

**Isometrically Deforming Cloth
with Simultaneous Collision Handling**

by

Zheng Wang

B. Engineering, Xi'an Jiaotong University, 2000

B. Science, Simon Fraser University, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

February 2012

© Zheng Wang, 2012

Abstract

We propose an isometrically deforming cloth model for physics-based simulation, which works with a conforming triangle mesh yet doesn't suffer from locking. This in itself is an important step forward for cloth animation, as many common materials essentially do not stretch, yet the only prior method capable of handling this regime used a non-conforming mesh which considerably complicated collision handling. We further introduce a general integration scheme for constrained dynamics with additional potential energy terms and frictional contact, all fully and implicitly coupled together rather than staggered in time. Shells with stiff bending forces greatly benefit from the simultaneous coupling. Moreover, we show that solving for one time step in this scheme can be transformed to solving a Newton sequence of unconstrained linear least-squares problems, resulting only in sparse, symmetric positive definite matrices which can be handled particularly efficiently.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
Acknowledgments	viii
1 Introduction	1
2 Related Work	2
3 The Method	4
3.1 Isometrically Deforming Cloth Model	4
3.1.1 Inextensibility and Conforming Constraints	4
3.1.2 Boundary Constraints	6
3.2 Solving the Constrained Dynamics	7
3.2.1 Bending Energy	7
3.2.2 Conservative Dynamics as a Minimization Problem	9
3.2.3 Lagrange Multipliers	10
3.2.4 Combining Constraints with Potential Energy	11
3.2.5 Transforming to a Sparse SPD System with Gauss-Newton	12
3.3 Collisions and Friction	13
3.3.1 Modelling Collisions as Constraints	14

3.3.2	Friction	16
3.3.3	Friction Regularization	17
4	Results	19
5	Conclusions	27
	Bibliography	29

List of Tables

Table 3.1	Pseudocode of constraints projection with collision detection	15
Table 4.1	Simulation performances in different scenarios	21

List of Figures

Figure 3.1 Continuous discretization of isometric mesh. Edge midpoints were added to the initial triangle mesh as red nodes. The shaded triangles remain rigid through simulation via isometry constraints. The positions of black nodes are determined by the conforming constraints using the surrounding shaded triangles. 5

Figure 3.2 Three types of boundary nodes on cloth mesh. Vertex A is a corner node with one incident triangle, so its position is enforced to the extrapolated value of triangle nodes $\triangle a_0 a_1 a_2$. Vertex B has more than two incident triangles, and its boundary constraint enforces that the extrapolated values of two boundary triangles $\triangle a_0 a_1 a_2$ and $\triangle b_0 b_1 b_2$ must match. Vertex C has exactly two incident triangles, so it's similar to B and requires triangles $\triangle b_0 b_1 b_2$ and $\triangle c_0 b_2 c_1$ to match on the extrapolated values. 6

Figure 3.3 Edge-based Crouzeix-Raviart basis function drawn in blue (Left) and the discrete Laplacian entry for adjacent nodes i, j is evaluated as $L_{ij} = -2 \cot \angle e_i, e_j$ 8

Figure 3.4	2D illustrations of staggered collision processing and our method simulating stiff material. The staggered version (a) pushed the strand nodes up to non-intersection state, but violated inextensibility constraints (thickened red edges) and the stiffness property. Our method (b) coupled collisions with all the constraints and bending energy and produced the expected result.	14
Figure 4.1	Comparison of a stiff surface bending simulation using staggered collision detection (left) and simultaneous collision detection (right)	20
Figure 4.2	Inextensibility comparison between staggered collision handling and our method. Running 200 frames of the stiff surface bending simulation on a sphere, we measured the maximum strain of the original mesh edges.	20
Figure 4.3	Bending energy comparison between staggered collision handling and our method. With the same stiff surface bending simulation, we plot the bending energies accumulated by the end of each frame from two methods.	21
Figure 4.4	Comparison of simulation time using staggered collision handling (red) and our method (blue)	22
Figure 4.5	Cloth draping on a rotating ball	23
Figure 4.6	Armadillo model collapsing with low bending stiffness $k_b = 5.0 \times 10^3$	24
Figure 4.7	Armadillo model collapsing with high bending stiffness $k_b = 5.0 \times 10^4$	25
Figure 4.8	Walking model dressed with a shirt	26

Acknowledgments

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I offer my deep and sincere gratitude to my supervisor, Dr. Robert Bridson, who has supported me throughout my thesis with his great ideas, knowledge, patience and personal guidance whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

I would like also to thank Tyson Brochu for his friendly help and thesis reader Dinesh K. Pai for his valuable advice.

Last but not the least, I own my special thanks to my wife Anni for her great support, encouragement, understanding and patience at all times.

Chapter 1

Introduction

Cloth simulation is now a standard tool in animation, yet the most common material limit of isometric deformation (no in-plane stretching, only bending) is still poorly handled, and collision handling with stiffer materials is similarly problematic with current methods. We resolve these issues and introduce a much more general framework for efficient fully implicit time integration of constrained dynamics in the present paper.

We introduce an isometrically deforming cloth model with a standard continuous triangle mesh via careful subdivision of a base mesh, avoiding the need for English and Bridson’s non-locking conforming/non-conforming mesh coupling [8]. We design a fully implicit, fully coupled dynamics framework including nonlinear constraints such as isometry, frictional contact, and general potential energies such as bending with a possibly curved rest state. We further describe a technique to transform the usual indefinite KKT linear systems derived from the constrained optimization problem into equivalent sparse and symmetric positive definite systems, which can be solved much more efficiently.

Chapter 2

Related Work

Many previous works have tackled simulation of cloth as an inextensible material. One of the most common approaches models cloth with mass-spring systems with additional limits on deformation, iteratively fixing edges stretched or compressed beyond some threshold [5, 18]. The latter work also combined this with a multi-stage collision processing algorithm which we adopt at a lower level in our algorithm. Other systems have instead followed Baraff and Witkin use of stiff forces resist in-plane deformation of a mesh, handled efficiently with implicit integration [2]. Zero stretch can even be imposed as a hard constraint solved efficiently with Lagrange multipliers [13]. However, as English and Bridson [8] argued, all of these approaches fall victim to bad locking artifacts in the isometric limit, as isometry fundamentally overconstrains a triangle mesh from bending — Liu et al. [15] show an n -triangle mesh is only left with $O(\sqrt{n})$ degrees of freedom. Goldenthal et al. [10] sidestepped the issue by only restricting deformation along certain lines, allowing shearing in a quad mesh; English and Bridson instead solved locking with a non-conforming discretization on edge midpoints, complicating collision handling.

