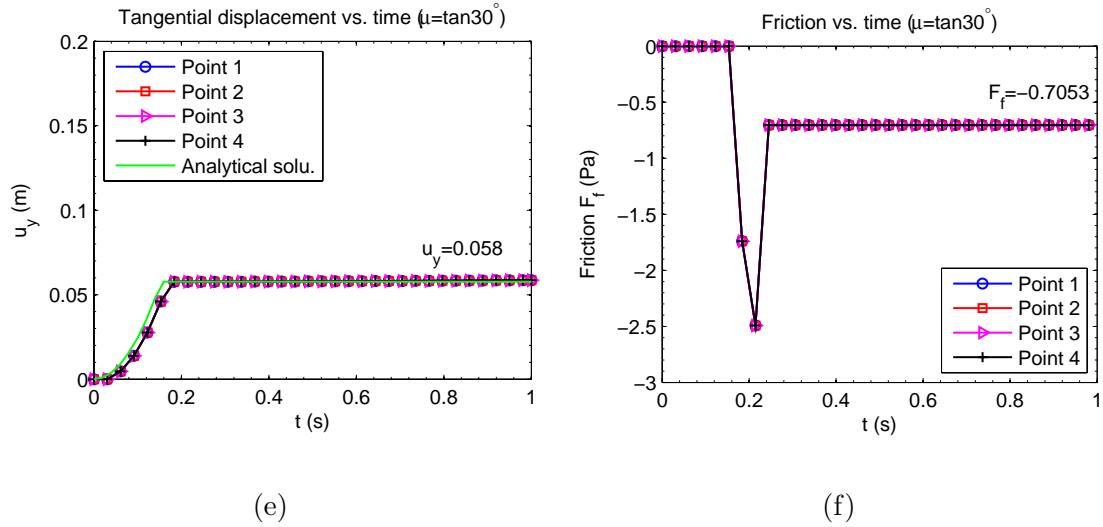
Figure 3.5: Continued

contact layer thickness is assumed to be $\delta = h$. The friction coefficient is $\mu = 0.2$. The cloth is represented by a second order NURBS patch with 1156 control points. The sphere is represented by a discrete mesh with 2048 cells. The time increment Δt at the beginning of simulation is about $0.0003\,s$, and it is adjusted automatically as the cloth deforms.

Snapshots of the draping process are presented in Fig. 3.7. Because of the non-uniqueness of the problem and the sensitivity of the results to model parameters, it is difficult to compare wrinkles with experiments. Nonetheless, the wrinkles appear vivid and visually appealing. The CPU time for this simulation is 55 seconds. The simulation results are rendered using Rhinoceros, a NURBS-based CAD software.

To check the momentum balance, we report the total contact force in the vertical direction and compare it with the weight of the cloth. The contact force
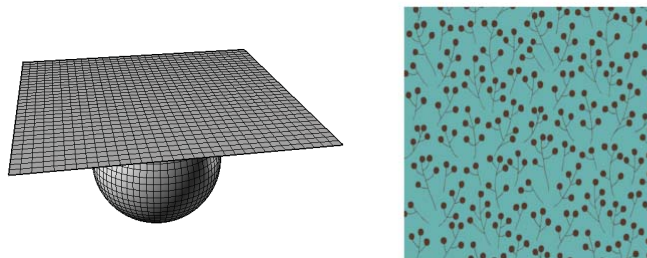
Figure 3.6: Initial configuration of the cloth that drapes over a sphere. Left: NURBS mesh with 2048 cells; right: texture view.
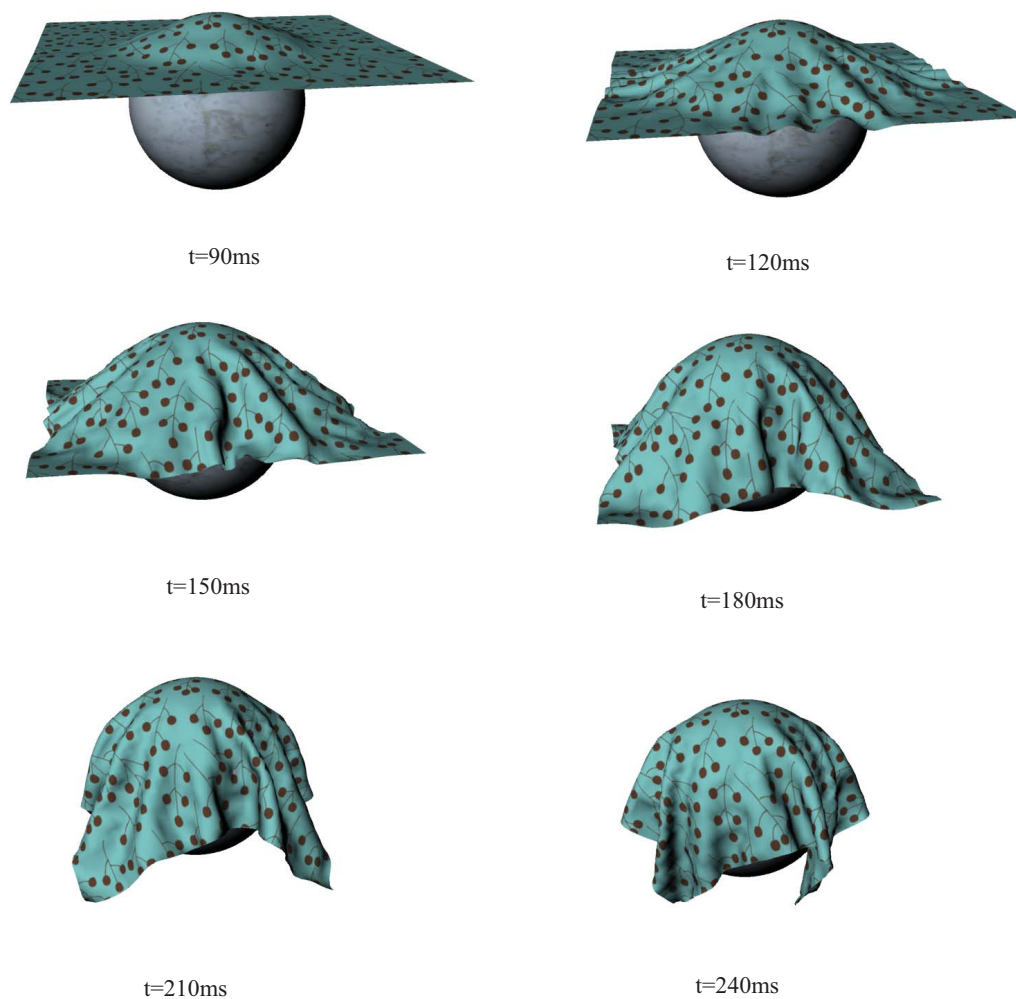

t=90ms

t=120ms

t=150ms

t=180ms

t=210ms

t=240ms

Figure 3.7: Snapshots of draping process.

in vertical direction is evaluated by $F_{cZ} = \sum_I m_I \Delta v_I / \Delta t$, where v_I is the vertical velocity of grid point I and m_I is its lumped mass (computed from the tessellation mesh). When the contact force is evaluated each step, the value oscillates even after the kinetic energy is almost damped out, as shown in Figure3.8(a). This oscillatory force may be attributed to the fact that balance of linear momentum is not enforced at the end of each time step. However, when the contact force is averaged over each frame (a frame = 0.033 s), the contact force appears smooth and converges to the total weight. See Figure 3.8(b).

3.7.3   Flag Draping

This example is introduced to test the self-contact handling. The flag drapes down by its own weight from the initial straight position. Without a proper self-contact treatment, self-penetrations are expected during the later stage of the motion. The NURBS model is shown in Fig. 3.9. The cloth is represented by a second order NURBS patch with 140 control points.

The in-plane elasticity parameters are taken to be $E = 30000\,Pa$ and $\nu = 0.3$. The bending parameters are $B_0 = 3.3 \times 10^{-6}\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The mass density is $\rho = 0.144\,kg/m^2$ and damping constant is $\eta = 0.432\,kg/(m^2 \cdot s)$. Fabric thickness is $h = 0.001177\,m$, and the persistent contact layer thickness is $\delta = h$. The friction coefficient is set to be $\mu = 0.2$. The left edge of the flag is fixed.

The time increment Δt at the beginning of the simulation is set to be $0.0014\,s$. The simulation results are shown in Fig. 3.10, rendered in Rhinoceros with texture. In
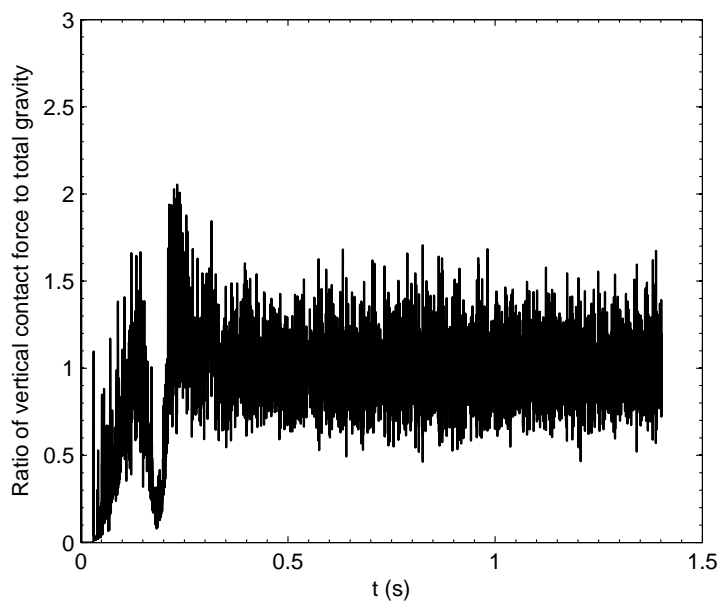
(a)
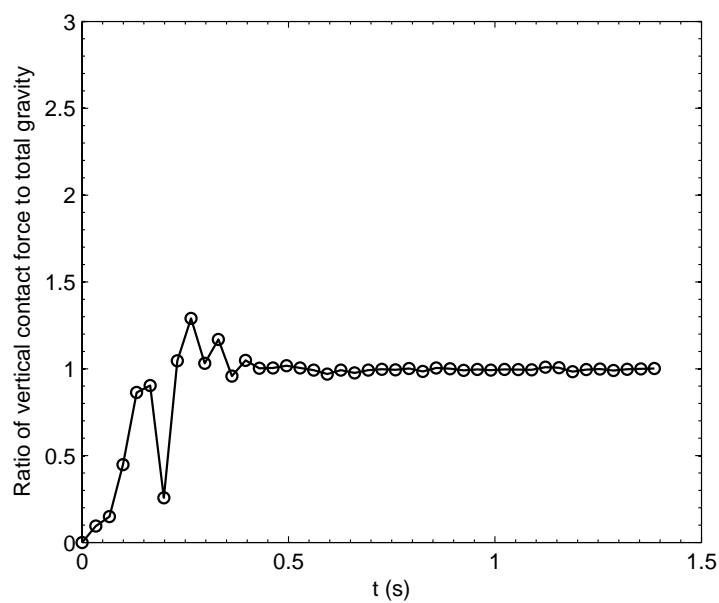
(b)

Figure 3.8: Total vertical contact force. (a) Evaluated over each time step; (b) averaged over each frame (0.033s)
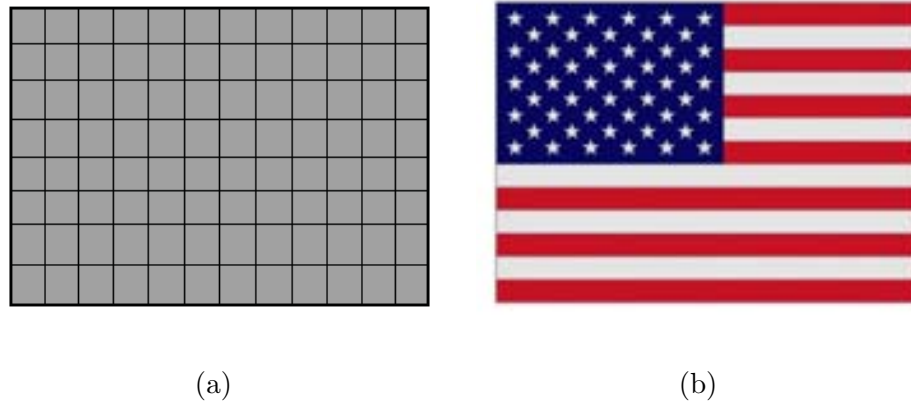
(a)   (b)

Figure 3.9: NURBS mesh of the flag

the computation, multiple occurrences of self-contact are logged, and yet the results show no self-penetration. The CPU time for this simulation is 33 seconds.

3.7.4   A Skirt

A practical garment is modeled. The 2D patches of the skirt design given in Fig. 3.11 consist of 8 second order NURBS patches. The entire model contains 960 control points. The woman's body is represented by a discrete mesh of 1755 cells. The membrane constitutive parameters are $E = 30000Pa$ and $\nu = 0.3$. The bending parameters are $B_0 = 1.0 \times 10^{-5} N \cdot m$ and $\kappa_0 = 30 m^{-1}$. The mass density is $\rho = 0.144 kg/m^2$ and damping constant is $\eta = 0.432 \, kg/(m^2 \cdot s)$. The fabric thickness is $h = 0.001177m$, and the contact layer thickness is taken to be $\delta = 0.005m$. A friction coefficient $\mu = 0.8$ is assumed.

The increment Δt is initially set to be $0.001\,s$. The garment shape at $t = 0.3\,s$, when the vibrations are mostly damped out, is shown in Fig. 3.12. The simulation
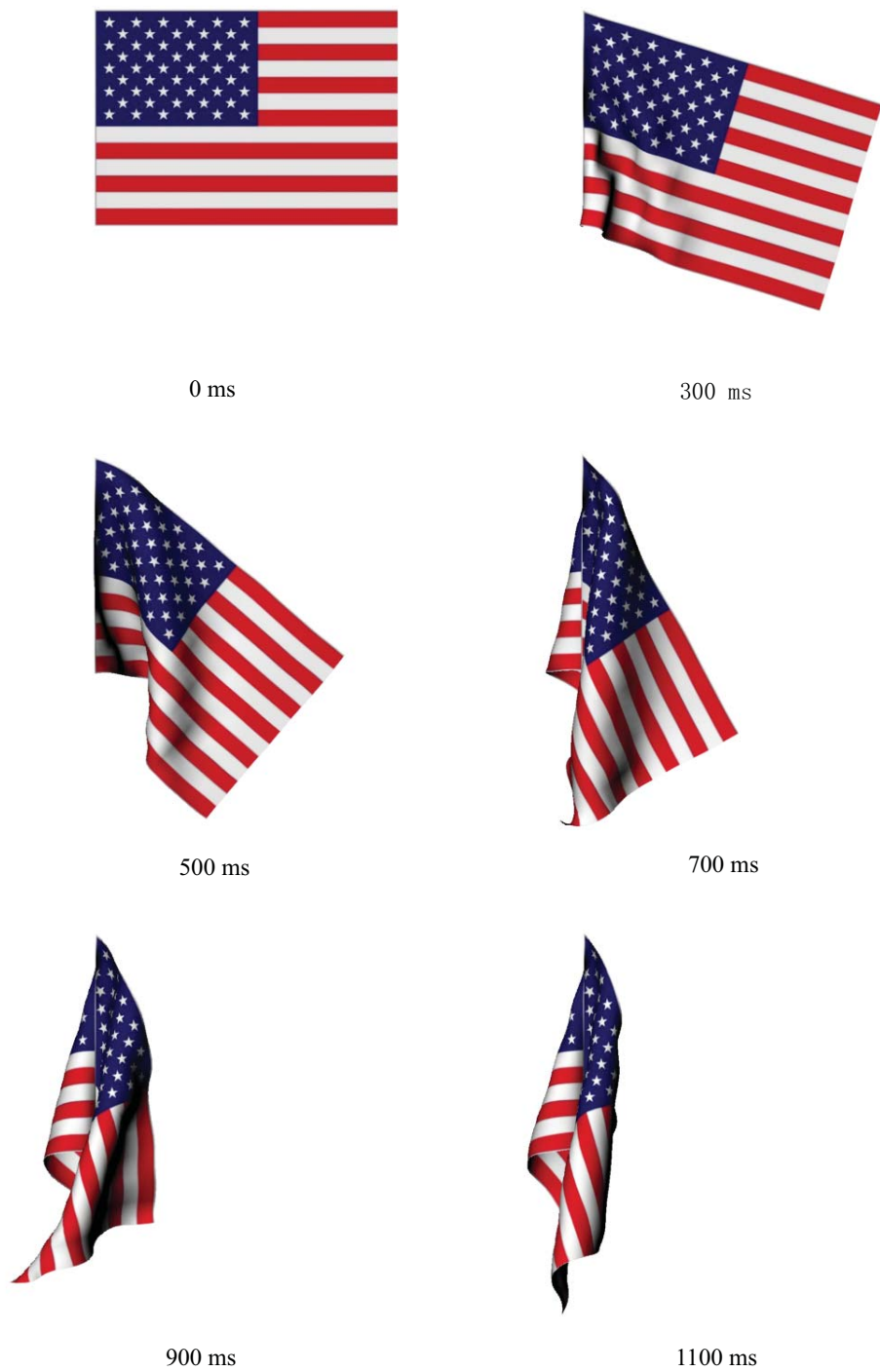
Figure 3.10: Snapshots of flag draping
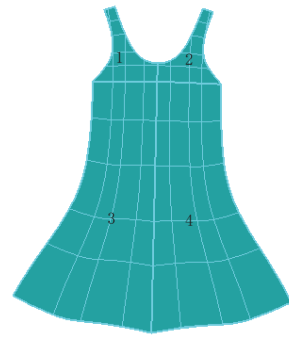
(a) 2D design of a skirt

(b) Human body

Figure 3.11: 2D design model of a skirt and a human body.

(a) Front view

(b) Back view

Figure 3.12: Simulated try-on effect

for virtual projecting and the following draping all together take 296 seconds of CPU time.

3.8   Conclusions

A contact/impact scheme is proposed in this chapter. The present scheme is based on an time integration in which the contact/impact force is singled out from the momentum equation. The contact/impact forces includes three parts: the persistent contact force, the trajectory impact force, and the intersection resolution force. Compared with the scheme in literatures, we have a slight change on the persistent contact force, and we proposed a novel intersection-resolution scheme through minimizing contour area. Numerical examples demonstrated that the current scheme can accurately predict constant quasi-static pressure field and treat complicated contact problems effectively.

CHAPTER 4
INTERSECTION RESOLUTION

4.1   Minimization of Intersection Contour Length

The detection of self-intersections is completed by check the intersections between each element edge and all the other elements. We follow the assumption in [4, 89] that no mesh vertex lies exactly in the plane of any non-neighboring mesh face, and no mesh edge intersects any non-neighboring mesh edge. We probablistically guarantee this assumption by adding a small amount of random noise to all vertex positions (actually we also need such noises for buckling simulation). Thus the intersection between two elements can be classified into two cases, as shown in Fig. 4.1, where the intersection line segment is denoted as $P_i P_{i+1}$. In order to study the reduction of contour length, we introduce a relative translation $\boldsymbol{d}$ between to two elements, and consider element I as reference.