Constrained dynamics have a long history in computer graphics. We highlight Baraff’s treatment of rigid bodies [1] and House et al.’s use in cloth [13]. It is common to enforce constraints at the velocity level; as this suffers from numerical drift, post-stabilization steps are required to correct positions. House et al. reported difficulties in tuning post-stabilization terms; Cline and Pai [7] provide an excellent survey on post-stabilization techniques. Goldenthal et al. [10] instead proposed a

position-level constraint algorithm called fast projection, which projects the positions onto the constraint manifold such that the constraints will be satisfied without drift, and was later extended to second order accuracy with BDF2 by English and Bridson [8]. We further extend this method with general potential energies.

Contact and collision detection for deformable objects is a huge area. While we use Bridson et al.'s collision methods at a low level [5], we also couple contact into the full dynamics of the system. Baraff and Witkin [2] incorporated contact constraints into their time integration via modifications to the mass matrix. Irving et al. [14] further integrated object contact and self-collision as linear constraints into their incompressible Poisson equations and setting the correct normal velocity for colliding elements. Otaduy et al. [17] took an even more sophisticated approach with inequality constraints, formulating the contact deformations in linear complementarity problems (LCP) and solving the constrained deformation problem by iterative constraint anticipation (ICA).

Chapter 3

The Method

3.1 Isometrically Deforming Cloth Model

English and Bridson’s non-conforming cloth model leads to a discrete surface which is no longer continuous at vertices, therefore it cannot be directly used in collisions or rendering. To get a conforming mesh for collision and rendering, they computed a ghost mesh at each time step by averaging the non-conforming vertices of the simulation mesh. Drawbacks of this approach include the overhead of converting back and forth between simulation and ghost meshes (involving a large linear system to solve), and the difficulty of coupling contact with internal dynamics which, we shall see, is critical for stiff shells.

We were inspired by the non-conforming model, but seek an embedding of it in a standard conforming triangle mesh. Our solution is to use one step of 4 : 1 subdivision. We label the new vertices (edge midpoints of the base mesh) “red” nodes and the rest (the original vertices of the base mesh) “black” nodes. The internal dynamics are captured by the red nodes, avoiding locking, while the black nodes keep the mesh conforming with simple position constraints.

3.1.1 Inextensibility and Conforming Constraints

In Figure 3.1, the circles indicate red and black nodes. To make the surface mesh inextensible, we force the parameter space distances between red nodes in each triangle to remain unchanged. For a pair of nodes (x_i, x_j) , the inextensibility con-

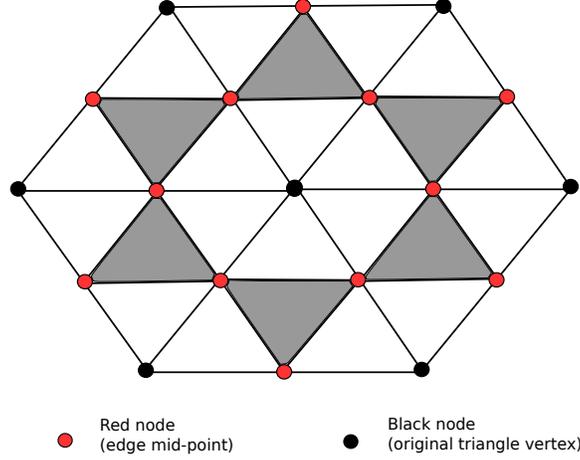


Figure 3.1: Continuous discretization of isometric mesh. Edge midpoints were added to the initial triangle mesh as red nodes. The shaded triangles remain rigid through simulation via isometry constraints. The positions of black nodes are determined by the conforming constraints using the surrounding shaded triangles.

straint is defined as

$$C_{ij}(x) = \|x_i - x_j\| - d_{ij} = 0, \quad (3.1)$$

and the derivative of the constraint function w.r.t. x_i is

$$\nabla_i C_{ij}(x) = \frac{x_i}{\|x_i - x_j\|}. \quad (3.2)$$

The position of a black node will remain the average of the extrapolated position of the red nodes from all incident triangles. In triangle i with edges a , b and c , the position of vertex k located opposite to edge c is extrapolated as $x_i^{(a)} + x_i^{(b)} - x_i^{(c)}$. The averaged position of black node x_k with n incident triangles is computed as $\frac{1}{n} \sum_{i=1}^n (x_i^{(a)} + x_i^{(b)} - x_i^{(c)})$, and so the accompanying constraint is

$$C_k(x) = x_k - \frac{1}{n} \sum_{i=1}^n (x_i^{(a)} + x_i^{(b)} - x_i^{(c)}) = 0. \quad (3.3)$$

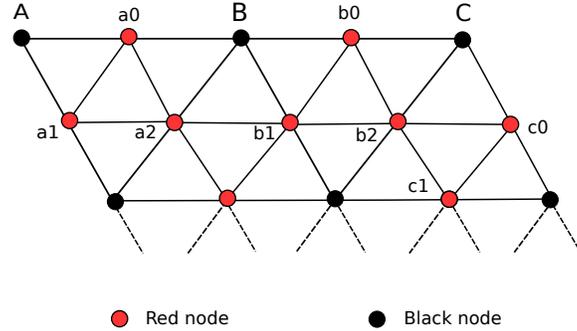


Figure 3.2: Three types of boundary nodes on cloth mesh. Vertex A is a corner node with one incident triangle, so its position is enforced to the extrapolated value of triangle nodes $\triangle_{a_0 a_1 a_2}$. Vertex B has more than two incident triangles, and its boundary constraint enforces that the extrapolated values of two boundary triangles $\triangle_{a_0 a_1 a_2}$ and $\triangle_{b_0 b_1 b_2}$ must match. Vertex C has exactly two incident triangles, so it's similar to B and requires triangles $\triangle_{b_0 b_1 b_2}$ and $\triangle_{c_0 b_2 c_1}$ to match on the extrapolated values.

Note that while (3.3) constrains the positions of black nodes in terms of red nodes, all nodes are fully dynamic particles, each with its own mass.

Here is a sketch of why this subdivision scheme plus constraints gives enough DOFs for non-locking simulation. The subdivision adds a new vertex per edge, so it increases the total number of vertices to $e + v \approx 4v$, which gives $12v$ DOFs. The inextensibility requirement enforces 3 constraints per triangle, so it contributes $3t \approx 6v$. The conforming constraints are applied on the original vertices, so they remove $3v$ DOFs. Summing this up gives $12v - 6v - 3v = 3v$ DOFs, which is adequate for good simulation.

3.1.2 Boundary Constraints

Although our mesh is conforming and continuous along boundaries, similar artifacts to the non-conforming discretization still arise due to insufficient bindings among the red nodes located at boundary and corner triangles. The conforming constraints specified above only guarantee the continuity of black node with respect to its surrounding red nodes, but they do not prevent some corner triangles from rotating arbitrarily [8]. Therefore, additional boundary constraints are re-

quired to make the simulation well-posed. Our solution is to constrain red nodes belonging to triangle pairs on the boundary so that the extrapolated positions of these nodes matches: see Figure 3.2. For a pair of triangles t_i and t_j with edges c opposite the shared vertex, the requisite boundary constraint is:

$$C_{ij}(x) = \left(x_i^{(a)} + x_i^{(b)} - x_i^{(c)}\right) - \left(x_j^{(a)} + x_j^{(b)} - x_j^{(c)}\right) = 0. \quad (3.4)$$

Comparing to the prior non-conforming model, we were able to remove the constraints on the interior triangles which share edges with boundary triangles. In our experiments, just enforcing pairwise adjacent boundary triangles to conform to the shared vertices was sufficient, such that no artifacts occurred due to erroneous degrees of freedom on boundary and corners. As a result of less constraints, more degrees of freedom on these interior triangles allow the simulation to produce finer wrinkles close to boundaries.

3.2 Solving the Constrained Dynamics

3.2.1 Bending Energy

Besides the inextensibility requirements, another important fact of fabrics is that they resist bending deformations. However, the bending stiffness varies vastly depending on the material. Silk exhibits very weak bending stiffness for example, perhaps just enough to attain smoothness in typical situations, yet thick rubber is much stiffer so we don't normally see any wrinkles.

Grinspun et al. [11] and Bridson et al. [6] worked out reasonably convergent low-order bending models based on the dihedral angle of each pair of edge-adjacent triangles, with the possibility of non-zero rest angles. Our work is built first on Bergou et al.'s later elaboration of these ideas to a quadratic-in-position bending energy for (near-)isometrically deforming surfaces with flat rest state [4].

The bending potential energy of an isometrically deformed cloth is written as:

$$B(x) = \frac{1}{2} k_b x^T (L^T M^{-1} L) x. \quad (3.5)$$

Here k_b is the bending stiffness coefficient, x is a vector of all vertex positions,



Figure 3.3: Edge-based Crouzeix-Raviart basis function drawn in blue (Left) and the discrete Laplacian entry for adjacent nodes i, j is evaluated as $L_{ij} = -2 \cot \angle e_i, e_j$

L a discrete surface Laplacian derived with the Finite Element Method (FEM), and M a corresponding form of mass matrix. In particular, view this as a positive weighted sum of squared Laplacians over the surface mesh. Taking the the negative gradient of $B(x)$, the explicit formula for the bending force is:

$$F_b = -\nabla B(x) = -k_b M^{-1} Lx \quad (3.6)$$

Stiffness and mass matrices L and M are defined based on the choice of basis function ϕ [22]:

$$L_{mn} = \int_S \nabla \phi_m \cdot \nabla \phi_n dA \quad \text{and} \quad M_{mn} = \int_S \phi_m \cdot \phi_n dA \quad (3.7)$$

The FEM basis functions space can be chosen from either vertex-based linear Lagrange or edge-based linear Crouzeix-Raviart basis functions. Observing that our discretization model uses the edge mid-points as the red nodes, it's more suitable to apply edge-based FEM basis functions. For each edge i (or red node) in our triangle mesh, ϕ_i is defined as a linear function on triangles: $\phi_i(i) = 1$ and $\phi_i(j) = 0$ if $j \neq i$, where $\phi_i(j)$ denotes the value of ϕ_i at the red node j . The corresponding discrete Laplacian matrix is derived as: $L_{ij} = -2 \cot \angle e_i, e_j$ if $j \neq i$ and e_i and e_j share an angle.

This bending model works well on cloth with a flat rest state. However, when artists prefer pre-shaped folds on cloth, for example, cloth with non-zero rest angles is required. Garg et al. [9] further developed this hinge-based bending model for thin shells and demonstrated that bending energy is a cubic polynomial in positions

under isometric deformations — though still is naturally expressible as a sum of squared (but nonlinear) terms over the mesh. However, to accelerate our implicit solver, we also developed an approximation for handling nonzero rest angles with a purely quadratic energy, which means the Hessian of the energy need not be recomputed each Newton step.

Suppose the initial curvature is $\delta = Lx_0$ where x_0 is the vector of rest-state positions. For each red node of our surface, δ is the mean curvature normal of this node. The length of this normal vector is a scalar which captures the initial curvature of the node (which is translation and rotation invariant), but unfortunately the direction of δ is only translation invariant, *not* rotation invariant. Since bending energy must be translation and rotation invariant for physical plausibility, we need to correct for the direction of δ . We introduce a transformation matrix $T(x)$ to match the current “average” rotation and restore rotation invariance, giving a modified bending energy formula:

$$B(x) = \frac{1}{2}k_b (Lx - T(x)\delta)^T M^{-1} (Lx - T(x)\delta) \quad (3.8)$$

During time integration, we take the further approximation of freezing the transformation $T(x)$ as $T(\tilde{x})$, where \tilde{x} is the predicted position before constraint projection. Since both \tilde{x} and δ are known, $T(\tilde{x})$ is computed as a constant for the time step, leading to a purely quadratic energy. This last approximation can violate rotation invariance if the final positions end up significantly rotated from the predicted positions, but in practice appears to be acceptable while providing a useful code optimization.

3.2.2 Conservative Dynamics as a Minimization Problem

Although the bending force can be expressed as a linear function of positions in (3.6), for stiff materials explicit integration becomes inefficient due to stability time step restrictions. We turn to stiff-capable implicit solvers such as Backwards Euler or BDF methods, but restated as a minimization problem. Letting $\Phi(x)$ express the sum of potential energies of the system, one step of Backwards Euler is:

$$x^{(n+1)} = x^{(n)} + \Delta t v_{(n+1)} \quad (3.9)$$

$$v^{(n+1)} = v^{(n)} - \Delta t M^{-1} \frac{\partial \Phi(x^{(n+1)})}{\partial x} \quad (3.10)$$

Introducing \tilde{x} as the inertial predicted position $\tilde{x} = x^{(n)} + \Delta t v^{(n)}$ and eliminating $v_{(n+1)}$ simplifies this to:

$$M(x^{(n+1)} - \tilde{x}) + \Delta t^2 \frac{\partial \Phi(x^{(n+1)})}{\partial x} = 0 \quad (3.11)$$

This is easily recognized as the first order optimality (KKT) condition for the following minimization problem:

$$x^{(n+1)} = \underset{x}{\operatorname{argmin}} \frac{1}{2} (x - \tilde{x})^T M (x - \tilde{x}) + \Delta t^2 \Phi(x) \quad (3.12)$$

3.2.3 Lagrange Multipliers

For a constrained dynamical system, suppose it has to satisfy a positional constraint written as $C(x) \geq 0$. Typically the inequality comes from non-penetrating contact constraints, which require non-negative distance between elements. In other words, the constraint forces push them away rather than pulling them together. However, for simplicity purposes we restrict attention to the simpler equality case $C(x) = 0$ for now, returning to contact constraints in section 3.3.1.