Now we take P_i as example to show how to calculate the intersections. Suppose the equation of the edge is $\mathbf{x} = \mathbf{a}t + \mathbf{b}$ before translation, and it changes to $\mathbf{x} = \mathbf{a}t + \mathbf{b} + \mathbf{d}$ after translation. Suppose the equation of the plane is $\mathbf{n} \cdot \mathbf{x} + c = 0$. Then we can solve the coordinates of P_i after translation by combine the two equations, given by

$$\mathbf{x}_{P_i} = -\frac{\mathbf{n} \cdot (\mathbf{b} + \mathbf{d}) + c}{\mathbf{n} \cdot \mathbf{a}}\mathbf{a} + \mathbf{b} + \mathbf{d}. \tag{4.1}$$
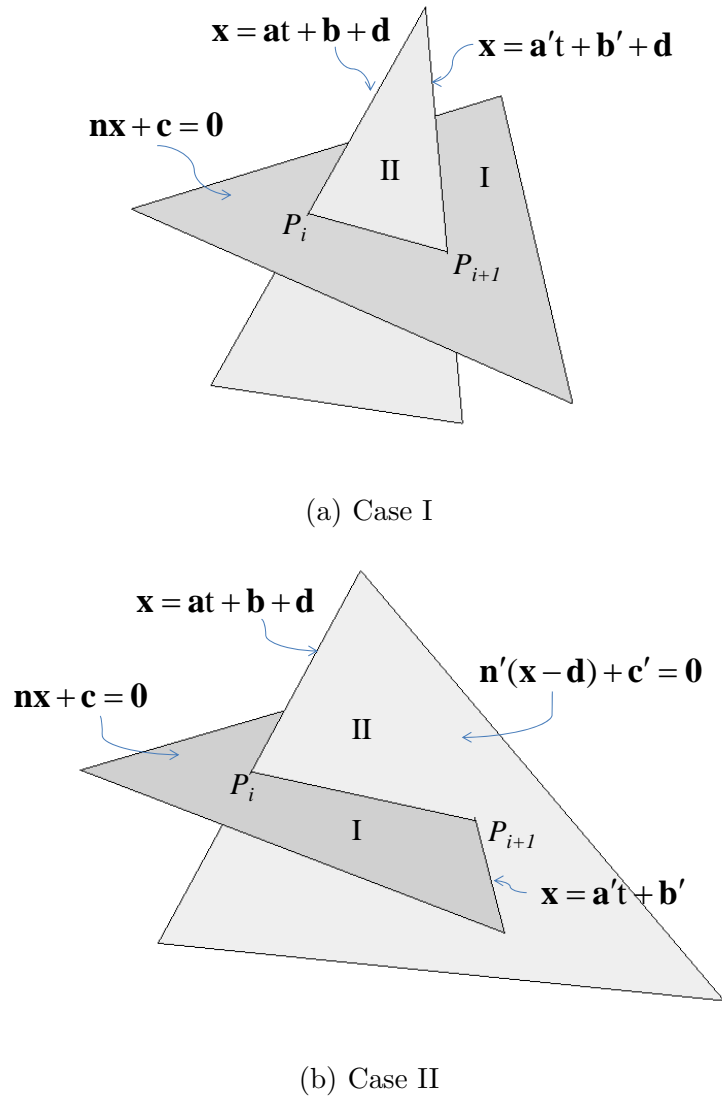
(a) Case I

(b) Case II

Figure 4.1: Intersection between two triangles.

In the same way we can obtain the coordinates of P_{i+1}, given by

$$
\mathbf{x}_{P_{i+1}} =
\begin{cases}
-\dfrac{\mathbf{n} \cdot (\mathbf{b}' + \mathbf{d}) + c}{\mathbf{n} \cdot \mathbf{a}'} \mathbf{a}' + \mathbf{b}' + \mathbf{d}, & \text{case I}, & (4.2) \\[3mm]
-\dfrac{\mathbf{n}' \cdot (\mathbf{b}' - \mathbf{d}) + c'}{\mathbf{n}' \cdot \mathbf{a}'} \mathbf{a}' + \mathbf{b}', & \text{case II}. & (4.2')
\end{cases}
$$

It is straight forward to check that

$$
\frac{\partial \overline{P_i P_{i+1}}}{\partial \mathbf{d}} =
\begin{cases}
\dfrac{\mathbf{a} \otimes \mathbf{n}}{\mathbf{a} \cdot \mathbf{n}} - \dfrac{\mathbf{a}' \otimes \mathbf{n}}{\mathbf{a}' \cdot \mathbf{n}}, & \text{case I}, & (4.3) \\[3mm]
\dfrac{\mathbf{a} \otimes \mathbf{n}}{\mathbf{a} \cdot \mathbf{n}} - \mathbf{I} + \dfrac{\mathbf{a}' \otimes \mathbf{n}'}{\mathbf{a}' \cdot \mathbf{n}'}, & \text{case II}. & (4.3')
\end{cases}
$$

Let $\mathbf{r} = \overline{P_i P_{i+1}}/|\overline{P_i P_{i+1}}|$ denote the direction of $\overline{P_i P_{i+1}}$. Because $\mathbf{r}$ is perpendicular to both $\mathbf{n}$ and $\mathbf{n}'$, and $\mathbf{n}$ and $\mathbf{n}'$ don't change during translating element II, $\mathbf{r}$ also keep unchanged during the translation. Using this property, we have

$$
\frac{\partial |\overline{P_i P_{i+1}}|}{\partial \mathbf{d}} = \mathbf{r} \cdot \frac{\partial \overline{P_i P_{i+1}}}{\partial \mathbf{d}}. \tag{4.4}
$$

Substituting Eq.(4.3) and (4.3') into Eq.(4.4) yields

$$
\frac{\partial \overline{P_i P_{i+1}}}{\partial \mathbf{d}} =
\begin{cases}
\dfrac{\mathbf{r} \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{n}} \mathbf{n} - \dfrac{\mathbf{r} \cdot \mathbf{a}'}{\mathbf{a}' \cdot \mathbf{n}} \mathbf{n}, & \text{case I}, & (4.5) \\[3mm]
\dfrac{\mathbf{r} \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{n}} \mathbf{n} - \mathbf{r} + \dfrac{\mathbf{r} \cdot \mathbf{a}'}{\mathbf{a}' \cdot \mathbf{n}'} \mathbf{n}', & \text{case II}. & (4.5')
\end{cases}
$$

For an intersection path, the contour length is the summation of the length of each line segment on the path. Thus, the gradient of contour length C is expressed as

$$
\frac{\partial C}{\partial \mathbf{d}} = \sum_{i=0}^{n-1} \frac{\partial |\overline{P_i P_{i+1}}|}{\partial \mathbf{d}}, \tag{4.6}
$$

where n is the number of vertices on the intersection contour.

4.2 Minimization of Intersection Area

For an arbitrary 3D surface domain Ω, it can be cut into two subdomains by a "closed contour". The "closed contour" here means that the contour either closes
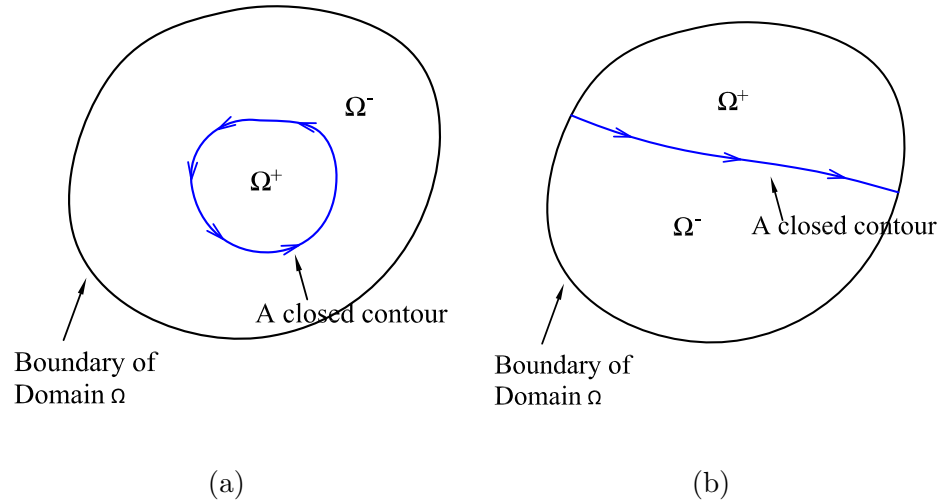
Figure 4.2: A closed contour divides a domain into two parts.

by itself as shown in Fig. 4.2(a), or ends on the boundary of the domain as shown in Fig. 4.2(b). We name the subdomain in the left-hand side (look along the director of the contour) of the contour as Ω^+, and the other one as Ω^-. And then we define the signed area as follows:

$$A^+ = \text{The area of } \Omega^+,$$
$$A^- = -\text{The area of } \Omega^-. \tag{4.7}$$

It follows that $A^+ - A^- = \text{The area of } \Omega$. Thus, when the cutting contour is translated by $\mathbf{d}$, $A^+ - A^-$ keeps constant, which implies

$$\frac{\partial A^+}{\partial \mathbf{d}} = \frac{\partial A^-}{\partial \mathbf{d}}. \tag{4.8}$$

For triangular meshes, the area of intersection region can be calculated in element-wise manner. Because our purpose is to compute $\partial A^+/\partial \mathbf{d}$, we only need to consider the elements along the intersection contour. In the present context, a

(a) Type I

(b) Type II

(c) Type III

(d) One element contains more than one subchains
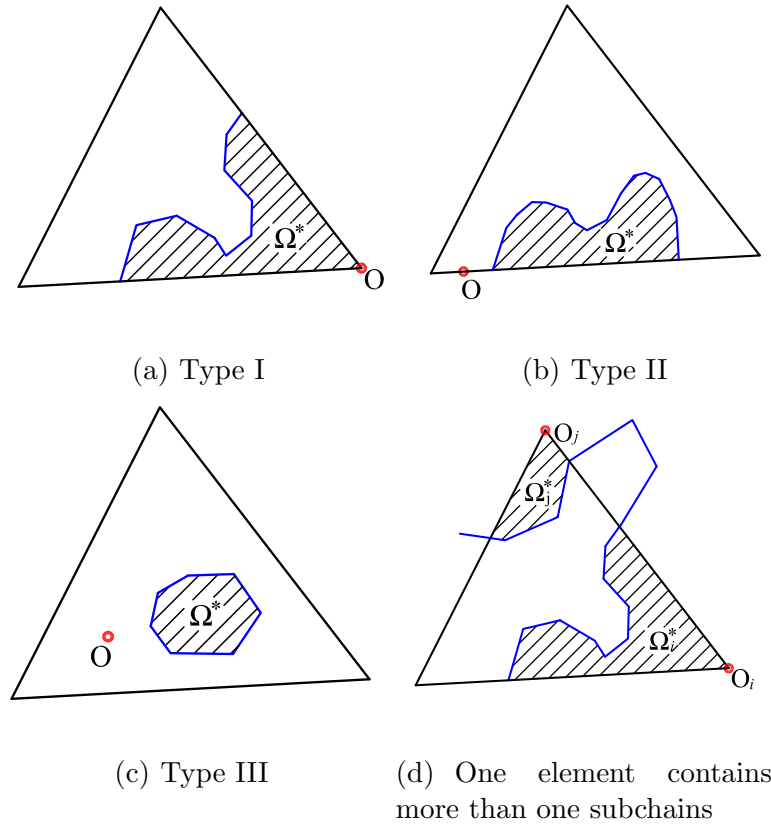
Figure 4.3: Intersection segments inside an element. (a), (b), (c) are three types of subchains. (d) shows an example that one element contains more than one subchains.

segment of an intersection contour is called a "subchain" if this segment belongs to the same element (we call it "host element"), and this segment cuts the element into two parts. As both parts of the host element are planar polygons, we introduce a way to calculate the signed area of a planar polygon. For an arbitrary planar polygon with vertices $\{v_1 v_2 ... v_n\}$, its singed area is given by

$$A(v_1 v_2 ... v_n) = \sum_{i=s}^{n-1} A(O v_i v_{i+1}), \tag{4.9}$$

where O is called reference point, and O can be an arbitrary point in the plane.

$A(O_iv_iv_{i+1})$ is the the signed area of the triangle $O_iv_iv_{i+1}$, given by

$$A(O_iv_iv_{i+1}) = \frac{1}{2}\,\text{sign}\,(\mathbf{s}_i \cdot \mathbf{n})\,|\mathbf{s}_i|, \tag{4.10}$$

in which $\mathbf{s}_i = \overline{O_iv_i} \times \overline{v_iv_{i+1}}$ and $\mathbf{n}$ is the normal vector of the plane.

The relation between a subchain and its host element can be classified into 3 types: (I) the subchain cuts two edges of the element, (II) the subchain cuts one edge of the element, (III) the subchain cuts zero edge of the element, as illustrated in Fig. 4.3(a), (b) and (c). It should be noted that even when one element contains more than one subchians, as shown in Fig. 4.3(d), we still handle the subchains one by one and don't need to care anything special. When a subchain cuts the element k into two parts, we use A_k^+ and A_k^- to denote the signed area of the to parts. It is straight forward to show that

$$\frac{\partial A^+}{\partial \mathbf{d}} = \sum_k \frac{\partial A_k^+}{\partial \mathbf{d}}. \tag{4.11}$$

Now we focus on one of subchains. It can be inferred that $\partial A_k^+/\partial \mathbf{d} = \partial A_k^-/\partial \mathbf{d}$, so we can focus on either one of the two parts. We use Ω^* to denote the part which is focused on (Ω_k^* equals either Ω_k^+ or Ω_k^-), and the way to find Ω^* is as follows: as shown in Fig. 4.3, in type I and II, the subdomain that doesn't include the non-intersected element edge (edges) is Ω^*; in type III, the subdomain inside the subchain is Ω^*. One property of Ω^* is that it contains at most one vertex that is not on the subchain. This property will simplify the area calculation in the following context.

In the loop of calculating the area of a polygon, Eq.(4.9), we need a reference point O which can be an arbitrary point in the plane. But in calculating A_k^+, we

want to carefully select the reference point so that we only need a loop for the edges
of the subchain, rather than all the edges of domain Ω^*. As shown in Fig. 4.3(a),
for a subchain of type I, we select the common vertex of the host element and Ω^* as
the reference point. Such a reference point ensures that we can omit the two edges
that contain the reference point in calculating A_k^+. For type II, as shown in Fig.
4.3(b), the reference point should be an arbitrary point on the element edge which
is intersected by the subchain. In type III, the reference point can be an arbitrary
point in the element plane.

As each vertex of a subchain owns a global index in the whole intersection
contour, we denote a subchain as $\{P_s P_{s+1} ... P_t\}$, where s and t are the indices of the
first and the last vertex of the subchain, respectively. Thus, the signed area of Ω_k^*
reads

$$A_k^* = \sum_{i=s}^{t-1} A(O_i P_i P_{i+1}), \tag{4.12}$$

in which O_i is same as O.

Finally the gradient of A^+ is expressed as

$$\frac{\partial A^+}{\partial \mathbf{d}} = \sum_{i=0}^{n-1} \frac{\partial A(O_i P_i P_{i+1})}{\partial \mathbf{d}}. \tag{4.13}$$

Thus once the reference point of each intersection line segment is assigned, the
gradient of intersection area can be calculated as easy as the contour length.

We note here that the direction of an intersection contour has influence on
the sign of the area gradient. For the intersection between two different layers, if the
points in Ω^- are in incorrect layer order (in the negative side of lower layer or in the

positive side of high layer), we reverse the sign of $\partial A^+/\partial \mathbf{d}$. This makes us to place the cloth in a correct layer order. For the intersections of cloth in the same layer, if $|A^+| < |A^-|$, we will reverse the sign of $\partial A^+/\partial \mathbf{d}$. This makes us to always remove the smaller one of the two divided regions. Checking the condition $|A^+| < |A^-|$ doesn't really need to calculate the value of A^+ and A^-. [4] suggested a flood-fill algorithm to pick the smaller one quickly and easily. We label the mesh nodes on the intersection edges first, e.g. the nodes on the left side of the contour are labeled as '+1', and those on the right side as '-1'. The nodes with label '+1' are considered as a seed, and the '+1' region is advanced to '+2' region by involving the neighboring nodes of '+1' nodes, and so forth for region +2,+3,... We perform this simultaneously to '-1' region. When one side stops to increase and its area is smaller than the other one, the smaller side is decided. One detail that [4] doesn't mention is that when one node connects to more than one intersection points, we should pick the closest one to judge its side. Otherwise, the cases in Fig. 4.3(b),(d) will be misjudged.

Now let's focus on one line segment $\{P_iP_{i+1}\}$. The coordinates of point P_i is given by Eq.(4.1). During translating element II, point O_i is not changed, and thus $\partial \mathbf{x}_{O_i}/\partial \mathbf{d} = \underline{0}$. It is straight forward to check that

$$\frac{\partial \overline{O_iP_i}}{\partial \mathbf{d}} = -\frac{\mathbf{a} \otimes \mathbf{n}}{\mathbf{n} \cdot \mathbf{a}} + \mathbf{I}. \tag{4.14}$$

Since the direction of $\overline{P_iP_{i+1}}$ keeps unchanged during the translation, we have

$$\frac{\partial \overline{P_iP_{i+1}}}{\partial \mathbf{d}} = \mathbf{r} \otimes \frac{\partial l}{\partial \mathbf{d}}. \tag{4.15}$$

We define the vector operation $\boldsymbol{\Omega}(\mathbf{a})$ as follows:

$$\boldsymbol{\Omega}(\mathbf{v}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \tag{4.16}$$

and then we have the relation $\boldsymbol{\Omega}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

Using Eq.(4.14) and Eq.(4.15), and noting that $\overline{O_i P_i} \times \overline{P_i P_{i+1}}$ is in the same or opposite direction of $\mathbf{n}$, we finally yield

$$\frac{\partial A(O_i P_i P_{i+1})}{\partial \mathbf{d}} = \frac{1}{2}\mathbf{n} \cdot \left[\frac{\mathbf{s}}{l} \otimes \frac{\partial l}{\partial \mathbf{d}} + \boldsymbol{\Omega}(\overline{P_i P_{i+1}}) \left(\frac{\mathbf{a} \otimes \mathbf{n}}{\mathbf{n} \cdot \mathbf{a}} - \mathbf{I} \right) \right], \tag{4.17}$$

where $\mathbf{s} = \overline{O_i P_i} \times \overline{P_i P_{i+1}}$, and $l = |\overline{P_i P_{i+1}}|$.

It should be noted that the change of contour area on element I and element II are different. We use $\mathbf{n}'$ to denote the normal vector of element II, and use O' to denote the reference point of on element II. Noting that $\partial \mathbf{x}_{O'}/\partial \mathbf{d} = \mathbf{I}$, we have

$$\frac{\partial A(O_i' P_i P_{i+1})}{\partial \mathbf{d}} = \frac{1}{2}\mathbf{n}' \cdot \left[\frac{\mathbf{s}'}{l} \otimes \frac{\partial l}{\partial \mathbf{d}} + \boldsymbol{\Omega}(\overline{P_i P_{i+1}}) \frac{\mathbf{a} \otimes \mathbf{n}}{\mathbf{n} \cdot \mathbf{a}} \right], \tag{4.18}$$

where $\mathbf{s}' = \overline{O_i' P_i} \times \overline{P_i P_{i+1}}$.

4.3   Restoring Force

Section 4.2 only introduces the intersection area of closed contours. For the unclosed intersection contour, we consider the intersection region as a square of length C, where C is the contour length. The area minimization vector $\mathbf{G}$ is defined as $-\frac{\partial A}{C \partial \mathbf{d}}$, and then we have

$$\mathbf{G} = \begin{cases} -\dfrac{\partial A}{C \partial \mathbf{d}}, & \text{if the contour is closed,} \\[2ex] -\dfrac{2\partial C}{\partial \mathbf{d}}, & \text{otherwise.} \end{cases} \tag{4.19} \tag{4.19'}$$
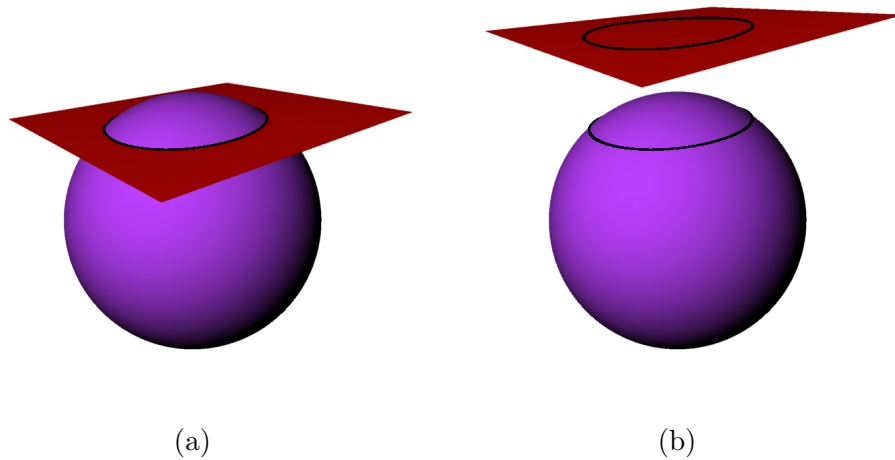
Figure 4.4: Intersection Comtour. (a) normal view; (b) exploded view.

For an intersection contour pair, the two contours coincident but the two intersection regions are different, as shown in Fig. 4.4. We use $\mathbf{G}_I$ and $\mathbf{G}_{II}$ to represent the directions obtained from two different intersection regions, and the one with bigger norm is used for contact response, which means that

$$
\mathbf{G} =
\begin{cases}
\mathbf{G}_I, & \text{if } |\mathbf{G}_I| > |\mathbf{G}_{II}|, \\
\\
\mathbf{G}_{II}, & \text{otherwise.}
\end{cases}
\tag{4.20}
$$

$$(4.20')$$

For the contact response, we use the same scheme as [89]. The velocity change $\Delta\mathbf{v}_r$ is in the same direction as $\mathbf{G}$, and its norm is a function of the norm of $\mathbf{G}$, given by

$$
|\Delta\mathbf{v}_r| = h_0 \frac{|\mathbf{G}|}{\sqrt{|\mathbf{G}| + g_0^2}},
\tag{4.21}
$$

in which h_0 and g_0 are two user defined constants.