Using Lagrange multipliers, we can write the constraint force as $F_c = \frac{\partial C(x)^T}{\partial x} \lambda = J^T \lambda$. Here J is the Jacobian of the constraint function and λ is the Lagrange multiplier vector, which has the same dimension as $C(x)$. Backwards Euler subject to constraints can then also be expressed as a constrained minimization:

$$x^{(n+1)} = \underset{x: C(x)=0}{\operatorname{argmin}} \frac{1}{2} (x - \tilde{x})^T M (x - \tilde{x}) \quad (3.13)$$

This is essentially the goal of Goldenthal et al.'s fast projection algorithm [10]: seeking the point on the constraint manifold which is the closest to the predicted position. We take a standard Newton approach, linearizing the constraint function

$C(x)$ around a current guess as $C(x) \approx Jx + d$, then forming the resulting linear KKT conditions to find the next guess:

$$\begin{pmatrix} M & J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} M\tilde{x} \\ -d \end{pmatrix}. \quad (3.14)$$

Note that the matrix here is shared with velocity-level approaches, but the right-hand-side is not, and as part of a Newton iteration we end up with a drift-free solution without need for post-stabilization.

3.2.4 Combining Constraints with Potential Energy

Potential energy can easily be combined with the constrained system:

$$x^{(n+1)} = \operatorname{argmin}_{x: C(x)=0} \frac{1}{2}(x - \tilde{x})^T M(x - \tilde{x}) + \Delta t^2 \Phi(x). \quad (3.15)$$

In each Newton step, we can approximate the potential term with a second order Taylor series, giving $\Delta t^2 \Phi(x) \approx \frac{1}{2}x^T Hx + g^T x$ (ignoring the irrelevant constant term), as well as linearizing the constraint, leading to this subproblem:

$$\operatorname{argmin}_{x: Jx+d=0} \frac{1}{2}(x - \tilde{x})^T M(x - \tilde{x}) + \frac{1}{2}x^T Hx + g^T x. \quad (3.16)$$

The KKT conditions produce a modified linear system:

$$\begin{pmatrix} M+H & J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} M\tilde{x} - g \\ -d \end{pmatrix}. \quad (3.17)$$

We could eliminate positions x to get a symmetric positive definite (SPD) problem,

$$J(M+H)^{-1}J^T \lambda = J(M+H)^{-1}M\tilde{x} + d, \quad (3.18)$$

but for many potential energies, including bending, H has an irreducible graph and therefore $(M+H)^{-1}$ is fully dense even if M is only block diagonal, making this approach intractable. We take a slightly different approach to eliminating indefiniteness.

3.2.5 Transforming to a Sparse SPD System with Gauss-Newton

A recent work of Batty and Bridson [3] on incompressible viscous fluids illustrated that a Backwards Euler step of Stokes flow can be represented as the same form of constrained optimization problem, albeit with linear constraints and velocities instead of positions. Also they demonstrated a way to transform the associated symmetric indefinite system into a *sparse* SPD matrix, by formulating the time-scaled viscous stress as damping potential. In this paper, we apply the same technique to our optimization problem and arrive at an SPD system which is generally sparse. Assume, as is usually the case, that potential energy can be expressed as a sum of squares (of potentially nonlinear functions): $\Delta t^2 \Phi(x) = \|r(x)\|_2^2$. For example, our bending model gives:

$$\Delta t^2 B(x) = \frac{1}{2} k_b x^T (L^T M^{-1} L) x = \frac{1}{2} k_b \|M^{-\frac{1}{2}} L x\|^2, \quad (3.19)$$

with nonzero rest angles elided for simplicity. With $\frac{1}{2} \|r(x)\|^2$ denoting the potential energy (and possibly damping potential too), and the minimization problem now is in the form of constrained non-linear least-squares:

$$\operatorname{argmin}_{x: C(x)=0} \frac{1}{2} (x - \tilde{x})^T M (x - \tilde{x}) + t \frac{1}{2} \|r(x)\|^2 \quad (3.20)$$

Viewing $r(x)$ as a “residual”, it’s more sensible to approach this problem with Gauss-Newton than Newton: For example, Gauss-Newton is more robust as Newton can suffer from an indefinite Hessians. Moreover, Gauss-Newton only requires first derivatives, but Newton also needs the second derivatives, which makes for significantly simpler implementation.

For one step of Gauss-Newton, we linearize the residual $r(x) \approx Ax + b$ about the current guess, and similarly for $C(x) \approx Jx + d$, arriving at the following constrained linear least-squares problem:

$$\operatorname{argmin}_{x: Jx+d=0} \frac{1}{2} (x - \tilde{x})^T M (x - \tilde{x}) + \frac{1}{2} \|Ax + b\|^2 \quad (3.21)$$

Introducing a new variable r , we can write the KKT conditions as

$$\begin{aligned}
M(x - \tilde{x}) + A^T r + J^T \lambda &= 0, \\
r &= Ax + b, \\
Jx + d &= 0.
\end{aligned} \tag{3.22}$$

In matrix-vector form, we have a symmetric indefinite matrix:

$$\begin{pmatrix} M & A^T & J^T \\ A & -I & 0 \\ J & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ r \\ \lambda \end{pmatrix} = \begin{pmatrix} M\tilde{x} \\ -b \\ -d \end{pmatrix}. \tag{3.23}$$

Now we eliminate x by solving the first equation, producing a linear system in just r and λ :

$$\begin{aligned}
r &= A(\tilde{x} - M^{-1}A^T r - M^{-1}J^T \lambda) + b \\
J(\tilde{x} - M^{-1}A^T r - M^{-1}J^T \lambda) + d &= 0
\end{aligned} \tag{3.24}$$

In matrix-vector form this is:

$$\begin{pmatrix} I + AM^{-1}A^T & AM^{-1}J^T \\ JM^{-1}A^T & JM^{-1}J^T \end{pmatrix} \begin{pmatrix} r \\ \lambda \end{pmatrix} = \begin{pmatrix} A\tilde{x} + b \\ J\tilde{x} + d \end{pmatrix} \tag{3.25}$$

This matrix is guaranteed to be symmetric positive semi-definite, and strictly definite if J has full rank. Since this follows the same transformations as [3], we expect in future work this will lead to a simple coupling with viscous incompressible fluid flow. Note that Robinson-Mosher et al. [19] have arrived at a similar approach to monolithic fluid-structure interaction, but without the connection to general sum-of-squares energies, nonlinear constraints, Gauss-Newton, etc.