4.4 Implementation

Our intersection resolution is performed progressively along the simulation. It can be integrated with any kind of time integration, and our testing codes use the implicit-explicit time integration proposed by [15]. The work flow of each time step is as follows:

1. Detect the intersections; when intersections are detected, we introduce geometric corrections of velocity and position on the points along the intersection contours.

2. Perform time integration over current time interval, and predict the position and velocity at the end of current time step.

3. Detect the proximity and trajectory collision; when proximity or trajectory collisions are found, we correct the position and velocity at the end of current time step.

It should be noted that the points involved in self-intersection are exempt from the detection in step 3, so that the collision response will not stop the untangling correction. Besides, we don't have detection-correction iteration and RIZ as [15], because the missed collisions will be handled by the intersection resolution scheme at the beginning of next time step.
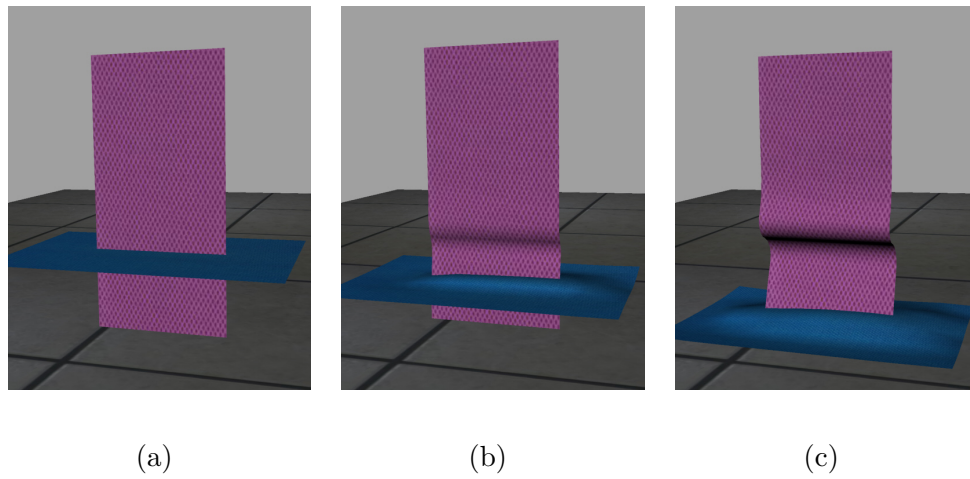
(a)   (b)   (c)

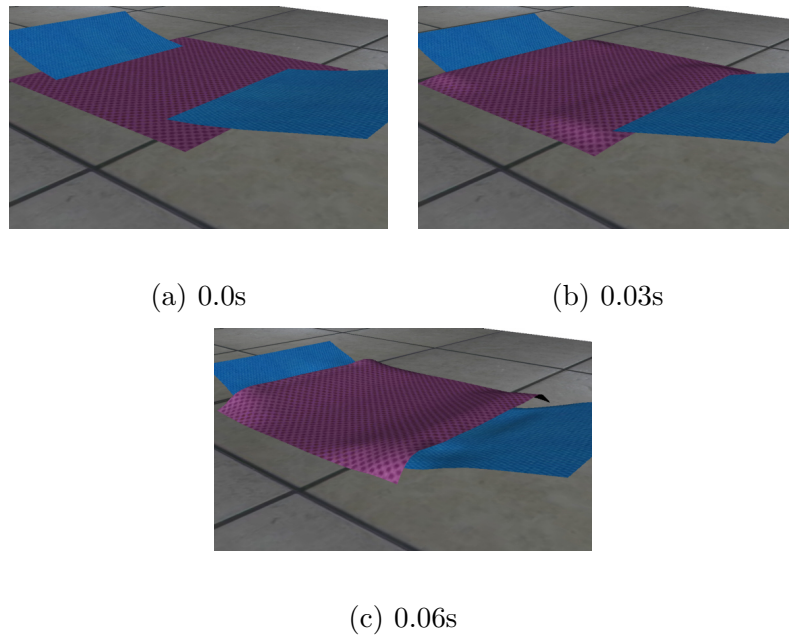Figure 4.5: An example of local minimization of contour length. (a) 0.0s (b) 0.1s,(c) 0.2s.

(a) 0.0s   (b) 0.03s

(c) 0.06s

Figure 4.6: Local minimization of contour length.

(a) 0.0s   (b) 0.06s

(c) 0.15s
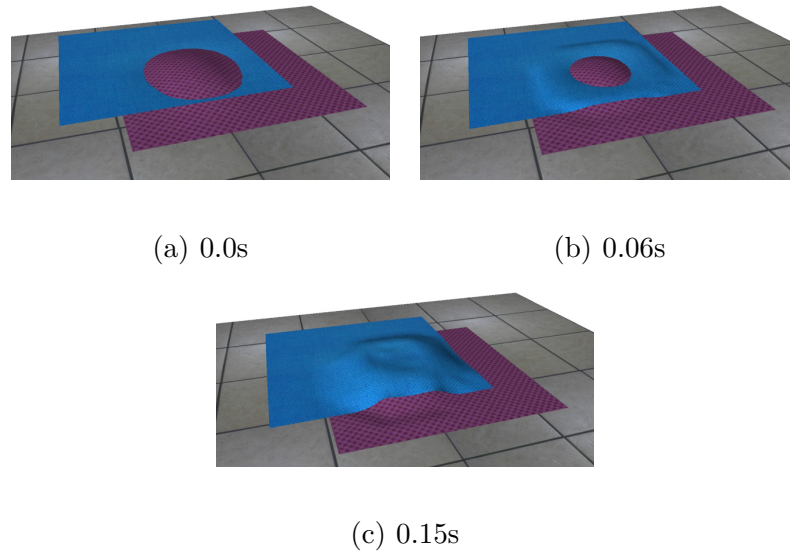
Figure 4.7: Both contours are closed.

(a) 0.0s   (b) 0.12s

(c) 0.24s
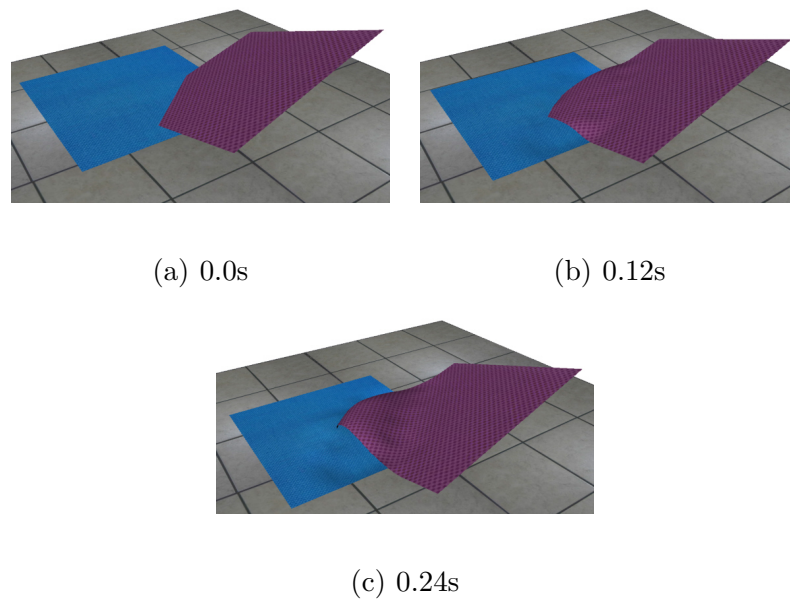
Figure 4.8: One contour is closed while the other one is unclosed.

(a) 0.0s   (b) 0.40s
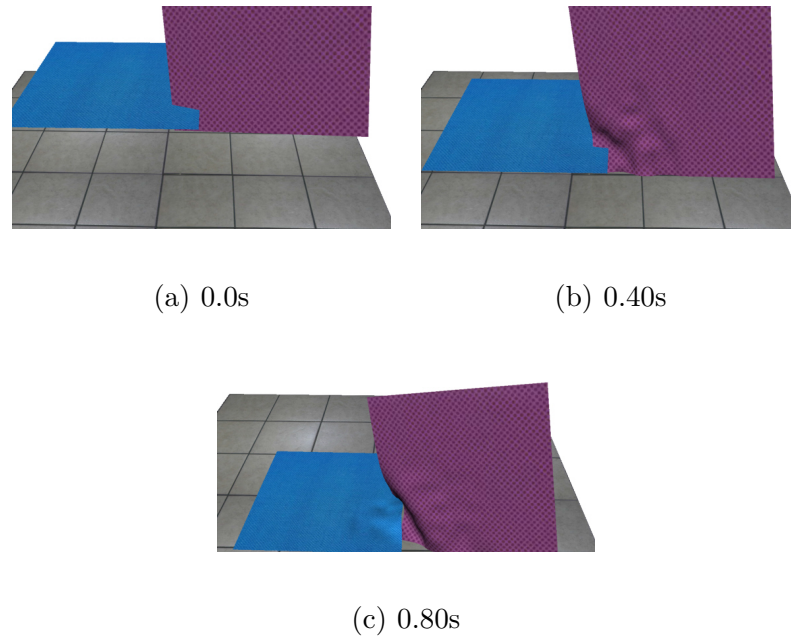

(c) 0.80s

Figure 4.9: Both contours are unclosed.

4.5   Examples

4.5.1   Typical Cases of Intersection Resolution

This section shows several simple examples representing different types of intersections. Fig. 4.5 and Fig. 4.6 shows that the present scheme can easily untangle two pieces of cloth with local minimization of contour length. Fig. 4.7, Fig. 4.8, Fig. 4.9 show intersected cloth with two, one and none closed contours, respectively. It is demonstrated that the present method can resolve various kinds of intersections, including the one with local minimization of contour length.
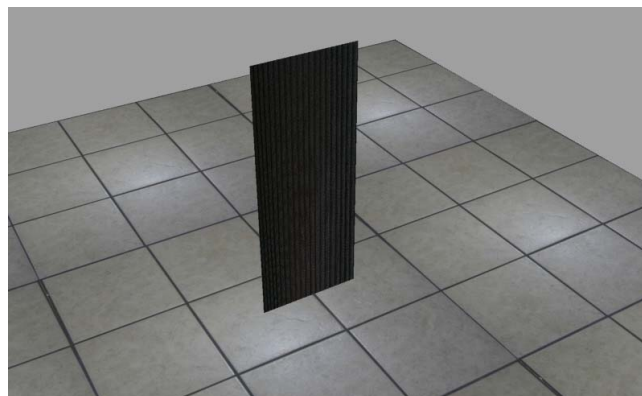
4.5.2 Vertical Drape

A piece of cloth of dimension $0.2\,m \times 0.5\,m$ drapes under its own weight to the ground , as shown in Fig. 3.6. Material properties of the cloth are taken to be $E = 30000\,Pa$, $\nu = 0.3$, $B_0 = 1 \times 10^{-6}\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The fabric thickness is $h = 0.002\,m$, and the surface density is $\rho = 0.144\,kg/m^2$. A linear damping force with damping constant $\eta = 0.288\,kg/(m^2 \cdot s)$ is included. The contact layer thickness is assumed to be $\delta = h$. The friction coefficient is $\mu = 0.2$. The cloth is represented by a second order NURBS patch with 612 control points. The time increment Δt at the beginning of the simulation is about $0.0003\,s$, and it is adjusted automatically as the cloth deforms.

Snapshots of the draping process are presented in Fig. 4.10. The wrinkles appear vivid and visually appealing, and no penetration is found between the cloth and the ground or the cloth itself in the whole simulation process. The CPU time for the whole simulation is 103 seconds. The simulation results are rendered using Rhinoceros, a NURBS based CAD software.

4.5.3 Untangle layers of Soft Armors

This sections shows an example of practical multi-layer soft armors. This set of armors contains three layers, as shown in Fig. 4.11(a). The first layer is a pair short pants, the second layer is a waist protector, and the third layer is a shirt. The try-on and draping simulation of each layer is performed separately, and when the three layers of armors are placed in the same space, we find severe self-intersections,

0ms
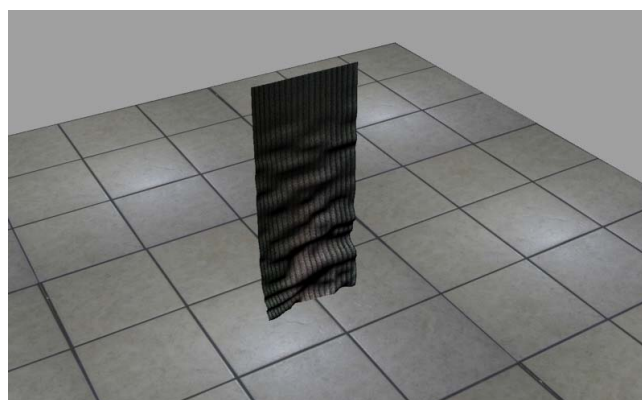
120ms
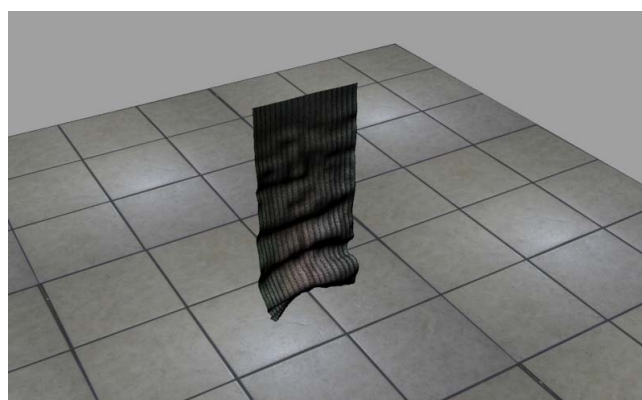
180ms

Figure 4.10: Vertical drape

240ms

300ms

360ms

Figure 4.10: Continued

420ms

480ms

540ms

Figure 4.10: Continued

600ms

660ms

720ms

Figure 4.10: Continued
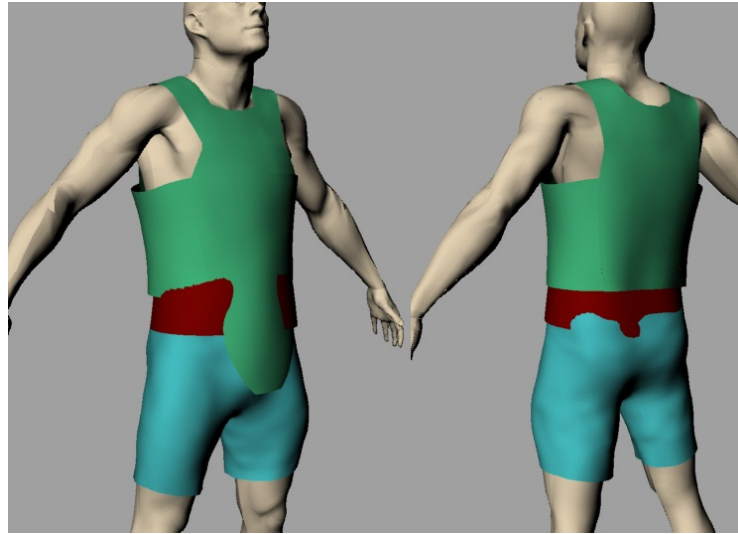
(a) After try-on and draping simulation
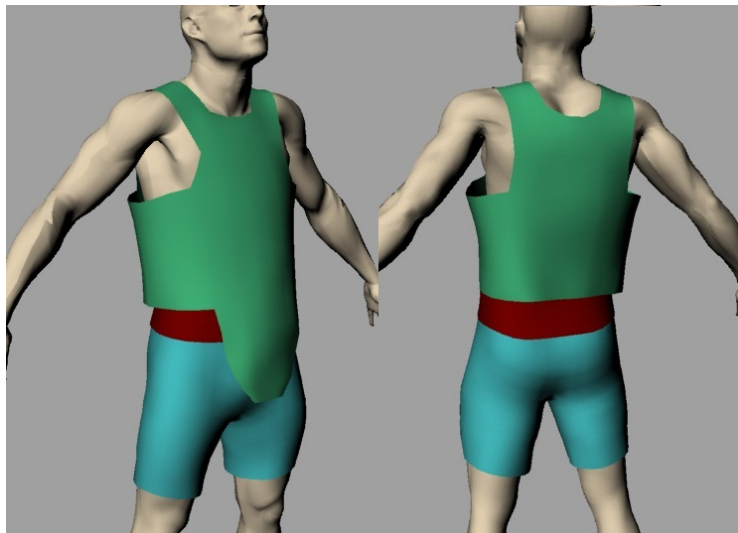
(b) Untangled armors

Figure 4.11: Simulation of multi-layer soft armors.

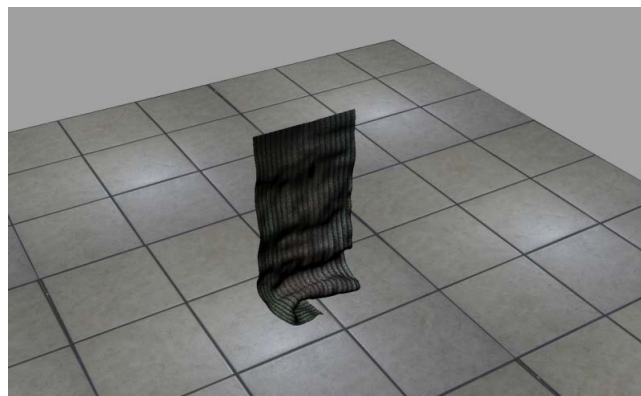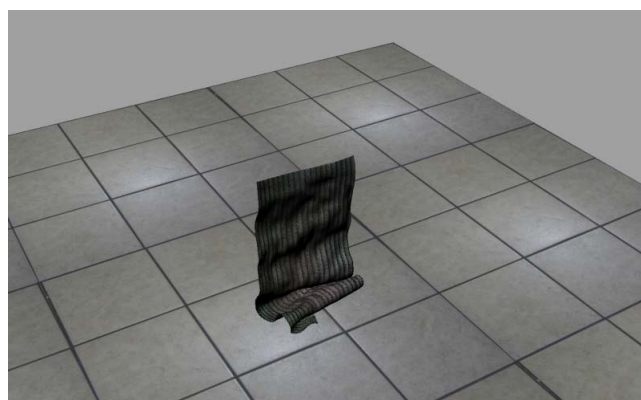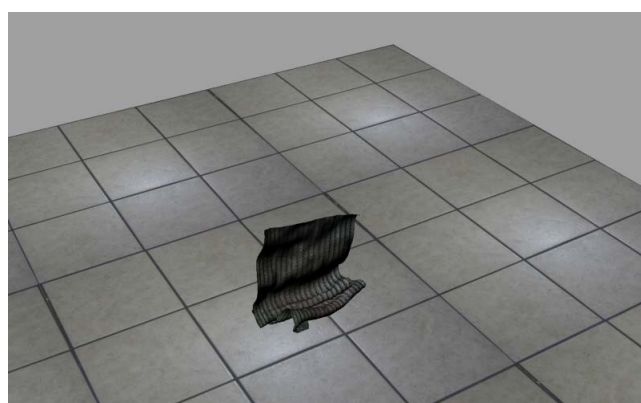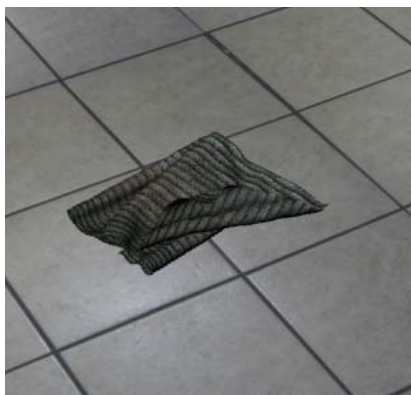as shown in Fig. 4.11(b). Then we use the present scheme to untangle the armors and place the three layers in the correct order. The final simulation results are shown in Fig. 4.11(c). It is shown that even for complicated intersections, the present method can still untangle the cloth efficiently.

The material property of this example is taken as follows: The volume density is $\rho = 1440\,kg/m^3$. In-plane stiffness is: $E = 1.44 \times 10^6\,Pa$, $\nu = 0.3$; bending rigidity is: $B_0 = 0.00371\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The damping coefficient is: $\eta = 4.0$.

Table 4.1 shows the average time cost per time step. It is found that the cost of intersection response accounts for no more than 10% in the total cost of intersection resolution. Compared with the contour minimization method (global version), the area minimization method only changes the way of contact response, therefore the extra cost of the present method should even lower than 10%. Actually the time cost of intersection response is ignorable compared with the total time cost of a time step.

Table 4.1: Average CPU time cost

Job	Time cost
Total	145.6 ms
Intersection resolution	18.9 ms
Intersection detection	17.5 ms
Intersection response	1.8 ms

CHAPTER 5
CONTINUUM-BASED STRAIN LIMITING

5.1 Background of Strain Limiting

A challenge of cloth simulation is that in fabric materials, the material stiffness in the weft and warp directions is much higher than shear stiffness, and the later is much higher than bending stiffness. Numerically this leads to a stiff problem, which is difficult to solve. For Velocity Verlet method, the stable time step is determined by the highest stiffness, and thus steps have to be very small if the physical properties are used. The strain limiting technique attempts to use lower stiffness and thus larger time steps while limiting the cloth stretch. One way to work around is to formulate cloth as a constrained material (in-extensible or nearly in-extensible in the weft and warp directions). Classical Lagrangian multiplier method could also be computation costly because it requires implicit solution of multiplies and the linear system contains zero diagonal sub-matrix. Recent fast projection method [35], shows advance in solving such problems. The fast projection method is typically based on augmented Lagrangian formulation, with a special technique for solving the constraint equation. A continuum-based fast projection method in NURBS geometry is presented in this chapter.

The weak form of the momentum equation is given by

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{damp} + \mathbf{f}_c \tag{5.1}$$

in which $\mathbf{M}, \mathbf{f}_{int}, \mathbf{f}_{ext}, \mathbf{f}_{damp}$ is defined in Eq.(2.22) and $\mathbf{f}_c$ is the contact force. Let's

omit the contact force in this chapter because the contact force can be singled out as the following substep. The momentum equation without contact is written as

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{damp} \tag{5.2}$$

Eq.(5.2) is equivalent to a Lagrange problem with the potential function given by

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}} - V(\mathbf{q}) \tag{5.3}$$

and V is the total potential energy given by

$$V(\mathbf{q}) = \int_{\Omega} \frac{1}{2}\mathbf{E} \cdot \mathbb{D}\mathbf{E}hd\Omega + V_{bending} + V_{external} \tag{5.4}$$

Now we consider the weft and warp response as continuum constraints, which implies the energy terms of weft and warp strain are replaced by constraint conditions $E_{warp} = 0$ and $E_{weft} = 0$.

Numerically, we will enforce the conditions discretely at certain points only. The constraint vector $\mathbf{C}$ represents constraint conditions of all constraint points, given by

$$\mathbf{C} = (\mathbf{C}_1, \mathbf{C}_2, ...\mathbf{C}_m)^T, \tag{5.5}$$

and

$$\mathbf{C}_i = \left[\ E_{weft}(\mathbf{x}_i), E_{warp}(\mathbf{x}_i) \ \right] \tag{5.6}$$

where m is the total number of constraint points. We will introduce the algorithm of solving the constrained problems in general case first, and the selection of constraint points in NURBS geometry will be discussed later in Section 5.4.

In the context of spring-based model, the constraints on strain are replaced with the length change of springs. Thus, the grid lines are required to align with the weft and warp of fabric.

5.2   Augmented Lagrange Method

Consistent with the standard augmented Lagrangian method, we replace the weft and warp stiffness with a small value rather than remove the energy terms totally. The problem now becomes a augmented Lagrangian multiplier problem. For general coordinates $\mathbf{r} = (\mathbf{q}^T, \boldsymbol{\lambda}^T)$, the potential function is

$$L(\mathbf{r}, \dot{\mathbf{r}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} - V(\mathbf{q}) - \mathbf{C}(\mathbf{q})^T \boldsymbol{\lambda} \tag{5.7}$$

and

$$V(\mathbf{q}) = \int_\Omega \frac{1}{2} \mathbf{E} \cdot \bar{\mathbb{D}} \mathbf{E} h d\Omega + V_{bending} + V_{external} \tag{5.8}$$

where the weft and warp stiffness in $\bar{\mathbb{D}}$ is much smaller than $\mathbb{D}$.

The Euler-Lagrange equation is given by

$$\frac{\partial L}{\partial \mathbf{r}} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{r}}} = 0 \tag{5.9}$$

Substituting Eq.(5.7) into Eq.(5.9) yields

$$\mathbf{M}\ddot{\mathbf{q}} = -\frac{\partial V}{\partial \mathbf{q}} - \frac{\partial \mathbf{C}^T}{\partial \mathbf{q}} \boldsymbol{\lambda} \tag{5.10a}$$

$$\mathbf{C} = \mathbf{0} \tag{5.10b}$$

Suppose we check the constraint conditions at the end of timestep: $\mathbf{C}(\mathbf{q}^{n+1} =$

0), and use Velocity Verlet method for the time integration of $\frac{\partial V}{\partial \mathbf{q}}$, we have

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \left(\dot{\mathbf{q}}^n - \frac{\Delta t}{2} \mathbf{M}^{-1} \frac{\partial V(\mathbf{q}^n)}{\partial \mathbf{q}} \right) - \Delta t^2 \mathbf{M}^{-1} \left(\frac{\partial \mathbf{C}(\mathbf{q}^{n+1})^T}{\partial \mathbf{q}} \boldsymbol{\lambda}^{n+1} \right) \quad \text{(5.11a)}$$

$$\mathbf{C}(\mathbf{q}^{n+1}) = \mathbf{0} \quad \text{(5.11b)}$$

Eq.(5.2) can be computed in two steps, given by

1. Introduce the intermediate position $\tilde{\mathbf{q}}^{n+1}$ given by

$$\tilde{\mathbf{q}}^{n+1} = \mathbf{q}^n + \Delta t \left(\dot{\mathbf{q}}^n - \frac{\Delta t}{2} \mathbf{M}^{-1} \frac{\partial V(\mathbf{q}^n)}{\partial \mathbf{q}} \right) \quad \text{(5.12)}$$

2. Calculate the actual position by $\mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n+1} + \Delta \mathbf{q}$, and $\Delta \mathbf{q}$ is solved from

$$\Delta \mathbf{q} = -\Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{C}(\mathbf{q}^{n+1})^T}{\partial \mathbf{q}} \boldsymbol{\lambda}^{n+1} \quad \text{(5.13a)}$$

$$\mathbf{C}(\mathbf{q}^{n+1}) = 0 \quad \text{(5.13b)}$$

Eq.(5.13a) and (5.13b) are equivalent to $\delta W = 0$, where

$$W = \frac{1}{2\Delta t^2} \Delta \mathbf{q}^T \mathbf{M} \Delta \mathbf{q} + \mathbf{C}(\mathbf{q}^{n+1})^T \boldsymbol{\lambda}^{n+1} \quad \text{(5.14)}$$

The second term of W projects $\mathbf{q}$ into the constraint manifold, while the first term minimizes the change of kinematic energy in this projecting process. All together, $\mathbf{q}$ will be projected to the "closest" point on the constraint manifold.

Linearize Eq.(5.13a) and (5.13b) by the Newton method, we have the following flowchart:

$\Delta \mathbf{q}^{[0]} = 0$, $\boldsymbol{\lambda}^{[0]} = 0$, for j=0,1,2,... do

1. Solve $\Delta \mathbf{q}$ and $d\boldsymbol{\lambda}$ from

$$\begin{bmatrix} \mathbf{I} + \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]T} \boldsymbol{\lambda}^{[j]}}{\partial \mathbf{q} \partial \mathbf{q}} & \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]T}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} d\mathbf{q} \\ d\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\Delta \mathbf{q}^{[j]} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]T}}{\partial \mathbf{q}} \boldsymbol{\lambda}^j \\ -\mathbf{C}^{[j]} \end{bmatrix}$$

$$\text{(5.15)}$$

2. Correct $\mathbf{q}$ and $\boldsymbol{\lambda}$ by

$$\Delta\mathbf{q}^{[j+1]} = \Delta\mathbf{q}^{[j]} + d\mathbf{q}, \quad \boldsymbol{\lambda}^{[j+1]} = \boldsymbol{\lambda}^{[j]} + d\boldsymbol{\lambda} \tag{5.16}$$

The loop is terminated when the maximum absolute value $\mathbf{C}$ is smaller than given tolerance ε. We note here that we have to solve a $(3n + m)$-dimensional asymmetric linear system for each iteration.

5.3 Fast Projection Method

We note that Eq.(5.13a) attempts to do a closet-point projection and Eq.(5.13b) enforces the constraint conditions. In an actual problem, we have stronger requirement on constraint conditions while we can tolerate the end point not being a closet projection to the constraint manifold. Based on this premise, Goldenthal et al. [35] suggested solving Eq.(5.13a) by Euler method while solving Eq.(5.13b) by Newtom method. We expand $\mathbf{C}$ by Taylor series at $\mathbf{q}^{[j]}$ and ignore the quadratic and higher-order terms,

$$\mathbf{C}^{[j+1]} = \mathbf{C}^{[j]} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}}d\mathbf{q} \tag{5.17}$$

Substituting Eq.(5.17) into Eq.(5.13a) yields

$$d\mathbf{q} = -\Delta t^2\mathbf{M}^{-1}\frac{\partial \mathbf{C}^{[j]^T}}{\partial \mathbf{q}}d\boldsymbol{\lambda} \tag{5.18}$$

Eq.(5.13b) is still linearized by Newton method,

$$\frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}}d\mathbf{q} = -\mathbf{C}^{[j]} \tag{5.19}$$

Substituting Eq.(5.18) into Eq.(5.19) yields the basic equation of the fast pro-

jection method

$$\Delta t^2 \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}} \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}}^T d\boldsymbol{\lambda} = \mathbf{C}^j \qquad (5.20)$$

The iteration flowchart of fast projection is as follows:

$\Delta \mathbf{q}^{[0]} = 0$, for j=0,1,2,...

1. Solve $d\boldsymbol{\lambda}$ from

$$\Delta t^2 \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}} \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}}^T d\boldsymbol{\lambda} = \mathbf{C}^j \qquad (5.21)$$

2. Correct $\mathbf{q}$ by

$$\Delta \mathbf{q}^{[j+1]} = \Delta \mathbf{q}^{[j]} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{C}^{[j]}}{\partial \mathbf{q}}^T d\boldsymbol{\lambda} \qquad (5.22)$$

The loop is terminated when the maximum absolute value of $\mathbf{C}$ is smaller than given tolerance ε. The advantage of the fast projection method is that (1) we only need to solve am m-dimensional symmetric positive definite (S.P.D.) linear system for each iteration; (2) it converges faster.

5.4   Constraint Points

The selection of constraint points is very important to the success of the fast projection method. Too many constraints will cause kinematic locking, while too few constraints can not eliminate overstretch effectively.

We tried to select Gauss points as constraint points, but it didn't work well. The asymptotic ratio of the number of elements and that of the control points is 1. The scheme of 2×2 Gauss points causes kinematic locking because the asymptotic ratio of Gauss points to the number of control points is 4. The scheme based one Gauss point or average strain doesn't have locking, but both had the hourglass problem. The
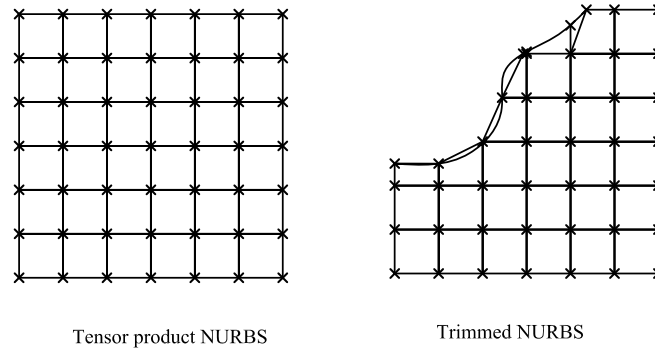
Figure 5.1: Constraint points

hourglass might be suppressed by adding a regulation term to the energy function, but we decided to pursue a simpler route. At the end, we employed the node-based constraints. For a regular NURBS mesh, we selected vertices of the knot mesh as constraint points, as shown in Fig. 5.1. For trimmed NURBS, we constrained the intersection points between the trimming curves and the knot mesh as well. The benefit of this scheme is that it doesn't have locking or hourglass, and it also leads to a smaller band width in the ensuing linear systems of fast projection method.

5.5   Stress Reconstruction

The constraint energy term $\boldsymbol{\lambda} \cdot \mathbf{C}$ in essence is the nodal integration form of strain energy. Thus the stress $\hat{\mathbf{S}}$ should include two terms: the stress from strain $\mathbf{S}(\mathbf{E})$ and the stress from constraint force. The stress on grid point i after fast projection is given by

$$\hat{\mathbf{S}}_i = \mathbf{S}_i(\mathbf{E}_i) + \boldsymbol{\lambda}_i/a_i \tag{5.23}$$

where a_i is the nodal area.

However we found checkerboard pattern in the values of $\boldsymbol{\lambda}$. This is mainly due to the lack of domain integration. Introduce the integration version of constraint energy may resolve the problem, but we believe that is to costly as the band width will be heavily enlarged. So we introduce a simple way to reconstruct the stress: by an interpolation, it is easy to obtain the stress on Gauss points, and then by an extrapolation fromm Gauss points, we can obtain the nodal stress without checkerboard problem.

5.6 Examples

Several examples are presented in this section to demonstrate the performance of the proposed method. The first is a benchmark testing the capability of handling overstretch. The second and the third concern two garment-level simulation. In order to evaluate the method, we considered three cases for each example: (a) small stiffness with strain limiting, (b) high stiffness without strain limiting, and (c) small stiffness without strain limiting. The simulation efficiency is evaluated by the average CPU-time cost of each frame, and one second of animation contains 50 frames.

5.6.1 Corner Kidnapped Cloth

A piece of cloth initially in the horizontal x-y plane is fixed at two corners and will swing under gravity in z-direction. The cloth is represented by a second order NURBS patch with 100 control points. The bending parameters are $B_0 = 3.3 \times 10^{-3} \, N \cdot m$, $\kappa_0 = 30 \, m^{-1}$. The mass density is $\rho = 0.117 \, kg/m^2$ and damping constant is $\eta = 0.351 \, kg/(m^2 \cdot s)$. Fabric thickness is $h = 0.001177 \, m$. For the in-plane

model, we assume $k = k_{weft} = k_{warp} = 2.5k_{shear}$ and test three cases: (a) $k/\rho = 100$ with strain tolerance $\varepsilon = 0.01$; (b) $k/\rho = 5000$ without strain limiting; (c) $k/\rho = 100$ without strain limiting.

The simulation results of all cases at $t = 0.6\,s$, when the cloth passes the vertical plane for the first time, are shown in Fig. 5.2. The simulation time and strain summary is shown in Table 5.1. It is observed that case (c) has obvious unrealistic stretching. The results of case (a) and case (b) are close, but the time increment of case (a) is much smaller than that of case (b). Because there is no contact involved, the saving from large time step is not very obvious, but still strain limiting obtains 2.5 times of speeding up.

Table 5.1: Summary of corner kidnapped cloth simulation

	Case (a)	Case (b)	Case (c)
CPU time (sec/frame)	0.13	0.325	0.054
Maximum E_{weft}	0.015	0.0828	2.24
Maximum E_{warp}	0.013	0.0512	2.20
Average E_{weft}	9.61×10^{-4}	4.07×10^{-3}	0.081
Average E_{warp}	1.30×10^{-3}	6.78×10^{-3}	0.111

5.6.2 Draping of Soft Armor

The draping of a soft armor is simulated in this example. The armor is represented by a second order NURBS patch with 1616 control points. The initial configuration is obtained by a virtual try-on simulation and is shown in Fig. 5.3. The
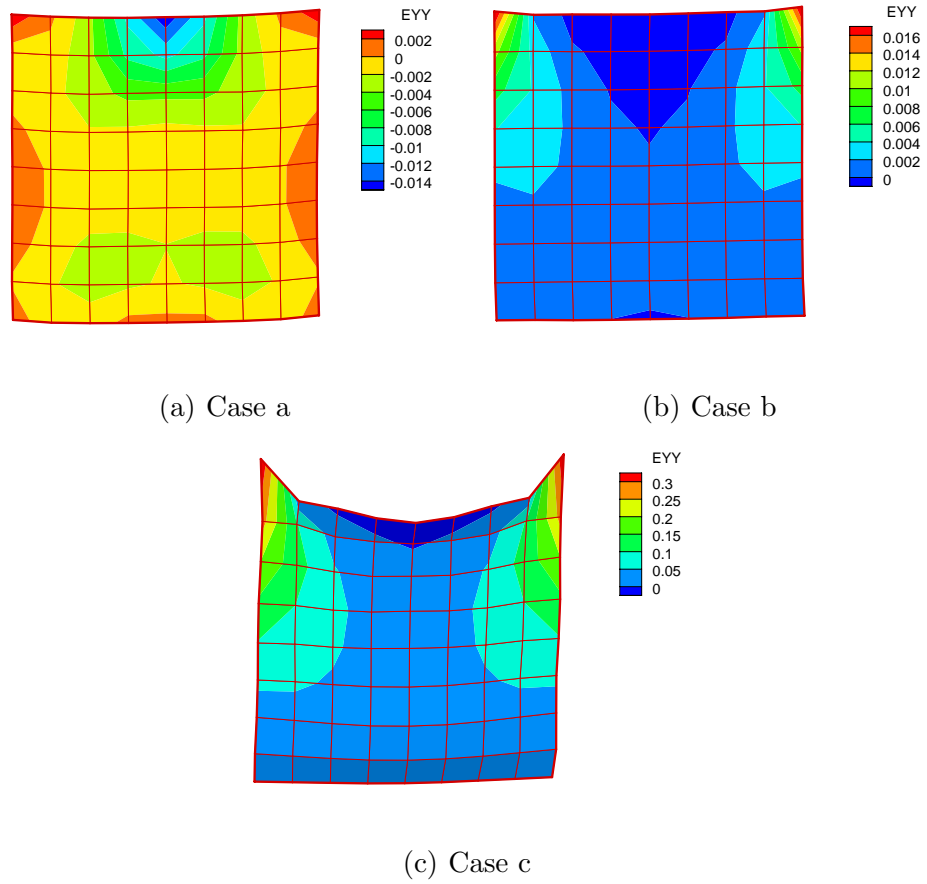
(a) Case a

(b) Case b

(c) Case c

Figure 5.2: Corner kidnapped cloth at $t = 0.6$.

bending parameters are $B_0 = 1 \times 10^{-4} \, N \cdot m$, $\kappa_0 = 30 \, m^{-1}$. The mass density is $\rho = 0.1177 \, kg/m^2$, and the damping constant is $\eta = 0.353 \, kg/(m^2 \cdot s)$. Fabric thickness is $h = 0.001177 \, m$. The friction coefficient is assumed to be $\mu = 0.2$. For the in-plane model, we assume $k = k_{weft} = k_{warp} = 2.5 k_{shear}$ and create three cases: (a) $k/\rho = 50$ with strain tolerance $\varepsilon = 0.01$; (b) $k/\rho = 5000$ without strain limiting; (c) $k/\rho = 50$ without strain limiting.

In order to show the results more clearly, the simulation results of upper-body and lower-body armor at $t = 0.5 \, s$ are shown separately in Fig. 5.4 and Fig. 5.5, respectively. The simulation time and strain results are summarized in Table 5.2. For the upper-body armor, case (c) obtains an unacceptable shape while for the lower body armor, and all three schemes obtain an acceptable shape. By checking the maximum and average strain, it is found that both case (a) and case (b) obtain low level of strain, but the time cost of case (b) is four times of that of case (a). This example demonstrates that the strain limiting method can maintain a low strain level while reducing the CPU time.

Table 5.2: Summary of soft armor draping simulation

	Case (a)	Case (b)	Case (c)
CPU time (sec/frame)	5.69	25.25	3.47
Maximum E_{weft}	6.61×10^{-3}	0.097	0.41
Maximum E_{warp}	0.018	0.075	0.34
Average E_{weft}	2.15×10^{-4}	5.22×10^{-3}	0.037
Average E_{warp}	1.91×10^{-4}	3.02×10^{-3}	0.043

Figure 5.3: Initial configuration of the soft armor

5.6.3 Draping of a Skirt

This example simulates the draping process of a skirt. The initial configuration of the skirt is shown in Fig. 5.6. The entire model contains 960 control points. The woman's body is represented by a discrete mesh of 17068 cells. The bending parameters are $B_0 = 1.0 \times 10^{-5} N \cdot m$ and $\kappa_0 = 30 m^{-1}$. The mass density is $\rho = 0.118 kg/m^2$, and the damping constant is $\eta = 0.354\, kg/(m^2 \cdot s)$. The fabric thickness is $h = 0.001177 m$. A friction coefficient $\mu = 0.8$ is assumed. For the in-plane model, we assume $k = k_{weft} = k_{warp} = 2.5 k_{shear}$ and create three cases: (a) $k/\rho = 50$ with strain tolerance $\varepsilon = 0.01$; (b)$k/\rho = 3000$ without strain limiting; (c) $k/\rho = 50$ without strain limiting.