3.3 Collisions and Friction

We process cloth collisions and friction using Bridson et al.'s algorithm [5] with Harmon et al.'s enhancements [12]. This algorithm is robust and independent of the internal cloth dynamics, so it can be easily incorporated into a variety of cloth

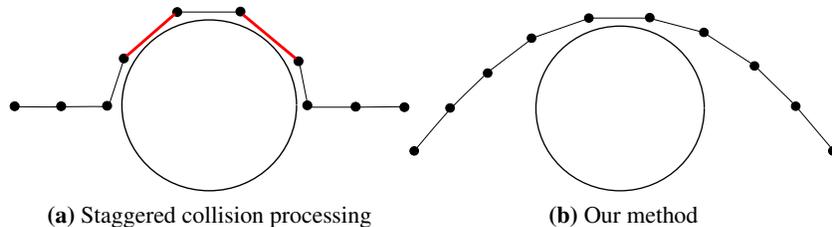


Figure 3.4: 2D illustrations of staggered collision processing and our method simulating stiff material. The staggered version (a) pushed the strand nodes up to non-intersection state, but violated inextensibility constraints (thickened red edges) and the stiffness property. Our method (b) coupled collisions with all the constraints and bending energy and produced the expected result.

dynamics models [8, 10]. In previous cloth solvers, collisions and frictions were processed after solving the internal dynamics, then the non-intersection positions of the cloth were taken as the simulation output of each time step — resolving cloth dynamics and processing collisions were *staggered* in time.

The main problem of staggering is that the collision handling routine may arbitrarily move the cloth positions to the non-intersection states, without knowing about potential energy or inextensibility constraints. This may lead the cloth positions to the states which violate those constraints, i.e. stretched or compressed, or such that large erroronous potential energies are accumulated due to undesired deformation. This problem is illustrated in Figure 3.4, which simplifies the cloth as a 2D strand and a circle as solid object. Suppose we simulate very stiff material, such as a piece of paper or even metal sheet. Figure 3.4b is the physically correct result: the strand is deformed uniformly under gravity. Figure 3.4a is the result generated by staggered collision handling, which is self-intersection free but leads to erroronous stretch and bending on the strand.

3.3.1 Modelling Collisions as Constraints

To solve the problems of staggered collision processing, we incorporate collisions as constraints into our cloth dynamics. The low-level collision algorithm resolves proximities and collisions for the elements which are within the proximity thresh-

Input: $x^{(n)}, v^{(n)}, \Delta t$	// initial position, velocity and time step
1: $\tilde{x} \leftarrow x^{(n)} + \Delta t v_0 + \Delta t^2 M^{-1} F_{\text{ext}}$	// prediction
2: $x_p \leftarrow \mathbf{project}(\tilde{x}, C)$	// projection of cloth constraints C
3: $x_c \leftarrow \mathbf{collision}(x_p, \Delta t)$	// resolve collisions
4: $C' \leftarrow C + C_{\text{collision}}$	// add collisions as constraints to C
5: $x'_p \leftarrow \mathbf{project}(x_c, C')$	// projection including collisions
6: $x^{(n+1)} \leftarrow \mathbf{collision}(x'_p, \Delta t)$	// re-process collisions
7: $v^{(n+1)} \leftarrow (x^{(n+1)} - x^{(n)}) / \Delta t$	// update velocity
Output: $x^{(n+1)}, v^{(n+1)}$	// new position and velocity

Table 3.1: Pseudocode of constraints projection with collision detection

old or intersecting, by applying repulsion forces along the relative normal directions. We then make use of these set of elements and normal directions to formulate our collision constraints, restricting the cloth nodes to only move on the tangent plane of repulsion normals in constraint projection, so the projection step does not pull them closer or push them further. This minimizes (though doesn't altogether eliminate) the chance of introducing new collisions.

For each collision registered at the current time step, we output the type of colliding elements (edge-edge or vertex-triangle), the positions of these elements after applying repulsion forces, and the corresponding normal vector. Regardless of edge-edge or point-triangle collision, we formulate the collision constraints on the cloth nodes (vertices) extracted from these elements.

Suppose the output position from collision detection is x_c and the repulsion force normal is n , we can write the equality constraint function for collision as below:

$$C_{\text{collision}}(x) = (x - x_c)^T \cdot n = 0 \quad (3.26)$$

Our implementation of the constraint projection is illustrated as pseudo-code in Table 3.1. The final collision step in line 5 resolves any new collisions created by the coupled dynamics, to avoid self-intersecting meshes. Since the chances of introducing new collisions in step 5 are minimal, the cost of re-processing collisions is only a small portion of the entire computation time. In step 7, the pseudocode uses Backwards Euler method for determining the new velocity, but just as English

and Bridson [8] extended the Fast Projection to second order accurate BDF2 time integration, we can use BDF2 with our framework by simply changing steps 1 and 7 to the following lines respectively:

$$\begin{aligned}\tilde{x} &\leftarrow \frac{4}{3}x^{(n)} - \frac{1}{3}x^{(n-1)} + \frac{8}{9}\Delta t v^{(n)} - \frac{2}{9}v^{(n-1)} + \frac{4}{9}\Delta t^2 M^{-1} F_{\text{ext}} \\ v^{(n+1)} &\leftarrow \frac{1}{\Delta t} \left[\frac{3}{2}x^{(n+1)} - 2x^{(n)} + \frac{1}{2}x^{(n-1)} \right]\end{aligned}$$

3.3.2 Friction

The above model doesn't include friction, which is essential for most cloth applications. Full Coulomb friction, satisfying a cone constraint on tangential forces with respect to normal forces, isn't immediately applicable. However, we have found a simple approximation that appears to be highly effective in typical scenarios.

Frictional forces are computed using the Coulomb's model implementation in [5]. When tangential movements of cloth-cloth or cloth-solid occur within the proximity threshold, the relative tangential velocity of cloth is updated by:

$$v_t = \max\left(1 - \mu \frac{\Delta v_n}{v_t^0}, 0\right) v_t^0 \quad (3.27)$$

Here μ is the friction coefficient (both static and kinetic), v_t^0 is the relative tangential velocity before applying friction and Δv_n is the change in relative normal velocity.

This friction algorithm works fine alone in the staggered collision process, since the final output cloth positions have directly taken the tangential velocity in (3.27). However, combining this with the collision constraint (3.26) leads to a problem — the constraint projection has so much freedom to adjust positions on the tangent plane that it may cancel out the relative movements applied by frictional forces.

In order to preserve the frictions, we need to append two constraints on the tangent plane in addition to the normal direction:

$$\begin{aligned}
C_{t1}(x) &= (x - x_c)^T \cdot t_1 = 0, \\
C_{t2}(x) &= (x - x_c)^T \cdot t_2 = 0.
\end{aligned}
\tag{3.28}$$

3.3.3 Friction Regularization

Particularly with the addition of tangential constraints, we often find that J is rank-deficient or nearly deficient, indicating incompatibilities between the various constraints, and causing troubles for convergence. Even if not, the tangential friction constraints shouldn't be treated as hard constraints in the face of much stronger forces from isometry or stiff bending. Therefore, we need to regularize our problem, and in particular soften the tangential constraints — we want to encourage friction but not at the expense of stronger effects.