The garment shape at $t = 0.8\, s$, when the vibrations are mostly damped out, is
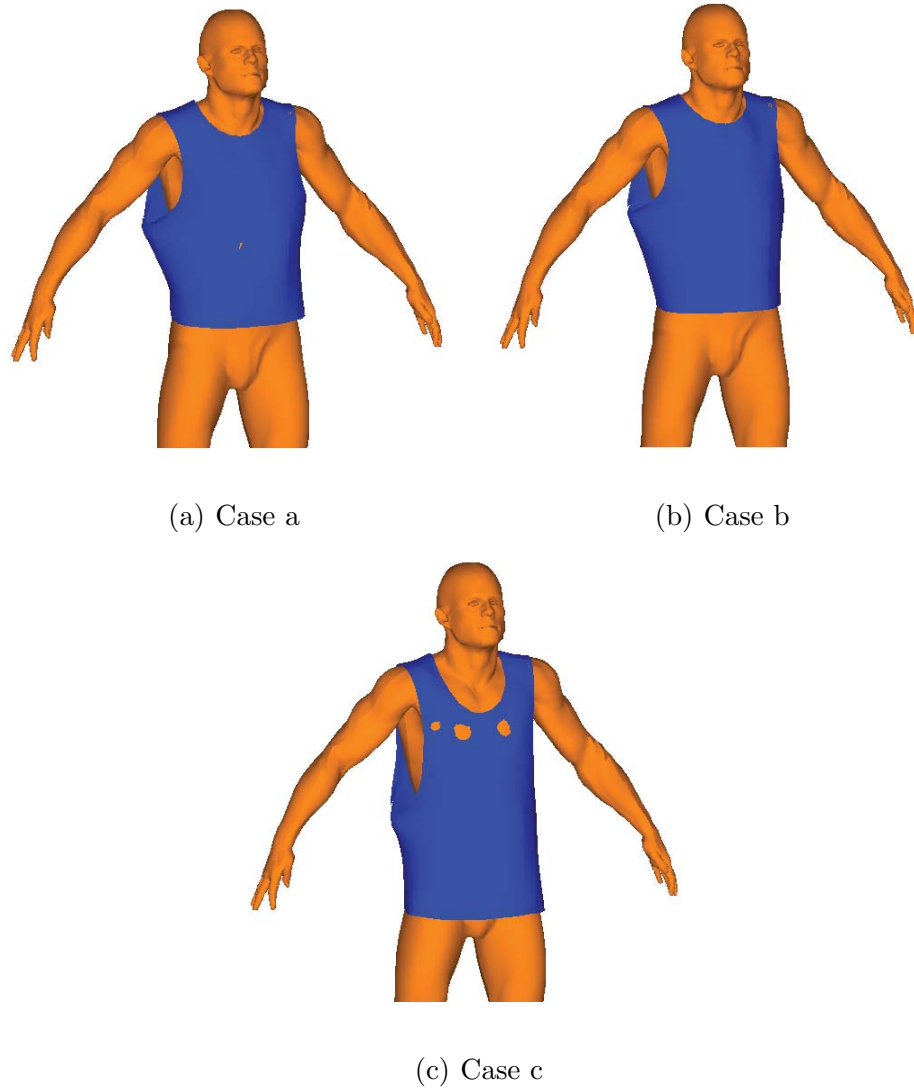
(a) Case a

(b) Case b

(c) Case c

Figure 5.4: Upper-body armor after draping simulation.

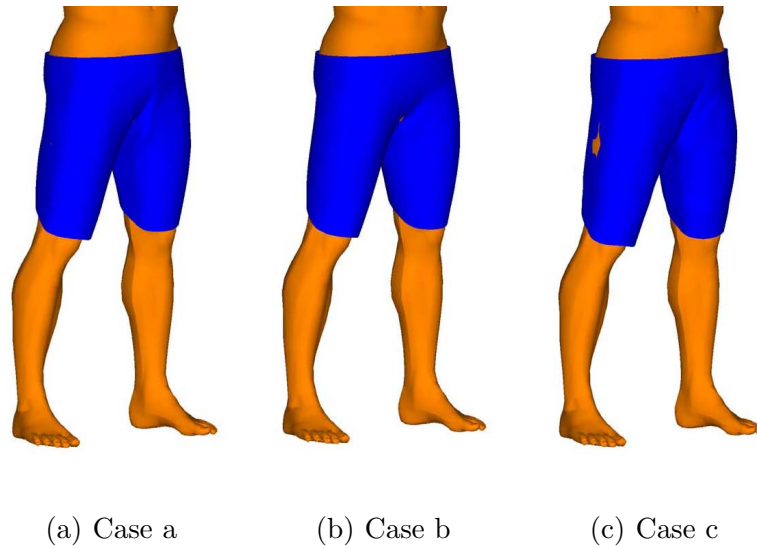(a) Case a   (b) Case b   (c) Case c

Figure 5.5: Lower-body armor after draping simulation.

reported in Fig. 5.7. The simulation time and strain summary is shown in Table 5.3. It is observed that the cloth is overstretched when low stiffness is used there is no strain limiting. When $E/\rho = 3000$ and strain limiting is turned off, the simulation shape looks visually better, but local overstretching is still very serious. We didn't try higher stiffness because it will take too much time. On the other hand, by using the strain limiting scheme with low stiffness, both the simulation shape and strain values satisfy the requirement while CPU time is reduced.

Figure 5.6: Initial configuration of the skirt

Table 5.3: Summary of skirt draping simulation

	Case (a)	Case (b)	Case (c)
CPU time (sec/frame)	7.4	20.4	8.2
Maximum E_{weft}	0.0093	0.26	0.64
Maximum E_{warp}	0.012	0.13	0.65
Average E_{weft}	4.12×10^{-4}	1.42×10^{-2}	0.073
Average E_{warp}	3.79×10^{-4}	1.04×10^{-2}	0.10

5.6.4 Patch Test

Finally we study the standard patch test to check the stress obtained from fast projection method. A 1 m by 1 m square is fixed on the top edge and applied 1 N/m of uniform force on the bottom edge. As the problem itself is based on dynamic analysis,
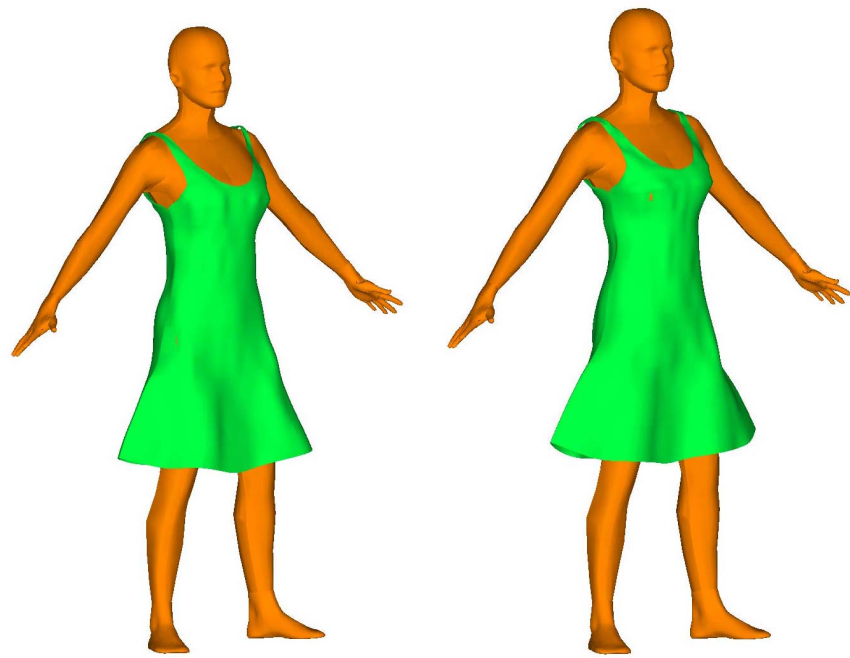
(a) Case a

(b) Case b

(c) Case c

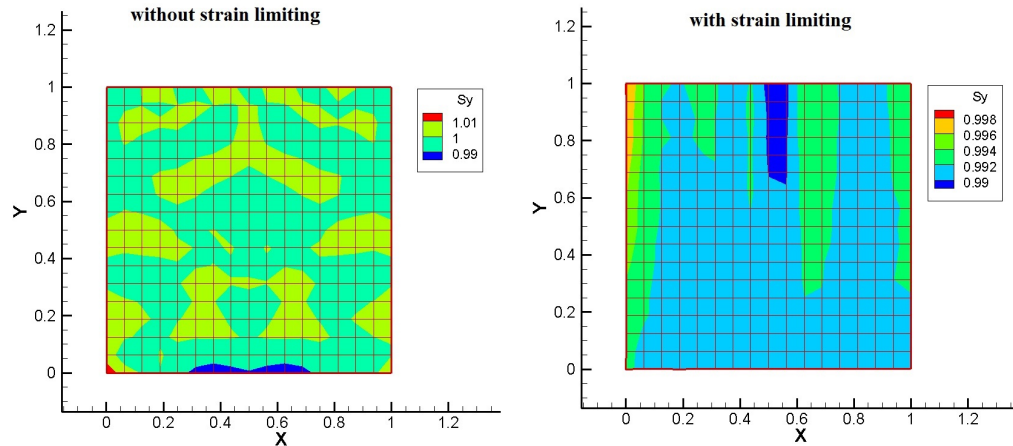Figure 5.7: Skirt after draping simulation.

Figure 5.8: Stress of patch test

we model the static patch test by applying a big damping factor and waiting until the vibration is damped out. Fig. 5.8 shows the static stress obtained from analysis with or without fast projection method. It is observed that for both cases, the stress are close to $1\,\mathrm{Pa}$, which means the fast projection method can not only eliminating unrealistic unrealistic strain, but also provide reliable stress field.

5.7   Conclusions

A continuum-based strain limiting method is developed in this chapter. The method is based on augmented Lagrangian method, with a fast projection scheme employed to solve the constraint equation. The constraint conditions are defined on the basis of continuum stains, and thus, the mesh doesn't need to be aligned with weft and warp of fabric. Examples show that small stiffness without strain limiting typically produces excessive stretches, as expected, and high stiffness requires much

longer simulation time. The proposed strain limiting method yields small stretch and visually realistic deformation with reduced CPU time.

CHAPTER 6
SIMULATION OF SOFT ARMORS

6.1   Integration with Digital Human Body

The NURBS-based cloth simulation was integrated to a digital human environment, Santos^{TM}. Santos is a virtual environment for physics-based digital human modeling and simulation. It has a real human-like appearance with deformable skin, optimization-based posture, and dynamic motion prediction capabilities. This environment can give different types of feedback, such as whether Santos can reach its target, what the discomfort level is in such a case, and the joint actuator torques in all joints involved in performing a task [94].

Fig. 6.1 shows the work flow between CAD program, digital human modeling and cloth simulation. The CAD programs provide design models in NURBS format, and the digital-human-body program provides a series of snapshots of human bodies. Additional information for simulation, such as load, material properties, etc., is de-
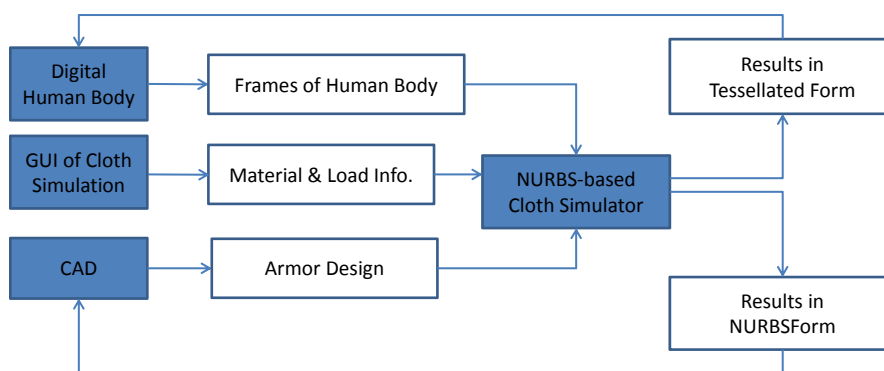

Figure 6.1: Work flow of cloth simulation

fined in a GUI of cloth simulation. For convenience of rendering in different software, we export each of the simulations results in two different versions: tessellated form, and NURBS form.

6.2   Description of Soft Armor

In this study, we use a soft armor provided by company "Crye Precision" as example, as shown in Fig. 6.2(a). In order to perform analysis on the NURBS models, we need some "clean-up" work. First of all, the designing end and the analysis end have different requirements on the grid density. For a design model, very fine mesh is used in the region with complicated geometry, while very coarse mesh is used in the region close to plane. For an analysis model, we require the finer mesh locates on the region with more complicated mechanics behaviors, such as stress concentration, and we don't want the mesh size varies too much within the same model. In addition, when the model contains more than one pathes, analysis models require a compatible mesh between the neighboring pathes, while the designers don't care such issues.

An analysis-ready CAD model is a set NURBS geometries that satisfy the requirements of isogeometric analysis while keeping the original geometry intact or nearly intact. We convert a design model to an analysis-ready CAD model by using the edit tools in CAD program, like Rhinoceros. The major work of such change is to modify the mesh density and the locations of NURBS isocurves. The analysis-ready CAD model of the original design is shown in Fig. 6.2(b).

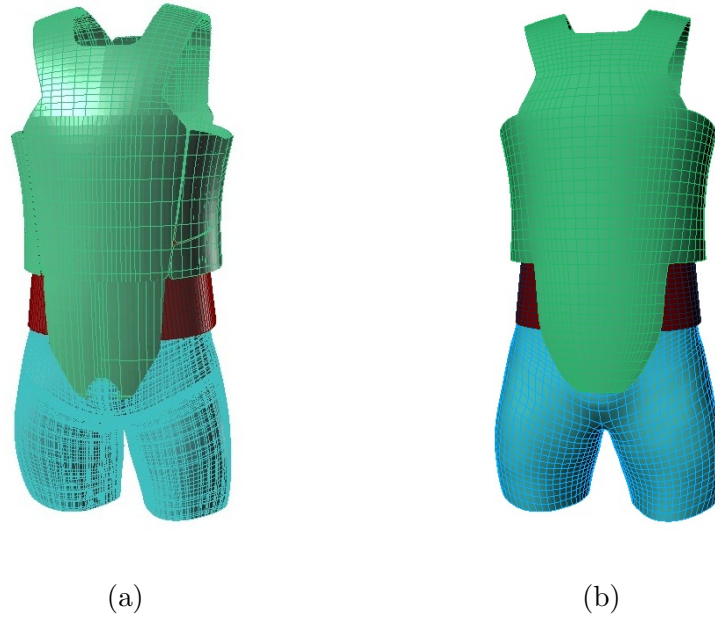The armor contains 3 layers. The first layer is a pair of short pants, and there

Figure 6.2: A set of soft armor. (a) Original design; (b) analysis-ready model.

are 6 patches. The design thickness of this layer is $0.762\,mm$. The second layer is a

belt, which is composed of a single patch. Its design thickness is $2.29\,mm$. The third

layer is a shirt-like shape plus a belly protector. This layer contains 6 patches. The

whole armor is represented by 2nd order NURBS surfaces, with 6308 control ponits

and 1 trimmed edge. The connection between different patches is achieved through

our multi-patch model, as discussed in Chapter 2.

The material property of the soft armor in this chapter is taken as follows: The

volume density is $\rho = 1440\,kg/m^3$. In-plane stiffness is: $E = 1.44 \times 10^6\,Pa$, $\nu = 0.3$;

bending rigidity is: $B_0 = 0.00371\,N \cdot m$, $\kappa_0 = 30\,m^{-1}$. The damping coefficient is:
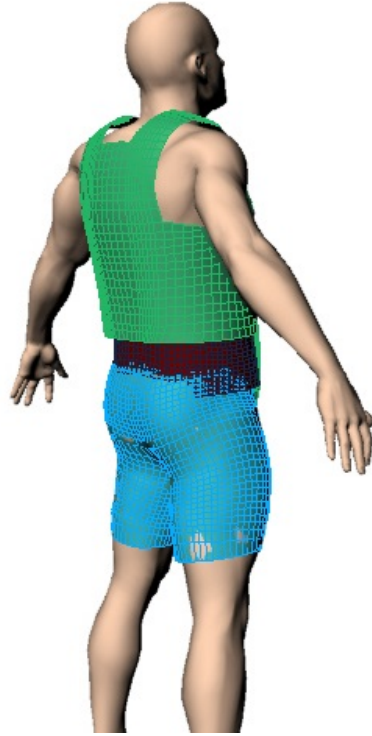
$\eta = 4.0$.

Figure 6.3: Armor model before try-on simulation

6.3 Virtual Try-on of Soft Armors

Because the armor design model and the digital human body come from different systems, they may intersect with each other when they are initially placed in the same space. The armor can be put on the digital human body by a try-on simulation. There are two kinds of virtual try-on scheme: the virtual stitching and the virtual projecting.

Virtual stitching contains three steps. First, the garment patch are placed around the human body. And then introduce a spring force between the edge pair which needs to be stitched. Finally perform a simulation and the stitching spring will

put the garment on the human body.

The way to define the stitching force is not unique, and we only introduce the linear model. Suppose a pair of edge points A and B should be stitched, and then the stitching force is given by

$$\mathbf{f}_A = k(\mathbf{x}_B - \mathbf{x}_A) - \eta v_A \tag{6.2}$$

$$\mathbf{f}_B = k(\mathbf{x}_A - \mathbf{x}_B) - \eta v_B \tag{6.3}$$

where k and η are user-defined constants.

The virtual projecting scheme is very simple. We scale and translate the human body until the garment does not interference with the body. Then during a simulation process, we linearly project the transformed human body to its original geometry while letting the cloth to interact with the body. When the process ends, the garment is put on the body.

When the original design in Fig. 6.2 and the human body were placed in the same space, lots of overlapping between them were found, as shown in Fig. 6.3.

The try-on simulation of upper body armor was finished by the virtual stitching, as shown in Fig. 6.4, because we were able to place the patches in a penetration-free status. The shorts and the belt were put on by virtual stitching, as shown in Fig.6.5, because it was difficult to get a penetration-free status without morphing the human body. After try-on simulation of each layers, we used the intersection resolution method to resolve intersections between different layers. The armors after the try-on simulation are shown in Fig. 6.6.
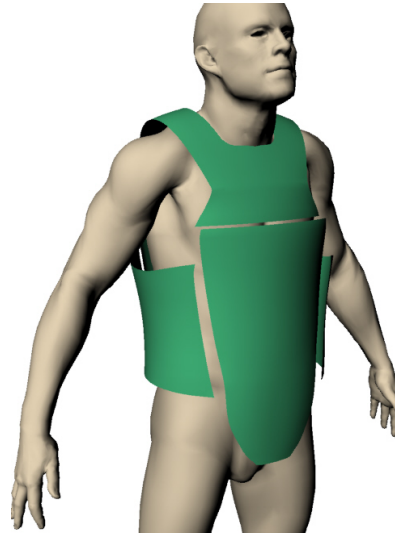
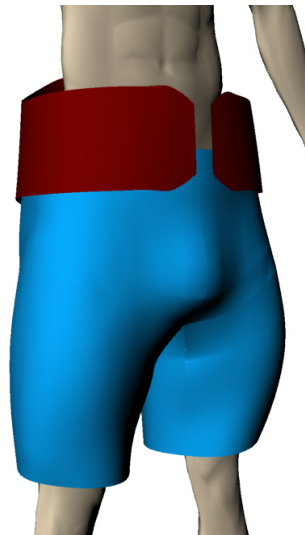Figure 6.4: Put the shorts and belt n by virtual stitching

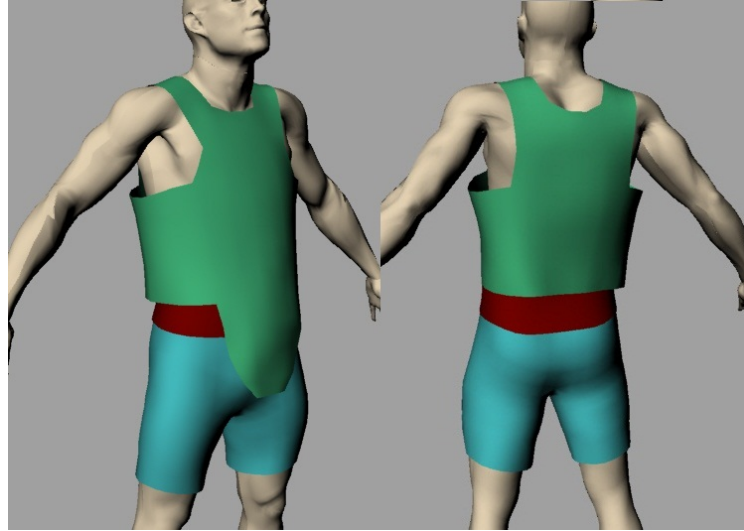Figure 6.5: Put the upper-body armor on by virtual projecting

Figure 6.6: Virtually trying on.

6.4   Walking

Starting from the try-on configuration, we deform the human body according to the frames given by Santos. Typical snapshots of a pace are shown in Fig. 6.7. The first snapshot, Fig. 6.7(a), is the end of the last pace and the beginning the current pace. The distance between the two feet attains maximum at this moment. If the forward torque on the hip-thigh joint is defined as positive, the torque on left leg should be negative at this moment. Fig. 6.7(b) is taken when the right leg raises to highest point. The torque on the right leg is expected to reach peak at this moment. Fig. 6.7(c) is the end of current pace. The whole simulation contains 3 paces, and the other 2 paces are presented in the same way.

Fig. 6.7 also plots the contact forces as arrows. The directions of the arrows represent the directions of contact forces and the lengths of the arrows represents the

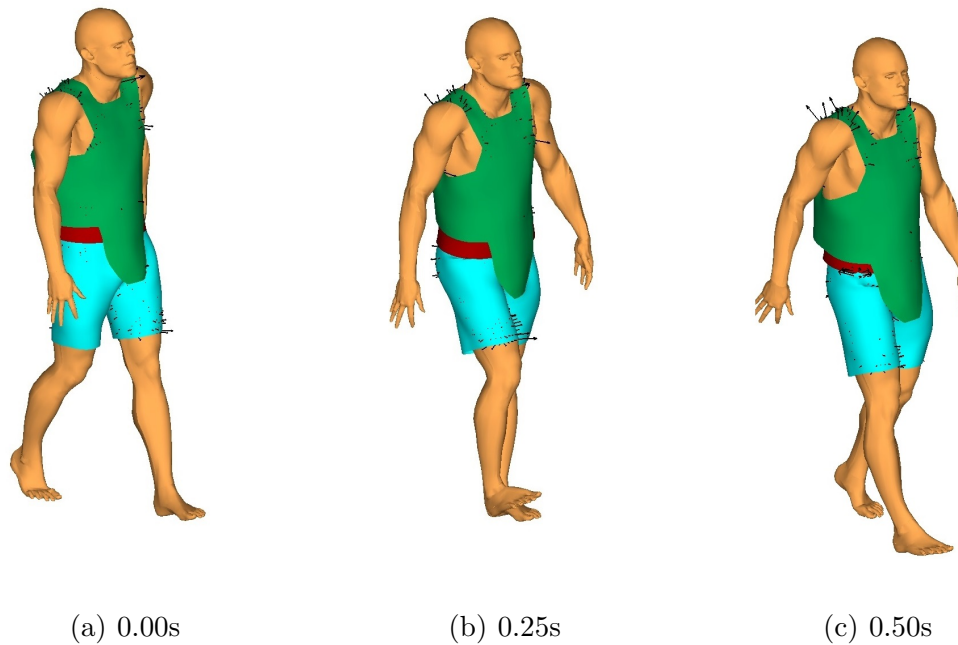(a) 0.00s   (b) 0.25s   (c) 0.50s

Figure 6.7: Contact forces of soft armor during walking

norms of the contact forces. It is observed that the contact forces mainly distribute on shoulder and thigh. The forces on the shoulder resist the gravity of the armor, while the forces on the thigh drive the pants to move forward or backward. We also find that the magnitudes of the contact forces on thighs are smaller or comparable with that on shoulders.

We are especially interested in the additional torque required by the soft armor on hip-thigh joints. The torque history is ploted in Fig. 6.8. The forward torque is defined as positive and the backward torque is defined as negative. Thus at the beginning, the torque of the left leg is positive and that of the right leg is negative. We find the torque on right leg arrives to peak at 0.25s, snapshot (b), when the right
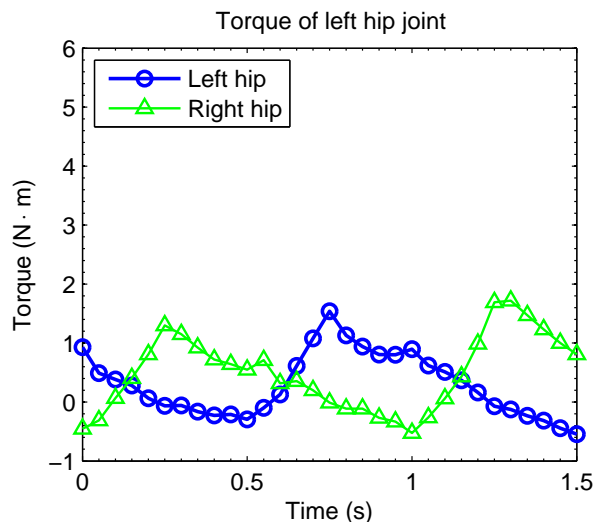
Figure 6.8: Torques on hip-thigh joints

leg is raised to the highest position. The torque history is periodical and symmetric, as the motion itself.

By measurement, we know that the weight of the upper body armor is $1.43\,kg$, and the width of shoulder band is $77\,cm$. Thus the approximate static stress in the shoulder area should be $91\,N/m$. The first principle stress (in-plane) is ploted in Fig. 6.9. In the simulation results, the dynamic stress in the shoulder area is around $100\,N/m$, except stress concentration part. The maximum stress roughly is in the range from 200 to 300 N/m, and the maximum stress appears in shoulder and pants. The additional stress is caused by the dynamic movement and the contact forces from human body.

We conducted a mesh-dependence study by evaluating the joint torques obtained from two meshes. We use two meshes: except the mesh in Fig. 6.2(b), we
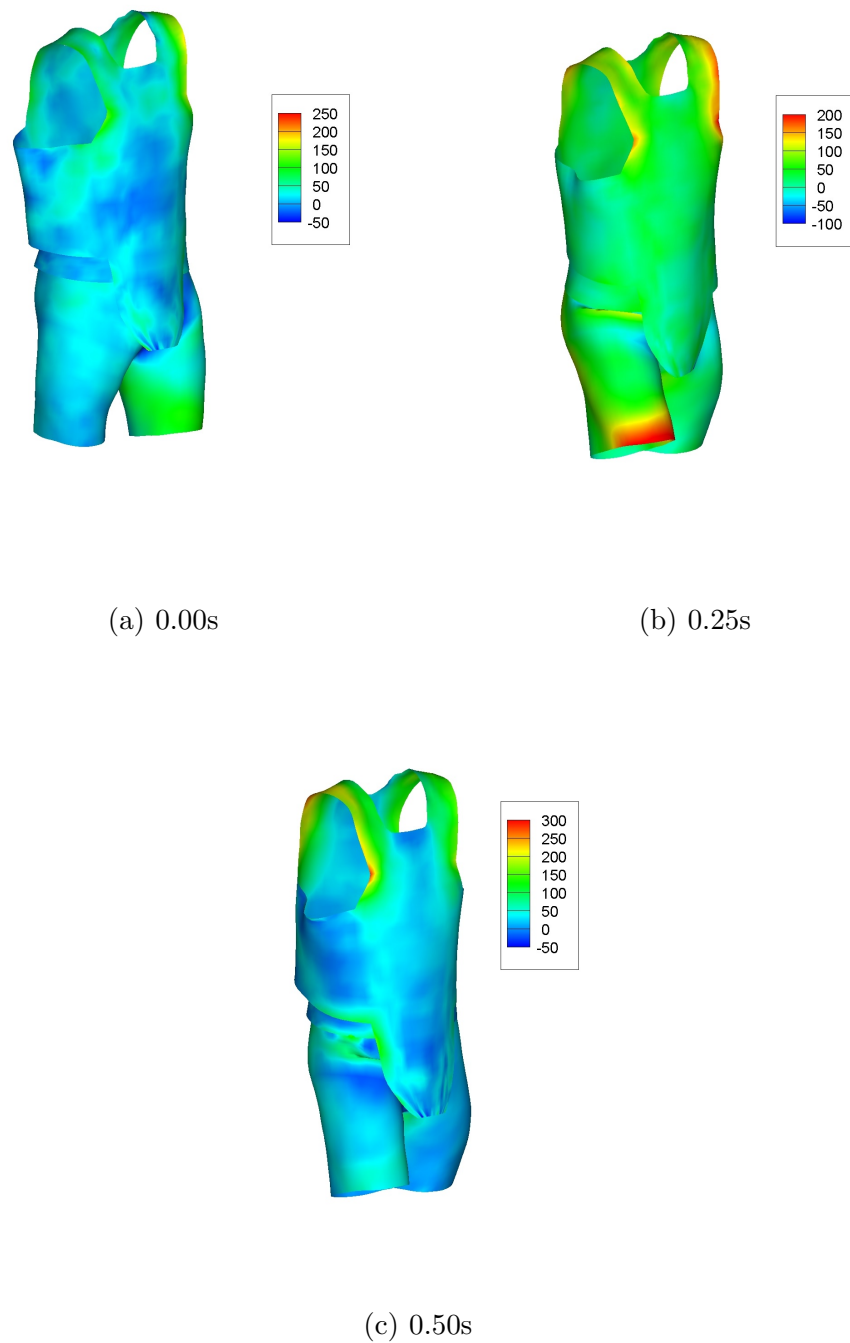
(a) 0.00s

(b) 0.25s

(c) 0.50s

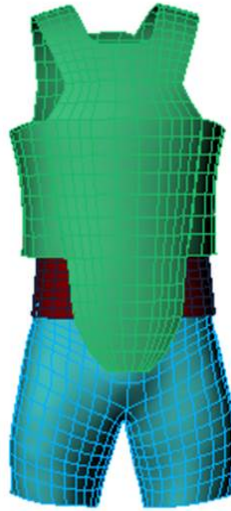Figure 6.9: Principle in-plane stress (N/m) of soft armor during walking

Figure 6.10: A coarse mesh for convergence study

created a coarse mesh with 1932 control points is shown in Fig. 6.10. Fig. 6.11 plots the torques of both meshes. Except the beginning time, the results of the fine and coarse mesh are close.

We also conducted the parametric study. The thickness of the armor is one of the most important design parameter, as it influences the comfortable level, the cost and the protection ability. Based on the original design, we cut the thickness by half, and we assume the bending rigidity is proportional to the cubic of thickness. Fig. 6.12 shows some snapshots of "thin" armor. First of all, we find the contact force reduce dramatically, compared with Fig. 6.7. We also find more wrinkles in the pants, which is the consequence of the reduction of bending rigidity.

Fig. 6.13 plots the comparison of torques of the two designs. We find the torques required by the armor are reduced when the thickness is halved, but the
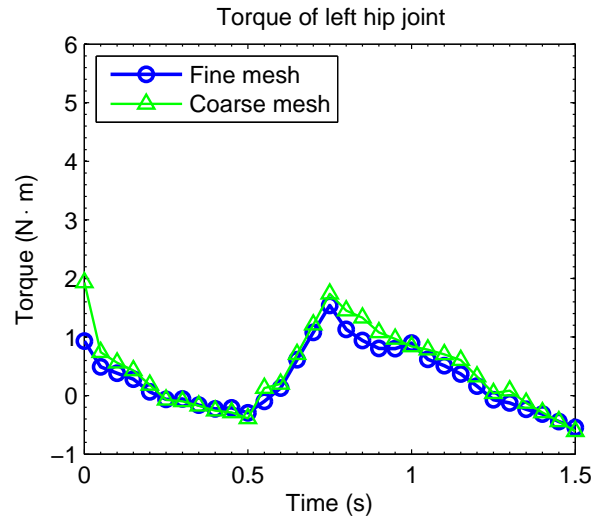
(a)
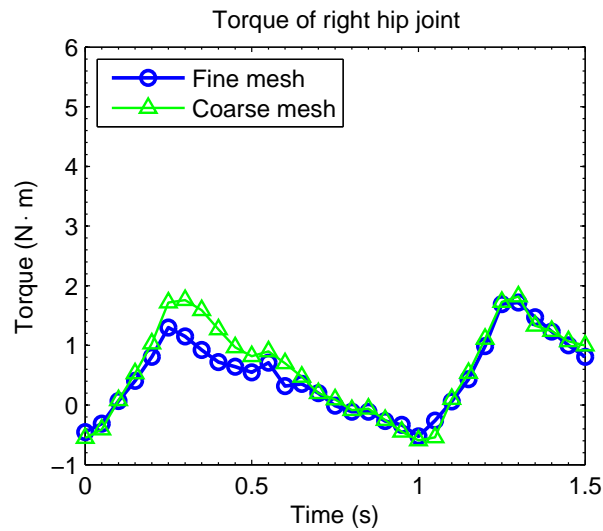
(b)

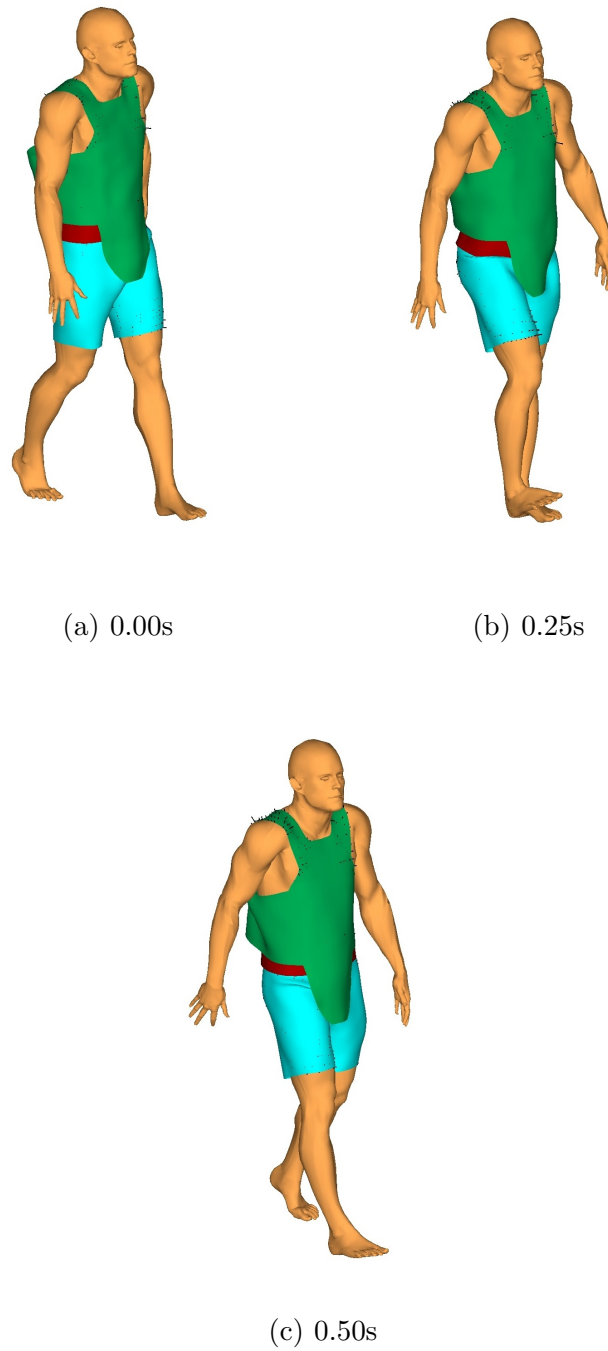Figure 6.11: Influence of mesh size to the torques on hip joints

(a) 0.00s   (b) 0.25s

(c) 0.50s

Figure 6.12: Contact forces of soft armor (1/2 design thickness) during walking

tendency remains similar. The peak values of the "thin" armor is about a third of the original design.

6.5   Stair Climbing

Starting from the try-on configuration, the soft armor is simulated when the human body climbs stairs. Fig. 6.14 plots the beginning and the end of a pace. The first snapshot, Fig. 6.14(a) is taken when the left leg raises to a higher step. At this time the torque on the left leg should be the maximum, while the torque on the right leg should be very small. Fig. 6.14(b) is taken when the right leg raises to a higher step. At this time the torque on the right leg should be the maximum, while the torque on the left leg should be very small. The whole simulation contains 2 paces, and the configuration at 1.2s is same as the beginning time.

Fig. 6.14 also plots the contact forces as arrows. It is observed that the contact forces mainly distribute on shoulders and thighs. The forces on the shoulders resistant the gravity of the armor, while the forces on the thighs drive the pants to move forward or backward. We also find that the magnitudes of the contact forces on thighs are a little bit higher than that on shoulders. The contact forces on legs during climbing stairs are higher than that during walking, because climbing stairs needs to bend the pants more severely.

The first principle stress (in-plane) is ploted in Fig. 6.15. In the simulation results, the dynamic stress in the shoulder area is around $100\,N/m$, except stress concentration part, comparable to the static stress. The maximum stress is about
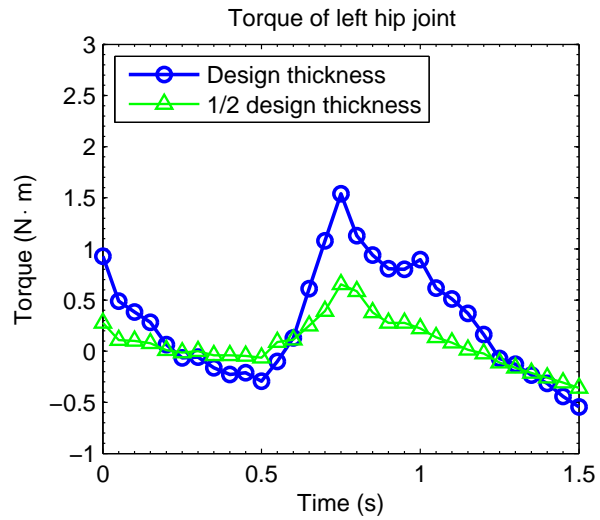
(a)
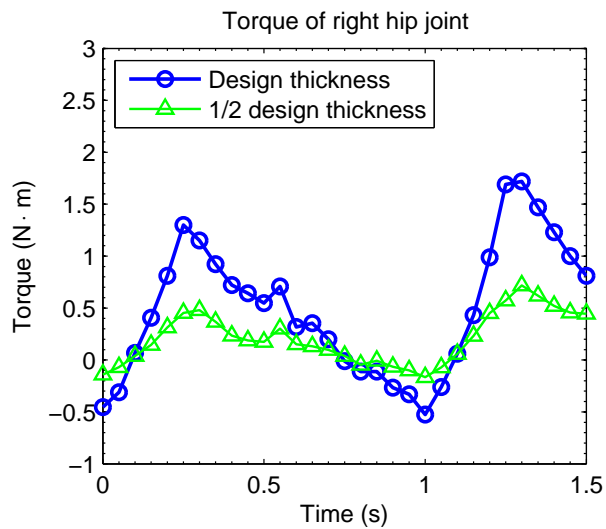
(b)

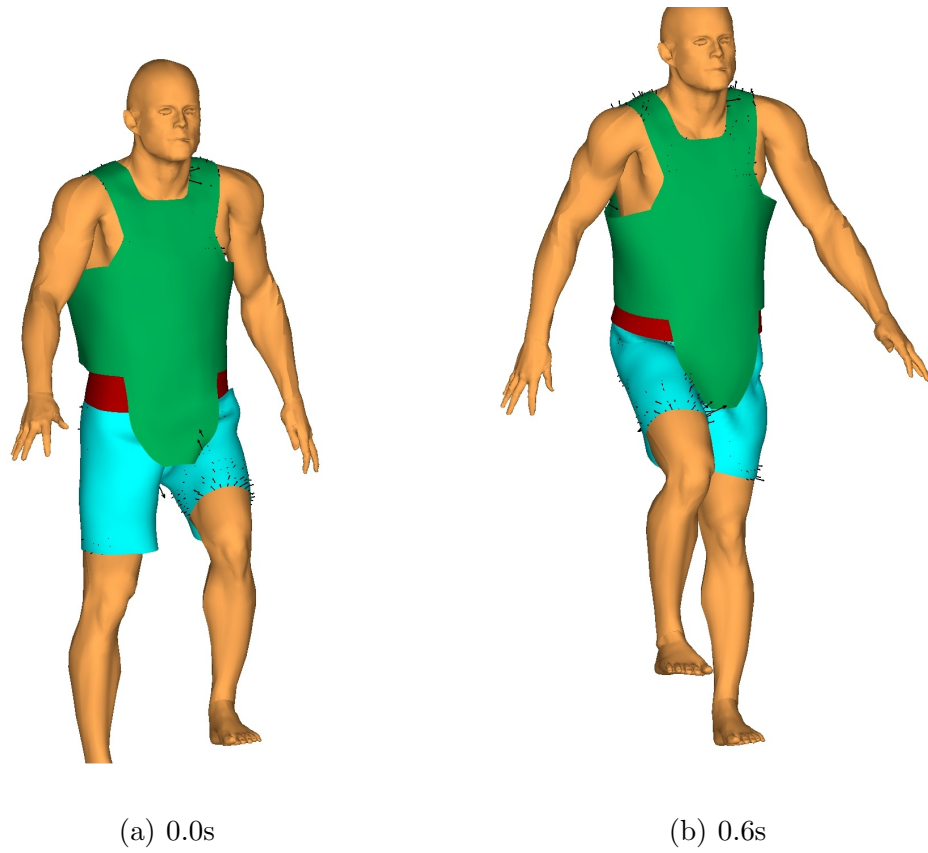Figure 6.13: Influence of thickness to the torques on hip joints

(a) 0.0s (b) 0.6s

Figure 6.14: Contact forces of soft armor during climbing stairs

(a) 0.0s   (b) 0.6s
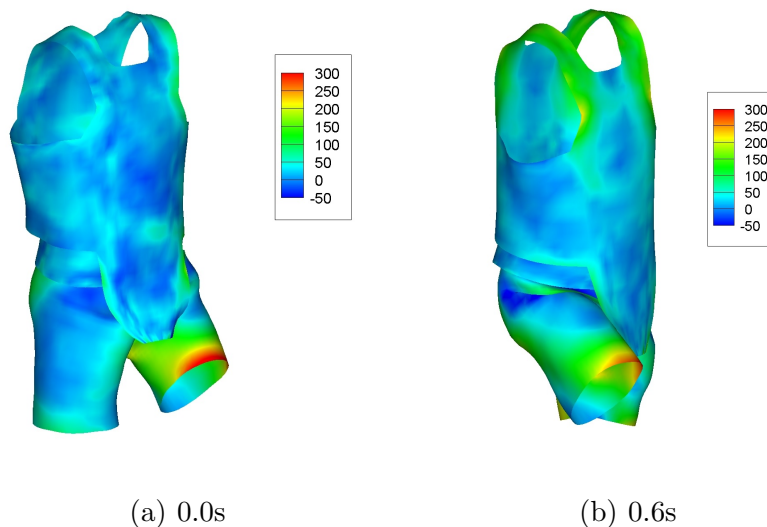
Figure 6.15: Principle in-plane stress (N/m) of soft armor during climbing stairs

300 N/m, and the maximum stress appears in the edge of the raised pant.

Fig. 6.16 plots the torque history of hip joints. At the end of each step, the torque on the raised leg reaches to peak value and the torque on the other leg approaches to zero, which agrees with our expectation. The torques at 1.2s are close to 0.0s, and a little bit bigger (absolute values). This is because the motion is periodical and the acceleration at 1.2s is bigger. Compared to walking, it is found that the peak value of stair climbing is higher, because the pants are bended more severely in stairs climbing.