After adding the Lagrange multipliers explicitly to our minimization problem and adding a standard regularization term penalizing large Lagrange multipliers, equation (3.12) is modified to:

$$\begin{aligned}
x^{(n+1)} = \operatorname{argmax}_{\lambda} \left[\operatorname{argmin}_x \frac{1}{2}(x - \tilde{x})^T M(x - \tilde{x}) + \Delta t^2 \Phi(x) \right. \\
\left. - C(x)^T \lambda - \frac{1}{2} \lambda^T D \lambda \right]
\end{aligned}
\tag{3.29}$$

Going through the derivations of SPD system in 3.2.5, from the KKT conditions of (3.29) we get the following linear system:

$$\begin{pmatrix} I + AM^{-1}A^T & AM^{-1}J^T \\ JM^{-1}A^T & JM^{-1}J^T + D \end{pmatrix} \begin{pmatrix} r \\ \lambda \end{pmatrix} = \begin{pmatrix} A\tilde{x} + b \\ J\tilde{x} + d \end{pmatrix}
\tag{3.30}$$

The non-negative diagonal regularization matrix D introduces a small perturbation to the original problem, but this generally helps to arrive at a well-posed problem when faced with incompatible constraints. In addition, it allows us to increase the D terms for the tangential friction constraints, so they may be overruled by the hard constraints. In our examples we only found it necessary to put nonzeros in D in the rows corresponding to tangential collision constraints, though in general any

constraint could be softened this way. We chose the values of diagonals in D by experimentation, and found that a wide range from 10^{-5} to 10^{-2} serves well.

Chapter 4

Results

We ran our simulations on Intel 2.0GHz Core 2 Duo with 4GB of RAM, and used PARDISO [20, 21] as the linear solver.

In order to evaluate the effectiveness of our simultaneous collision handling, we applied high bending stiffness to the surface and simulated the mesh draping on a ball as shown in Figure 4.1.

Using the staggered collision detection, the mesh vertices were stretched to the non-intersection positions, without respect to the inextensibility constraints or material's bending property, which caused spurious potential energy to accumulate at the end of frame. This potential energy was propagated to the following frames in the system and degraded the dynamics afterwards. The measurements of maximum strain were shown in Figure 4.2 and the staggered version resulted in oscillating stretch forces as we expected. Moreover, we plot the bending energy stored by the end of each frame in Figure 4.3, and observed similar spurious oscillation.

Although our simultaneous collision handling algorithm increases simulation time by an extra step of constraint projection and collision detection (step 5 and 6 in Table 3.1, we only observed minor impact on the overall performance. The differences were measured and shown by Figure 4.4. In this experiment our algorithm even converged faster than the staggered method for some time steps. One explanation is that the coupled constraint projection avoided propagating spurious potential energies to the following time steps, therefore the Newton solver required

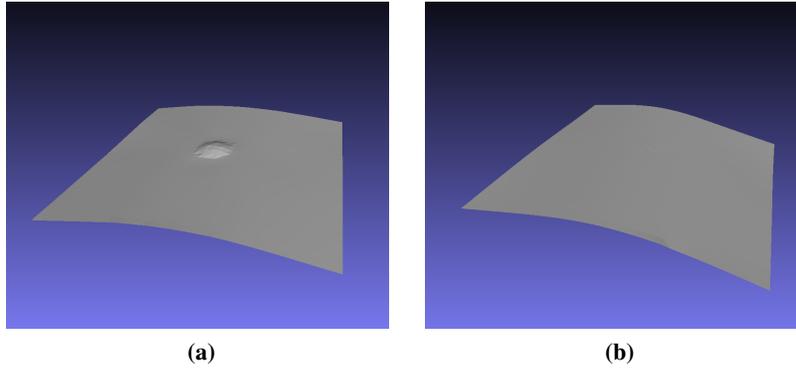


Figure 4.1: Comparison of a stiff surface bending simulation using staggered collision detection (left) and simultaneous collision detection (right)

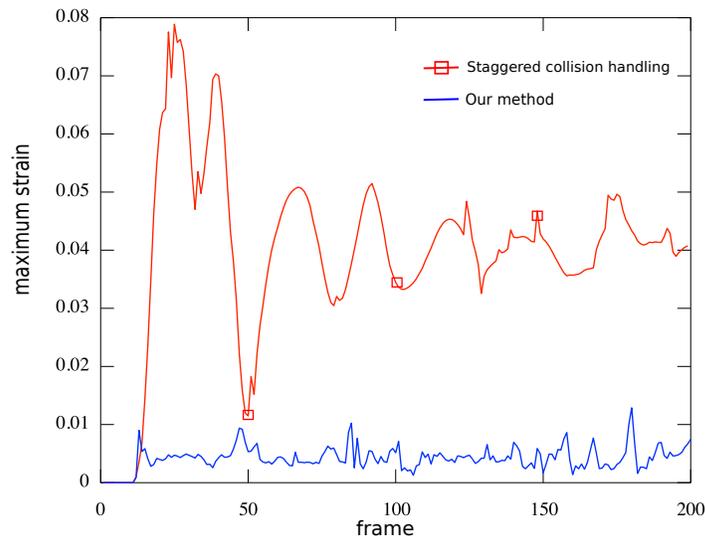


Figure 4.2: Inextensibility comparison between staggered collision handling and our method. Running 200 frames of the stiff surface bending simulation on a sphere, we measured the maximum strain of the original mesh edges.

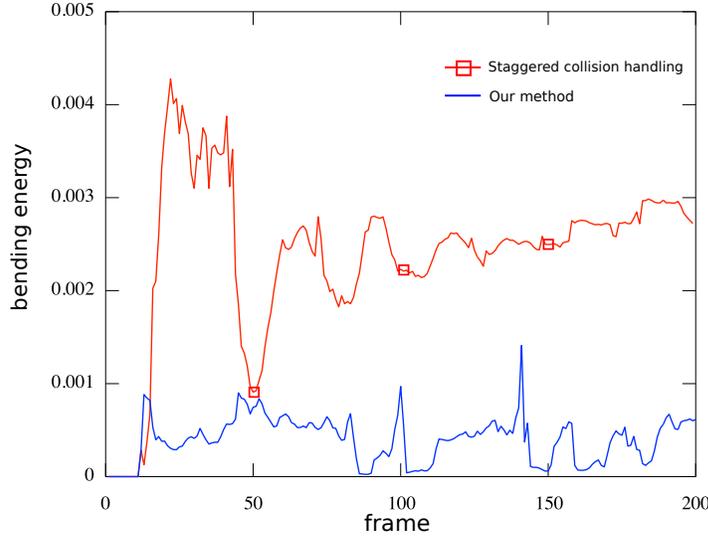


Figure 4.3: Bending energy comparison between staggered collision handling and our method. With the same stiff surface bending simulation, we plot the bending energies accumulated by the end of each frame from two methods.

Model	Number of triangles	Bending stiffness	Strain limit	Cost per time step	Steps per frame
cloth draping	20k	10	0.1 %	18.7s	8
shirt	10k	10^2	1 %	30.8s	8
Armadillo	10k	$10^3 - 10^5$	10 %	18.1s	1
Armadillo (rigid)	10k	10^6	10 %	22.5s	4

Table 4.1: Simulation performances in different scenarios

fewer iterations to reach the optimal solution.

The detailed simulation data used by our experiments are listed in Table 4.1. Figure 4.5 shows some selected frames from the cloth draping on a rotating ball. The cloth was simulated with a mesh of 20,000 triangles divided on a 100x100 grid. The tolerance of strain was set to 0.1%. The simulation ran with 8 steps per frame at for a playback rate of 100 frames/sec, where each time step took 18.7 seconds on average to compute.

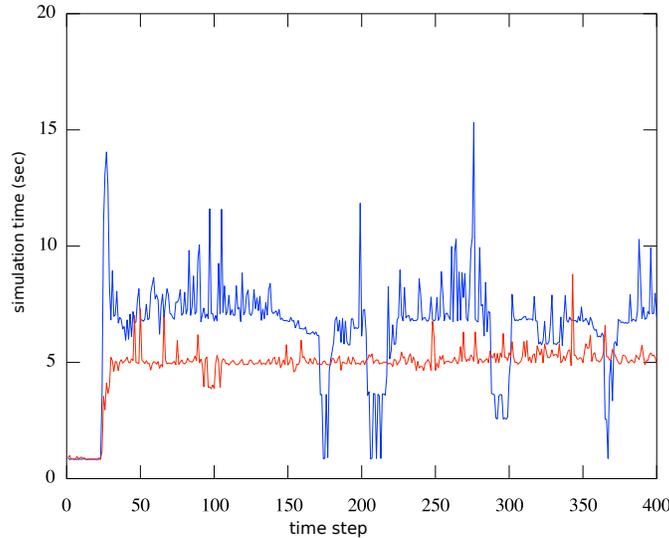


Figure 4.4: Comparison of simulation time using staggered collision handling (red) and our method (blue)

Our experiments with non-zero rest angles are illustrated by Figure 4.6 and 4.7. We did simulation of thin shell with the Armadillo mesh collapsing under gravity and the model had resolution of 10,000 triangles. We simulated at 100 frames/sec and 1 time step per frame, and experimented bending stiffness ranging from 10^3 to 10^5 . The stretch tolerance was increased to 10% since we were simulating rubber-like material. Our implicit integration scheme performed robustly with very weak and strong bending forces.

Frames in Figure 4.6 showed that the low stiffness model collapsed like a deflated balloon. Figure 4.7 illustrated that stronger bending forces were able to better retain the shape of the model, especially on sharp features. The average computation time of these simulations was 18.1 seconds per frame. We then increased the bending stiffness to 10^6 , and reduced the time step size down to 4 steps per frame at 50 frames/sec for better collision detection. The mesh collided to the floor with minimal deformations almost like a rigid body.

The walking sequence shown in Figure 4.8 simulated with a shirt mesh containing 10,000 triangles. Due to relatively low frame rate of the given sequence

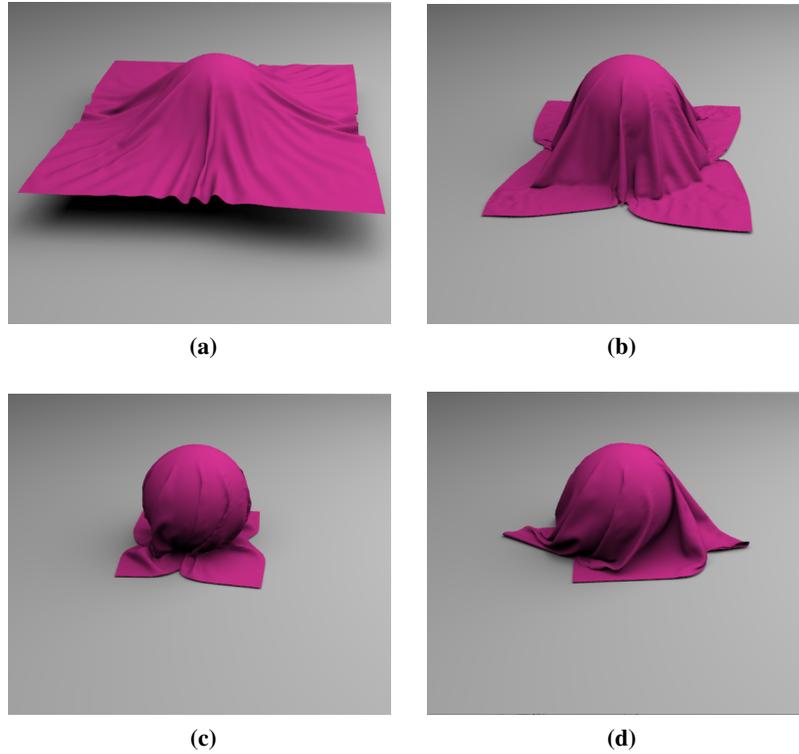
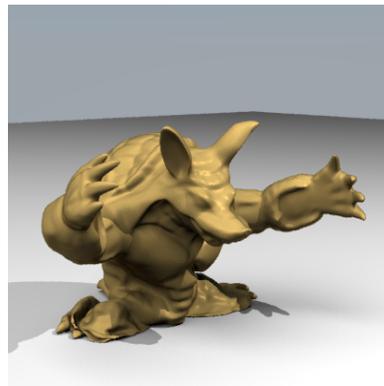


Figure 4.5: Cloth draping on a rotating ball

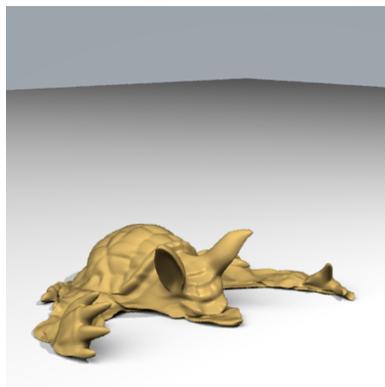
data (30 frames/sec), each frame was computed with 8 time steps at 30.8 seconds each. Our method handled the frictional contacts and complex collisions very well, and produced realistic wrinkles between the arms and body. In all our simulations, the most expensive computation was the projection of constraints and potential energy. In most cases the projection algorithm took less than 20 Newton iterations to converge. The time spent on each SPD solve largely depends on the problem size (i.e. number of nodes and constraints).