We also conducted the parametric study. Again, the thickness is halved based on the original design. Fig. 6.17 shows some snapshots of "thin" armor. First of all, we found the contact force reduce dramatically, compared with Fig. 6.14. We also find more wrinkles in the pants, which is the consequence of the reduction of bending

Figure 6.16: Torques on hip joints during climbing stairs

rigidity.

Fig. 6.18 plots the comparison of torques of the two designs. We found the torques required by the armor are reduced when the thickness is halved, but the tendency remain similar. The peak values of the "thin" armor is about a third of the original design, which implies the additional torque on hip-thigh joint is proportional to the bending rigidity.

6.6   Aiming

Starting from the try-on configuration, the soft armor is simulated when the human body aims to ground. Fig. 6.19 plots 3 snapshots of the process. Fig. 6.19 also plots the contact forces as arrows. Again the contact forces mainly distribute on shoulders and thighs. Because the body leans to the left side, the contact forces on the right shoulder is much bigger than that on the left side. The contact forces on

(a) 0.0s

(b) 0.6s
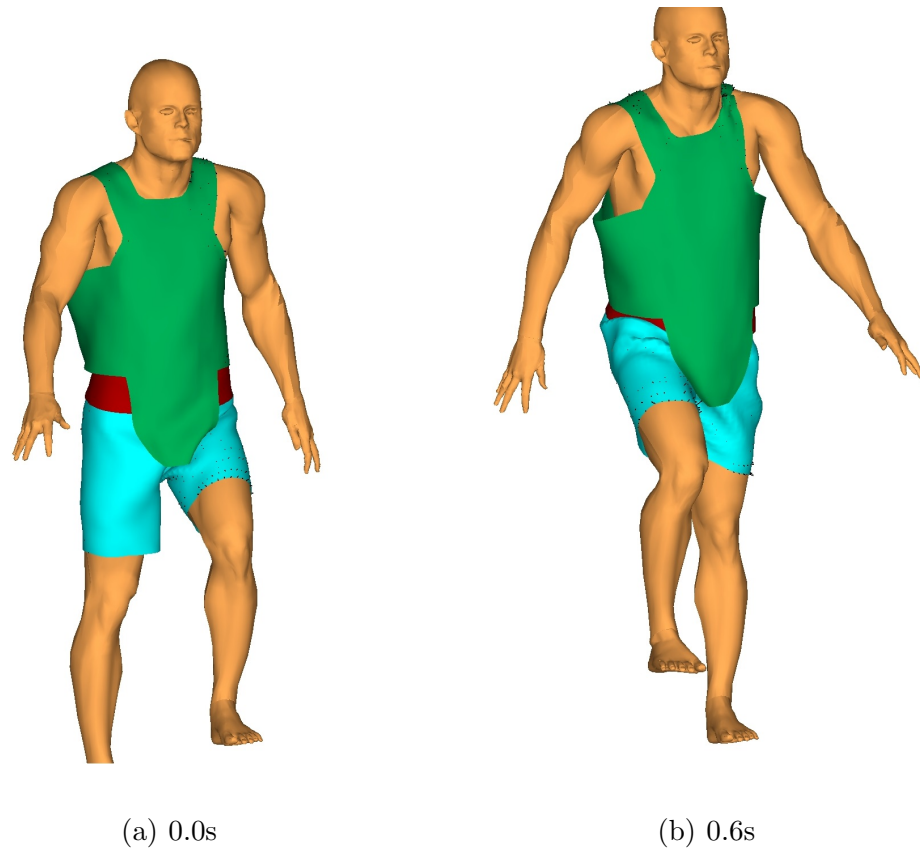
Figure 6.17: Contact forces of soft armor (1/2 design thickness) during climbing stairs
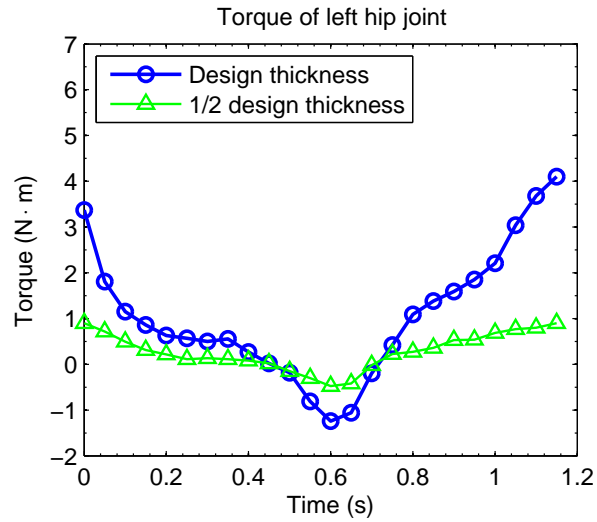
(a)
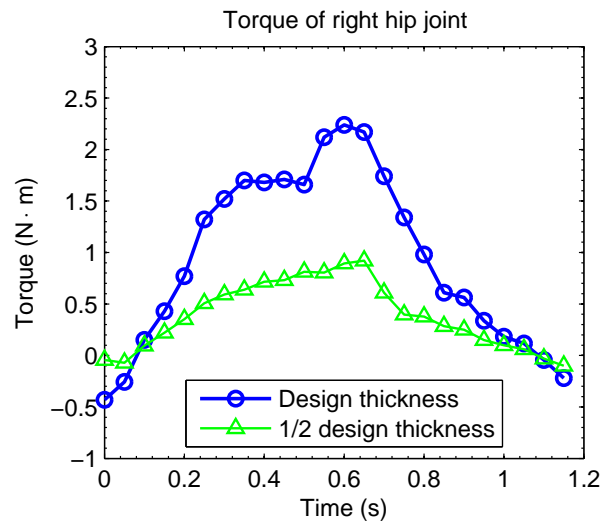
(b)

Figure 6.18: Influence of thickness to the torques on hip joints

legs during this motion is comparable with that during climbing stairs.

The first principle stress (in-plane) is ploted in Fig. 6.20. In the simulation results, the dynamic stress in the shoulder area is around $100\,N/m$, except stress concentration regions, comparable to the static stress. In snapshot (a), the maximum stress appears in the bottom of bicep, because the arm pressed the armor at this moment. In snapshot (b) and (c), the maximum stress appears in the left hip because of the stretch of the pants.

Fig. 6.21 plots the torques on the hip-thigh joints during the motion. Unlike the previous periodical torques, the torques ascend monotonously as the body gets down. It is observed that the left hip bears more torque than the other one. This is because the body leans to the left side.

We also conducted the parametric study. Again, the thickness is halved based on the original design. Fig. 6.22 shows some snapshots of "thin" armor. First of all, we found the contact forces reduce dramatically, compared with Fig. 6.19. We also find more wrinkles in the pants, which is the consequence of the reduction of bending rigidity.
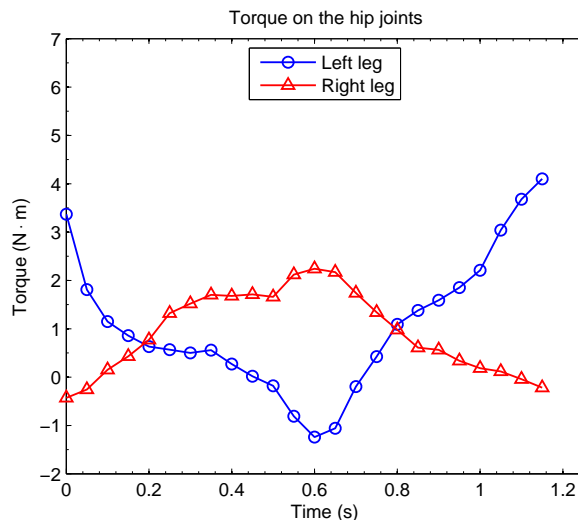
Fig. 6.23 plots the comparison of torques of the two designs. The change of the torques presents similar tendency as the previous motions and the maximum torque of the "thin" armor is about a third of the original design.

(a) 0.00s

(b) 0.25s

(c) 0.50s
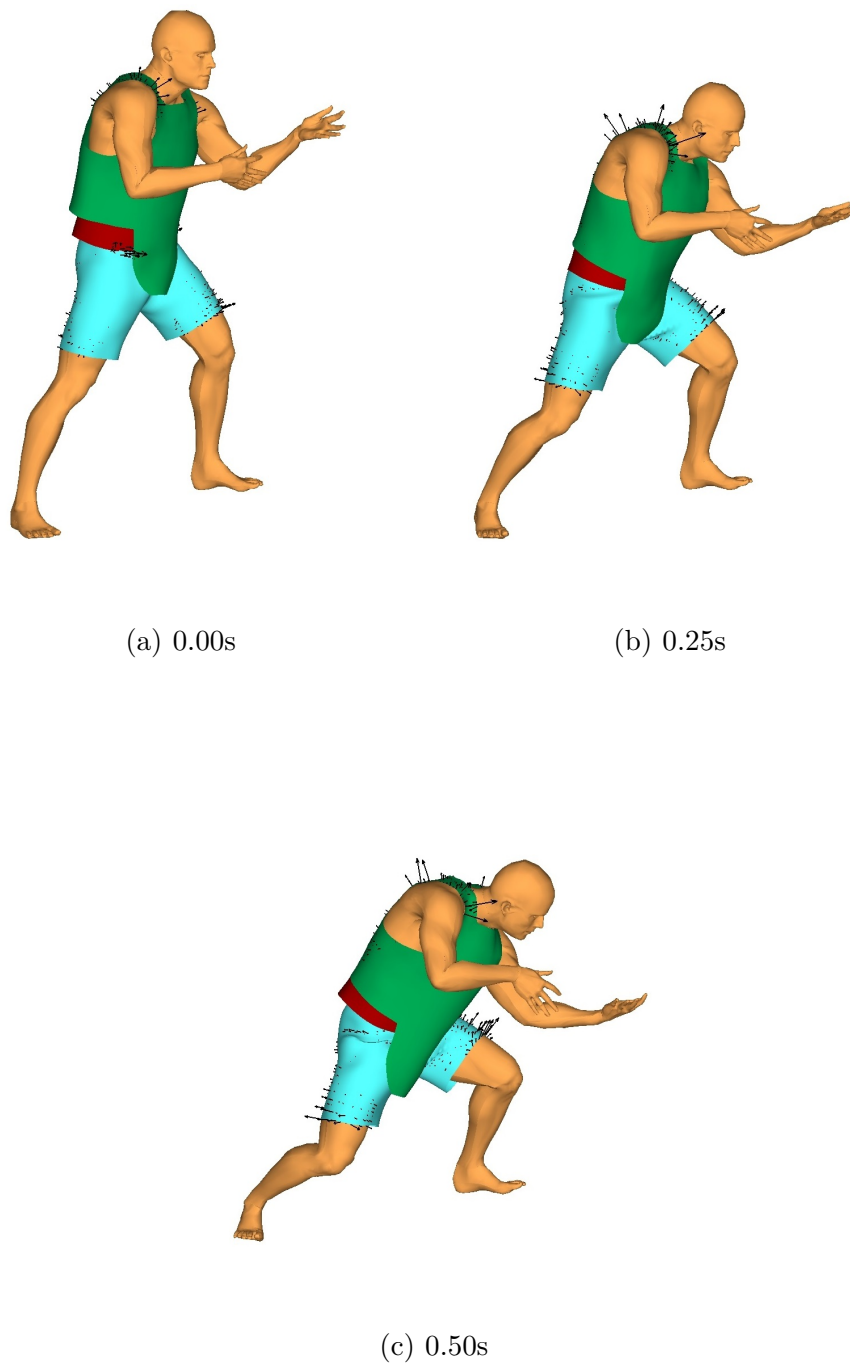
Figure 6.19: Contact force of soft armors during aiming

(a) 0.0s

(b) 0.25s

(c) 0.50s
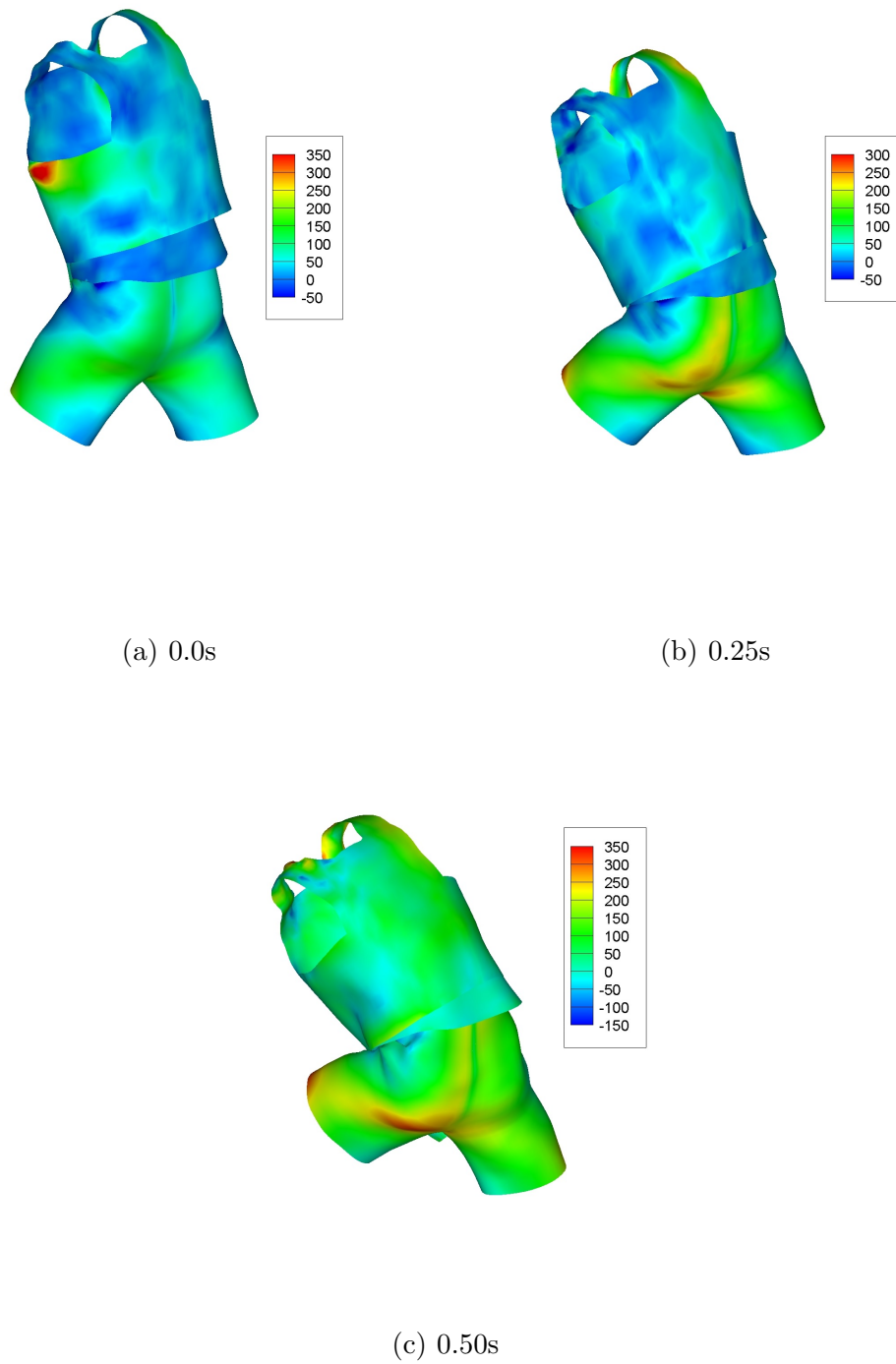
Figure 6.20: Principle in-plane stress (N/m) in aiming with armors
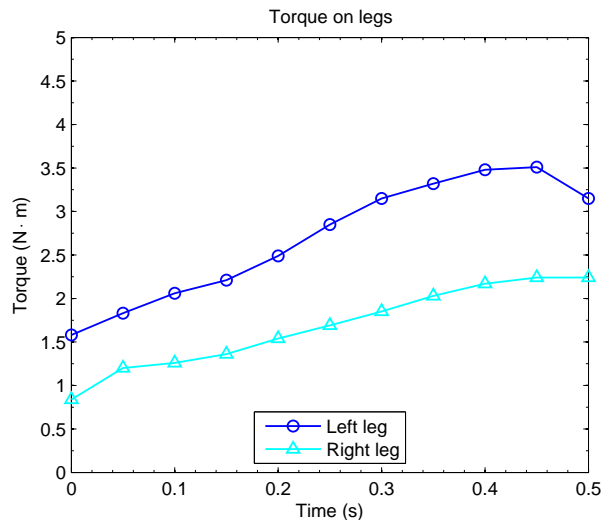
Figure 6.21: Torques on hip joints during aiming

6.7   Conclusions

The isogeometric cloth method was integrated with the predictive human model. The flowchart between CAD programs, digital human body and cloth simulation was introduced. The method was applied the analysis of a soft armor. The simulation of soft armor by far is a relative new field. In this chapter, the armor was put on the human body, and several motions were simulated. Mechanical metrics, such contact forces, torques, stresses, were reported by the simulation. The influence of armor thickness to the mechanical metrics was also studied.

The armor in this study contains three layers. The ability of simulating multi-layer armors is attributed to the new intersection resolution method and a robust self-contact treatment.