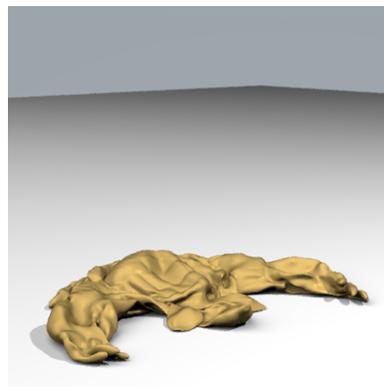
(a)



(b)



(c)

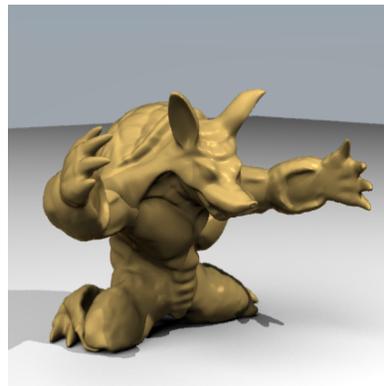


(d)

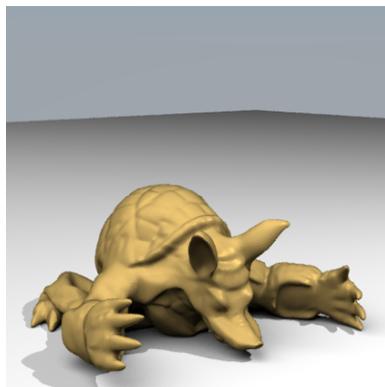
Figure 4.6: Armadillo model collapsing with low bending stiffness $k_b = 5.0 \times 10^3$



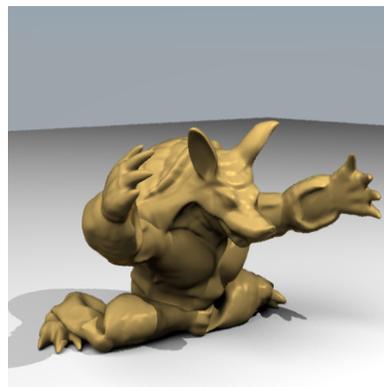
(a)



(b)



(c)



(d)

Figure 4.7: Armadillo model collapsing with high bending stiffness $k_b = 5.0 \times 10^4$



(a)



(b)



(c)



(d)

Figure 4.8: Walking model dressed with a shirt

Chapter 5

Conclusions

We have presented a continuous discretization model for simulating inextensible surfaces. With our new methods of enforcing inextensibility, conforming and boundary constraints, the surface model is isometrically deforming and non-locking. Thanks to the continuity of this model, it can be directly plugged into existing collision handling algorithms with little effort. We also proposed an approach to integrate frictional contact and collisions into the constrained dynamics, such that our method overcame the disadvantages of staggered collision handling, and produced high quality simulation results for soft and highly stiff materials. Moreover, we demonstrated a way to approximate the rotation-invariant quadratic bending energy form for non-zero rest angles.

We introduced an efficient fully implicit integration scheme combined with all constraints and general potential energies. Our integration framework offers drift-free position level constraints projection, allows larger time steps and handles extremely stiff bending forces.

Our current treatment of friction proved adequate for our examples, but doesn't fully capture Coulomb friction. In future work we hope to incorporate an exact solve of fully coupled tangential friction forces obeying a friction cone constraint by careful adjustments to the associated regularization parameters in the style of trust region methods for Gauss-Newton solution of nonlinear least-squares, i.e. the Levenberg-Marquardt algorithm [16]. We have also extended the model to include nonlinear damping potentials in the objective function, but are still working on full

second order accuracy for damping forces with BDF2. As we mentioned above, this model also is perfectly compatible with inviscid and viscous fluid flow, so we look forward to a fully implicit, fully coupled solver of all solid and fluid physics which still reduces to a sequence of SPD linear solves.

Bibliography

- [1] D. Baraff. Linear-time dynamics using lagrange multipliers. In *Proc. ACM SIGGRAPH*, pages 137–146, 1996. → pages 2
- [2] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proc. ACM SIGGRAPH*, pages 43–54, 1998. → pages 2, 3
- [3] C. Batty and R. Bridson. A simple finite difference method for time-dependent, variable coefficient Stokes flow on irregular domains. *ArXiv e-prints*, Oct. 2010. → pages 12, 13
- [4] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun. A quadratic bending model for inextensible surfaces. In *Symp. Geometry Processing*, pages 227–230, 2006. → pages 7
- [5] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 21(3):594–603, 2002. → pages 2, 3, 13, 16
- [6] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Symp. Comp. Anim.*, pages 28–36, 2003. → pages 7
- [7] M. B. Cline and D. K. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *ICRA*, pages 3744–3751, 2003. → pages 2
- [8] E. English and R. Bridson. Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.*, 27(3), 2008. → pages 1, 2, 3, 6, 14, 16
- [9] A. Garg, E. Grinspun, M. Wardetzky, and D. Zorin. Cubic Shells. In *Symposium on Computer Animation*, pages 91–98, Aug 2007. → pages 8
- [10] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):49, 2007. → pages 2, 10, 14

- [11] E. Grinspun, A. Hirani, M. Desbrun, and P. Schrder. Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 62–67, Aug 2003. → pages 7
- [12] D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun. Robust Treatment of Simultaneous Collisions. *SIGGRAPH (ACM Transactions on Graphics)*, 27(3):1–4, 2008. → pages 13
- [13] D. House, R. W. Devaul, and D. E. Breen. Towards simulating cloth dynamics using interacting particles. *International Journal of Clothing Science and Technology*, 8:75–94, 1996. → pages 2
- [14] G. Irving, C. A. Schroeder, and R. Fedkiw. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph.*, 26(3):13, 2007. → pages 3
- [15] Y.-J. Liu, K. Tang, and A. Joneja. Modeling dynamic developable meshes by the hamilton principle. *Comput. Aided Des.*, 39(9):719–731, 2007. ISSN 0010-4485. doi:<http://dx.doi.org/10.1016/j.cad.2007.02.013>. → pages 2
- [16] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. Watson, editor, *Lecture Notes in Mathematics 630 – Numerical Analysis*, pages 105–116, 1978. → pages 27
- [17] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. H. Gross. Implicit contact handling for deformable objects. *Comput. Graph. Forum*, 28(2): 559–568, 2009. → pages 3
- [18] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *GI*, pages 147–154, 1995. → pages 2
- [19] A. Robinson-Mosher, C. A. Schroeder, and R. Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Physics*, 230(4):1547–1566, 2011. → pages 13
- [20] O. Schenk, A. Wächter, and M. Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Comput. Optim. Appl.*, 36(2-3):321–341, April 2007. → pages 19
- [21] O. Schenk, M. Bollhöfer, and R. A. Römer. On large scale diagonalization techniques for the Anderson model of localization. *SIAM Review*, 50(1): 91–112, 2008. SIGEST Paper. → pages 19

- [22] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun. Discrete quadratic curvature energies. *Computer Aided Geometric Design*, pages 499–518, 2007. → pages 8