The convergence study was conducted, which demonstrated the simulation

(a) 0.0s

(b) 0.25s

(c) 0.50s
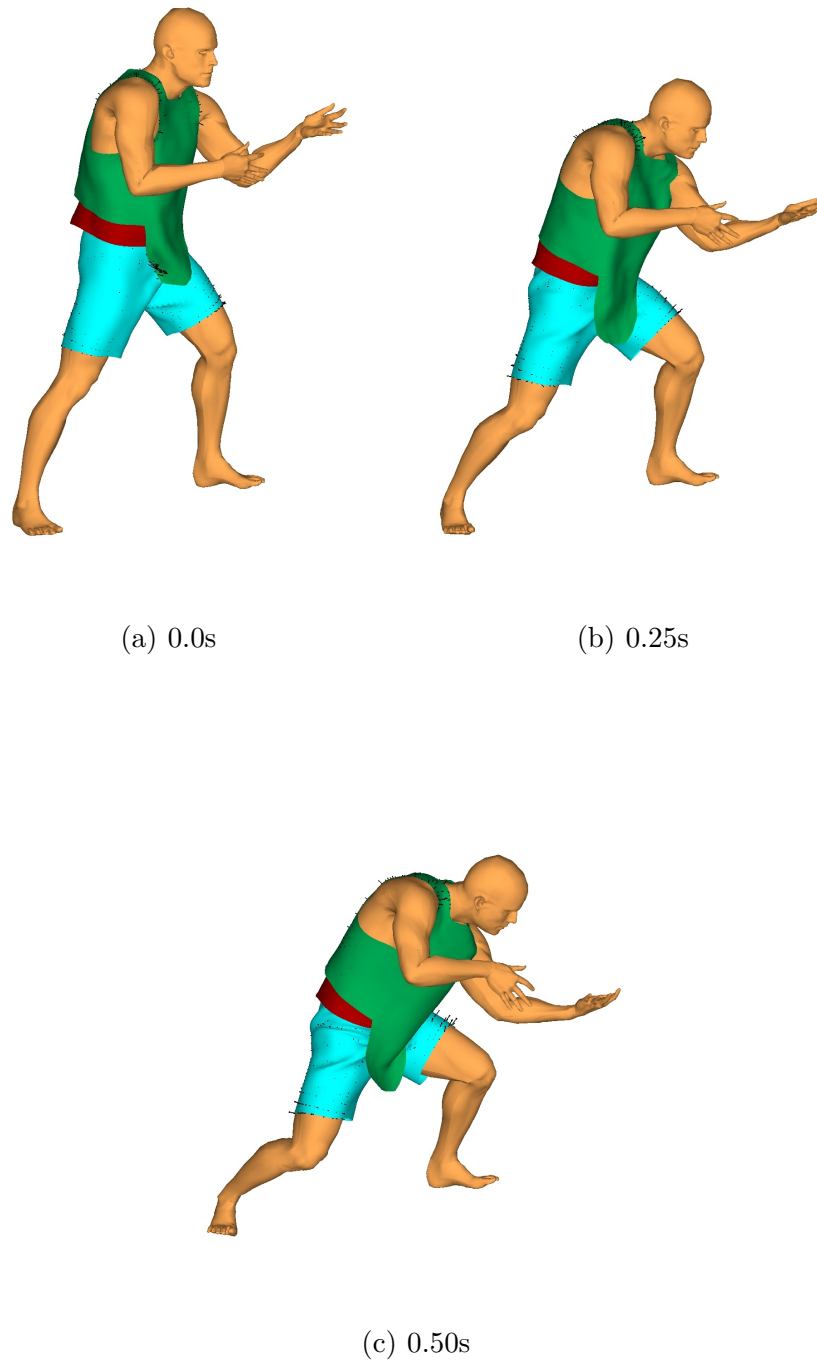
Figure 6.22: Contact force of soft armors (1/2 design thickness) during aiming
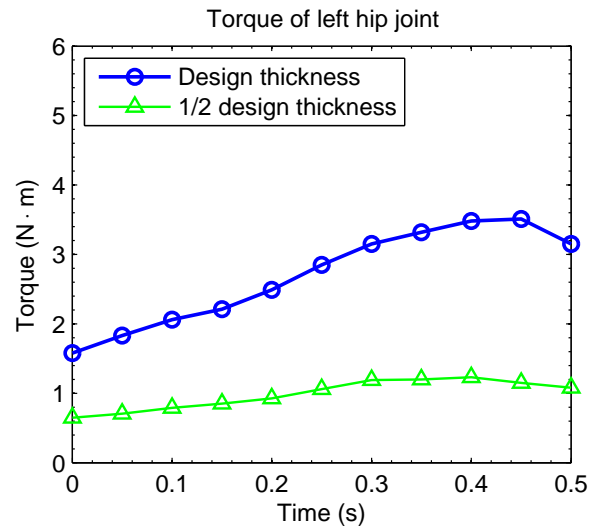
(a)
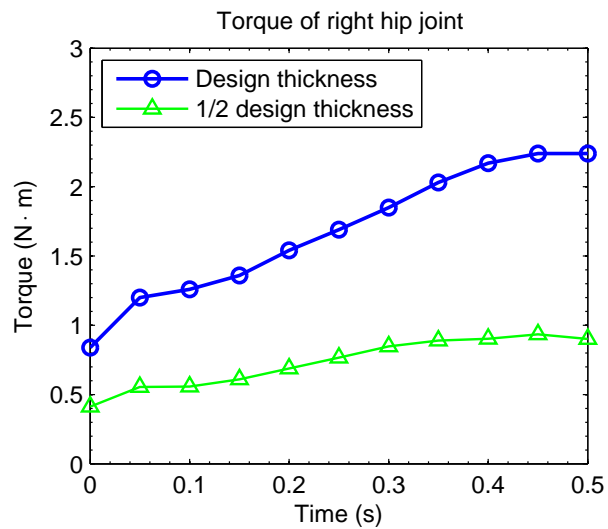
(b)

Figure 6.23: Influence of thickness to the torques on hip joints

results are convergent. This is an important difference between the current method and the mass-spring method.

Parametric studies were also conducted. By changing the thickness, different torques and contact forces are obtained. In the design process, this can help the designers to choose an optimized thickness, which can not only satisfy the protection requirement, but also satisfy the comfort requirement.

In coupling with the digital human program, the torques and the contact forces will influence the motion of human body. For example, if the human body feels a big contact force in walking, he might decide to shorten the pace length.

CHAPTER 7
CONCLUSIONS

A NURBS-based isogeometric method for cloth simulation is developed. This method includes a displacement-only NURBS Kirchhoff-Love shell model, a comprehensive contact/impact treatment algorithm, and a continuum-based fast projection method. These techniques all together significantly improve the capability of the continuum cloth simulation. A number of examples, including garment level simulations, are presented to show the performance. They demonstrated that the present method is robust, efficient, and versatile. As the first step towards a design application the method has been applied to soft armour modeling. It enables virtual try-on modeling as well as human-driven dynamic simulation.

A direct benefit of isogeometric method is that its geometry keeps consistent with the CAD model, which provides convenience in connecting the design and analysis. Another advantage of NURBS shell element is the high order of global continuity. This not only brings us smooth and visually plausible simulation results, but also enables us to simulate curved wrinkles with relative "large" elements. The spring-mass method can provide vivid pictures and animations in movie industry, but in the field mechanical design, like the design of armors, it is not accurate enough. As noted in [63], the lack of mesh convergence is a typical issue in spring-mass models. Compared with spring-mass method, NURBS model are more expensive in elastic energy computation, however, since the computation cost for elastic energy accounts only a small portion of the entire simulation cost, the isogeometric method is still competitive as

a design tool.

Contact and impact model is a bottle-neck in cloth simulation. When developing suitable algorithms for cloth, certain characteristics of cloth contact must be considered. In this work, we found it useful to break the contact responses into three subcases. In our approach, contact/impact interaction is broken out into three subtypes: persistent contact, trajectory impact, and intersection resolution. The first is essentially a smoothed treatment of lateral contact since the contact force is described by a smooth function over a fictitious contact thickness. The second captures the trajectory impact and stops the cloth particles to pass though cloth surfaces. The third can resolve cloth self-intersections, which may come from the initial configuration. The breakdown may seem unnatural and complicated at the fist glance, but it indeed enhances the robustness and efficiency of cloth simulation. The contact treatment is implemented in an operator-splitting framework wherein the three subtypes are handled sequentially. This implementation does not entail complicated computation structure. The price, however, is time step size and accuracy.

The method of intersection resolution is another contribution of this work. The traditional approach defines the resorting force in the same direction as the gradient of intersection length. This approach works for many cases, but the minimization of intersection length may not result in elimination of intersections in some cases, such as local minimization. This work develops an intersection area minimization method, which doesn't suffer any problem of local minimization. Examples demonstrate that this method is reliable and efficient.

This work also developed a continuum-based fast projection method. Since the traditional projection method was developed on the base of mass-spring model, here the method has been extended to continuum model. Examples demonstrate that the present method can largely improve the simulation efficiency.

7.1 Outlook

Although the current method presents significant improvements to the continuum approach of cloth simulation, there all still several limitations that call for future improvement.

1. All the study is focused on thin shell models. Soft armor may not be "thin", and in some occasions it may be better described by continuum element.

2. Simulations were not validated by experiments. Dynamic motions of cloth may be difficulty to validate physically because of the intrinsic non-unique nature of the motion. The deformation of cloth is a unstable buckling process, which are sensitive to material parameters, boundary/initial conditions as well as algorithmic setting. Nonetheless. there could be motions that can be used for validation. This will be left as a future study.

3. When cloth/body interaction is simulated, only the cloth side is considered as deformable models. The influence of the armor to the movement of human body is not considered.

4. This study doesn't concern any fiber-level model. The fiber-level simulation can optimize the braiding structure and thus improve the macro-scale properties.

For the armor material, Kevalar, the fiber modeling can be used to study the penetration mechanics. This is another direction of future study.

5. The material parameters and thickness in the current program are a constant inside one patch. Because of special designs and the random nature of fabrics, the materials and thickness sometimes are required to be defined as a distribution function over a patch. The definition of such function can utilize the parametric space of NURBS patch. This work will be studied in the near future.

REFERENCES

[1] G.M. Abbott, P. Grosberg, and G.A.V. Leaf. The mechanical properties of woven fabrics, Part VII: The hysteresis during bending of woven fabrics. *Textile Research Journal*, 41:345–358, 1971.

[2] J. Ascough, H.E. Bez, and A.M. Bricis. A simple finite element model for cloth drape simulation. *International Journal of Clothing Science and Technology*, 8(3):59–74, 1996.

[3] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH '98: Proceeding of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 43–54, New York, 1998. ACM.

[4] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 862–870, New York, NY, USA, 2003. ACM.

[5] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An introduction to splines for use in computer graphics & geometric modeling.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[6] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229 – 263, 2010. ¡ce:title¿Computational Geometry and Analysis¡/ce:title¿.

[7] Y. Bazilevs, L. Beirao de Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031C1090, 2006.

[8] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang. Patient-specific isogeometric fluidstructure interaction analysis of thoracic aortic blood flow due to implantation of the jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3534 – 3550, 2009.

[9] D.J. Benson, Y. Bazilevs, M.C. Hsu, and J.R. Hughes. Isogeometric shell analysis: the Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.

[10] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 63:1–63:12, New York, NY, USA, 2008. ACM.

[11] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. Estimating cloth simulation parameters from video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 37–51, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[12] D.E. Breen, D.H. House, and P.H. Getto. A physically based particle model of woven cloth. *The Visual Computer*, 8:264–277, 1992.

[13] D.E. Breen, D.H. House, and M.J. Wozny. Predicting the drape of woven cloth using interacitng particles. In *SIGGRAPH '94: Proceeding of the 21th Annual Conference on Computer Graphics and Interactive techniques*, pages 365–372, New York, 1994. ACM.

[14] R. Bridson. *Fluid Simulation for Computer Graphics*. Ak Peters Series. A K Peters, 2008.

[15] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21:594–603, 2002.

[16] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceeding of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 28–36, San Diego, 2003. ACM and Eurographics.

[17] B.J. Chen and M. Govindaraj. A physically-based model of fabric drape using flexible shell theory. *Textile Research Journal*, 65(6):324–330, 1995.

[18] Seonho Cho and Seung-Hyun Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38:53–70, 2009. 10.1007/s00158-008-0266-z.

[19] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 604–611, New York, NY, USA, 2002. ACM.

[20] F. Cirak and M. Ortiz. Fully C^1-conforming subdivision element for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51:813–833, 2001.

[21] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 30:175–186, 2000.

[22] J. R. Collier, Billie J. Collier, Gina O'Toole, and S. M. Sargand. Drape prediction by means of finite-element analysis. *Journal of the Textile Institute*, 82(1):96–107, 1991.

[23] Frederic Cordier, Hyewon Seo, and Nadia Magnenat-Thalmann. Made-to-measure technologies for an online clothing store. *Computer Graphics and Applications, IEEE*, 23(1):38 – 48, jan/feb 2003.

[24] J.A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257C5296, 2006.

[25] J.A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160C4183, 2007.

[26] C. Deboor. *A Practical Guide to Splines.* Springer-Verlag Berlin and Heidelberg GmbH & Co. K, December 1978.

[27] İ. Temizer, P. Wiggers, and T.J.R. Hughes. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200:1100–1112, 2011.

[28] J.W. Eischen, S.G. Deng, and T.G. Clapp. Finite-element modeling and control of flexible fabric parts. *IEEE Computer Graphics and Applications*, 16(5):71–80, SEP 1996.

[29] T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes. $\bar{B}$ and $\bar{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher order nurbs elements. *Computer Methods in Applied Mechanics and Engineering*, 197:2732C2762, 2008.

[30] G.E. Farin. *Curves and surfaces for CAGD: a practical guide.* Acadenic Press, San Diego, 4th edition, 2002.

[31] Gerald E. Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code.* Academic Press, Inc., Orlando, FL, USA, 4th edition, 1996.

[32] L. Gan, N.G. Ly, and G.P. Steven. A study of fabric deformation using nonlinear finite-elements. *Textile Research Journal*, 65(11):660–668, NOV 1995.

[33] Sarah F. F. Gibson and Brian Mirtich. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratories, 1997.

[34] Adrian R. Goldenthal. *Implicit Treatment of Constraints for Cloth Simulation.* PhD thesis, Hebrew University, 2010.

[35] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):to appear, 2007.

[36] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 62–67, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[37] David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. Robust treatment of simultaneous collisions. In *ACM Transactions on Graphics (TOG)*, volume 27, page 23. ACM, 2008.

[38] Min Hong, Min hyung Choi, Sunhwa Jung, and Samuel Welch. Effective constrained dynamic simulation using implicit constraint enforcement. In *In International Conference on Robotics and Automation*, pages 4520–4525, 2005.

[39] Donald House, Richard W. Devaul, and David E. Breen. Towards simulating cloth dynamics using interacting particles. *International Journal of Clothing Science and Technology*, 8:75–94, 1996.

[40] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[41] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5 - 8):301 – 313, 2010.

[42] CLO Virtual Fashion Inc. http://www.marvelousdesigner.com.

[43] Kwang jin Choi and Hyeong seok Ko. Research problems in clothing simulation. *CAD*, 37:585–592, 2005.

[44] Jonathan M. Kaldor, Doug L. James, and Steve Marschner. Simulating knitted cloth at the yarn level. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 65:1–65:9, New York, NY, USA, 2008. ACM.

[45] S. Kawabata. *The Standardization and Analysis of Hand Evaluation*. The Textile Machinery Society of Japan, Osaka, 1980.

[46] M. Keckeisen, S.L. Stoev, M. Feurer, and W. Strasser. Interactive cloth simulation in virtual environments. In *Virtual Reality, 2003. Proceedings. IEEE*, pages 71 – 78, march 2003.

[47] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with kirchhoff-love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[48] Duksu Kim, Jae-Pil Heo, Jaehyuk Huh, John Kim, and Sung-Eui Yoon. Hpccd: Hybrid parallel continuous collision detection using cpus and gpus. *Comput. Graph. Forum*, 28(7):1791–1800, 2009.

[49] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37-40):2982 – 2995, 2009.

[50] J. Kim. *Fabric mechanics analysis using large deformation orthotropic shell theory*. PhD thesis, North Carolina State University, Rayleigh, NC, 1991.

[51] T. J. Lahey and G. R. Heppler. Mechanical modeling of fabrics in bending. *Transactions of the ASME*, 71:32–40, 2004.

[52] John Lasseter. Principles of traditional animation applied to 3d computer animation. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 35–44, New York, NY, USA, 1987. ACM.

[53] Yongjoon Lee, Sung-eui Yoon, Seungwoo Oh, Duksu Kim, and Sunghee Choi. Multi-resolution cloth simulation. *Computer Graphics Forum*, 29(7):2225–2232, 2010.

[54] OptiTex International Pty Ltd. `http://www.optitex.com`.

[55] J. Lu. Isogeometric contact analysis: Geometric basis and formulation for frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 200:726–741, 2011.

[56] Jia Lu. Circular element: Isogeometric elements of smooth boundary. *Computer Methods in Applied Mechanics and Engineering*, 198(30-32):2391 – 2402, 2009.

[57] Jia Lu and Xianlian Zhou. Cylindrical element: Isogeometric model of continuum rod. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4):233 – 241, 2011.

[58] X. Man and C.C. Swan. A mathematical modeling framework for analysis of functional clothing. *Journal of Engineered Fibers and Fabrics*, 2(3):10–28, 2007.

[59] Xiaolin Man. *A mathematical & computational multiscale clothing modeling framework*. PhD thesis, University of Iowa, 2006.

[60] Xiaolin Man, Colby C. Swan, and Salam Rahmatalla. A clothing modeling framework for uniform and armor system design. In *SPIE Defense and Security Symposium*, pages 17–22, 2006.

[61] Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.*, 30(4):37:1–37:12, July 2011.

[62] M. Moore and J.Wilhelms. Collision detection and response for computer animation. *SIGGRAPH Computer Graphics*, 22:289–298, 1988.

[63] Andrew Nealen, Matthias Mller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[64] Seungwoo Oh, Junghyun Ahn, and Kwangyun Wohn. Low damped cloth simulation. *The Visual Computer*, 22(2):70–79, 2006.

[65] SeungWoo Oh, Junyong Noh, and Kwangyun Wohn. A physically faithful multigrid method for fast cloth simulation. *Computer Animation and Virtual Worlds*, 19(3-4):479–492, 2008.

[66] X.Q. Peng and J. Cao. A continuum mechanics-based non-orthogonal constitutive model for woven composite fabrics. *Composites: Part A*, 36:859–874, 2005.

[67] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, Berlin, 1995.

[68] X. Provot. Deformation constrains in a mass-spring model to describe rigid cloth behavior. In *Graphic Interface '95*, pages 147–154, New York, 1995. Canadian human-computer communication society.

[69] H. Qin. D-NURBS: a physics-based framework for geometric design. *IEEE Transactions on Visualization and Computer Graphics*, 2:85–96, 1994.

[70] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 64:1–64:11, New York, NY, USA, 2008. ACM.

[71] Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, pages 339–350, 2009.

[72] Yu-Deok Seo, Hyun-Jung Kim, and Sung-Kie Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures*, 47(11-12):1618 – 1640, 2010.

[73] J.C. Simo and D.D. Fox. On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parameterizatoin. *Computer Methods in Applied Mechanics and Engineering*, 72:267–304, 1989.

[74] J.C. Simo and D.D. Fox. On a stress resultant geometrically exact shell model. Part II: The linear theory; computatinal aspects. *Computer Methods in Applied Mechanics and Engineering*, 73:53–92, 1989.

[75] J.C. Simo and D.D. Fox. On a stress resultant geometrically exact shell model. Part III: Aspects of the nonlinear theory. *Computer Methods in Applied Mechanics and Engineering*, 79:21–70, 1990.

[76] Karl Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 15–22, New York, NY, USA, 1994. ACM.

[77] Jonathan Su, Craig Schroeder, and Ronald Fedkiw. Energy stability and fracture for frame rate rigid body simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 155–164, New York, NY, USA, 2009. ACM.

[78] J. G. Teng, S. F. Chen, and J. L. Hu. A finite-volume method for deformation analysis of woven fabrics. *International Journal for Numerical Methods in Engineering*, 46(12):2061–2098, 1999.

[79] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13:103–136, 1994.

[80] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 205–214, New York, NY, USA, 1987. ACM.

[81] J. A. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. *ACM SIGGRAPH Computer Graphics*, 24:129–137, 1990.

[82] B. Thomaszewski, M. Wacker, and W. Strasser. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 107–116, 2006.

[83] Bernhard Thomaszewski and Wolfgang Blochinger. Physically based simulation of cloth on distributed memory architectures. *Parallel Computing*, 33(6):377 – 390, 2007.

[84] Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straβer. Continuum-based strain limiting. *Computer Graphics Forum*, 28(2):569–576, 2009.

[85] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.

[86] P. Volino and N.M. Thalmann. Accurate collision response on polygonal meshes. In *Computer Animation 2000. Proceedings*, pages 154 –163, 2000.

[87] Pascal Volino, Frederic Cordier, and Nadia Magnenat-Thalmann. From early virtual garment simulation to interactive fashion design. *Computer-Aided Design*, 37(6):593 – 608, 2005.

[88] Pascal Volino and Nadia Magnenat-Thalmann. *Virtual Clothing*. Springer, 2000.

[89] Pascal Volino and Nadia Magnenat-Thalmann. Resolving surface collisions through intersection contour minimization. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1154–1159, New York, NY, USA, 2006. ACM.

[90] Wolfgang A. Wall, Moritz A. Frenzel, and Christian Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2976 – 2988, 2008.

[91] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James O'Brien. Example-based wrinkle synthesis for clothing animation. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 107:1–107:8, New York, NY, USA, 2010. ACM.

[92] Martin Wicke, Hermes Lanker, and Markus Gross. Untangling cloth with boundaries. In *Proceedings of Vision, Modeling, and Visualization (VMV)*, pages 349–356. Citeseer, 2006.

[93] Y. Wesley Wong and Sergio Pellegrino. Wrinkled membranes part II: Analytical models. *Journal of mechanics of materials and structures*, 1(1):25–59, 2006.

[94] Jingzhou Yang, Joo H. Kim, Karim Abdel-Malek, Timothy Marler, Steven Beck, and Gregory R. Kopp. A new digital human environment and assessment of vehicle interior design. *Computer-Aided Design*, 39(7):548 – 558, 2007.

[95] X. Zhou and J. Lu. NURBS-based Galerkin method and application to skeletal muscle modeling. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling, SPM'05*, pages 71–78, New York, 2005. ACM.