Projective Peridynamic Modeling of Hyperelastic Membranes with Contact

Zixuan Lu
Institute of Software, CAS &
University of Chinese Academy of Sciences
Beijing, China
luzx@ios.ac.cn

Xiaowei He
Institute of Software, CAS
Beijing, China
xiaowei@iscas.ac.cn

Yuzhong Guo
Institute of Software, CAS
Beijing, China
guoyuzhong@iscas.ac.cn

Xuehui Liu
Institute of Software, CAS &
University of Chinese Academy of Sciences
Beijing, China
lxh@ios.ac.cn

Abstract

Real-time simulation of hyperelastic membranes such as cloth still face a lot of challenges, such as hyperplasticity modeling and contact handling. In this study, we propose projective peridynamics that uses a local-global strategy to enable fast and robust simulation of hyperelastic membranes with contact. In the global step, we propose a semi-implicit strategy to linearize the governing equation for hyperelastic materials which are modeled with peridynamics. By decomposing the first Piola-Kirchhoff stress tensor into a positive and a negative part, successive substitutions can be taken to solve the nonlinear problems. Convergence is guaranteed by further addressing the overshooting problem. Since our global step solve requires no energy summation and dot product operation over the entire problem, it fits into GPU implementation perfectly. In the local step, we further present a GPU-friendly gradient descent method to prevent interpenetration. Putting the global and local solves together, experiments show that our method is robust and efficient in simulating complex models of membranes involving hyperelastic materials and contact.

***Keywords:** hyperelastic membranes, projective peridynamics, semi-implicit successive substitution method, contact handling*

1. Introduction

How to animate hyperelastic membranes with contact at interactive rates or even in real-time has been a long-standing challenge for computer graphics. To ensure the simulation efficiency, position-based dynamics [25, 3] simplified inner forces as equality or inequality constraints, which can be largely sped up via parallel implementations on GPU. However, position-based methods are associated with a problem that simulation results are controlled by the number of iterations and the mesh resolution, rather than a model with physical meanings [17, 23, 9]. As a result, classical hyperelastic models cannot be directly supported by position based dynamics in general. Certain reformulation or simplification should be done, e.g., by reformulating the energy density of neo-Hookean models into compliant constraints [19]. As an alternative method for real-time applications, projective dynamics [4] shows a great potential in bridging the gap between accuracy and efficiency. Since the original form of projective dynamics requires a special quadratic form of elastic potential energies, most hyperelastic materials are not supported. To model more complex materials, He et al. [9] has recently proposed the corotational formulation of peridynamics based on projective dynamics, providing an attractive way for the simulation of elastoplastic bodies exhibiting linear elastic response. However, their method cannot be easily extended to simulate materials with nonlinear elastic response.

In terms of simulating hyperelastic materials, the Newton or quasi-Newton methods are most commonly used. For example, Liu et al. [18] proposed a quasi-Newton method to simulate hyperelastic materials with the insight that projective dynamics can be reformulated as a quasi-Newton method. To guarantee the convergence, the notorious line search should be incorporated as a necessity to ensure stability. Its formal expression can be stated as to find a sufficient small step length λ for an objective function satisfying the following Armijo condition [1]

$$f\left(x_k + \alpha p_k\right) \leq f\left(x_k\right) + \lambda\alpha\nabla_k^T p_k, \tag{1}$$

where $f(\cdot)$ is the objective function, p_k is a descent direction, α is called the step length and $\lambda \in (0, 1)$ is a constant that controls the step length size. However, the Armijo condition is not enough by itself to ensure that the algorithm makes reasonable progress because it is satisfied for all sufficiently small values of α [27]. In our practical implementation, we find the right choice for λ not only depends on $f(x_k)$, but also on how to choose the descent direction p_k. The optimal value for λ may even change from iterations to iterations. The dramatic demand for line-search iterations could cause a sudden slow-down to the simulation, which is a nightmare for real-time applications. Another difficulty that we encounter with the Armijo condition is that it requires dot product operations, making it rather inefficient for implementation on GPU. To reduce the computational cost, Wang and Yang [43] proposed to eliminate the dot product by setting $\lambda = 0$ and do less line search by reusing values of α. However, both the problem with the Armijo condition and the one with the expensive energy summation remain unresolved.

Another difficulty in simulating hyperelastic membranes is how to model contacts. Since real world contacts are typically discontinuous, they make the contact problems very stiff. To decrease the stiffness of contact problems, the Incremental Potential Contact (IPC) solver [15] introduced smooth log-barrier functions to model the contacts. However, when hyperelasticity and contacts are solved together, the diversity in problem stiffness can quickly degenerate the overall simulation performance.

In this paper, we present projective peridynamics that uses a local-global strategy to enable fast and robust simulation of hyperelastic membranes with contact. We solve the material hyperelasticiy in the global step while the contact problem in the local step. As a result, different numerical methods can be used to solve material hyperelasticity and contact independently by fully exploiting features in each problem.

Our technical contributions can be summarized as

- A projective model based on peridynamics to simulate hyperelastic membranes with contact.
- A semi-implicit successive substitution method to efficiently simulate hyperelastic materials in the global step.
- A gradient descent method to solve contacts in the local step.

2. Related work

2.1. Peridynamics

As a nonlocal theory, peridynamics was formally established by Silling [34] to model problems with discontinuities, e.g., fracture simulation. The original form of the peridynamic model is called the bond-based theory, which relies on the assumption that the interaction forces between a pair of particles are equal in magnitude and parallel to the relative position vector in the deformed state [35, 6]. The problem with the bond-based peridynamics is that it is not possible to define a Poisson's ratio because it only captures a constant Poisson's ratio of 0.33 for two-dimensional and 0.25 for three-dimensional problems [31, 22, 20]. To address this issue, a generalized formulation referred to as state-based peridynamics was proposed [36]. The core in the generalization is to introduce a mathematical tool called *state*, which is able to model the response of a material at a point that depends collectively on the deformation of all bonds connected to the point. For example, a *deformation state* can be defined to map any bond onto its image under the deformation. Within state-based peridynamics, Silling and Lehoucq [37] proved that the elastic peridynamic model converges to the classical model in the limit of small horizon under certain assumptions. Lehoucq [14] also derived a notion of a peridynamic stress tensor from nonlocal interactions. That is to say, peridynamics is compatible with finite element approximations in a certain function space. On the other hand, given a hyperelastic model in classical elasticity theory, a state-based peridynamics model can also be derived based on the classical constitutive model. This has the advantage of enabling the description of the deformation/damage process in peridynamics according to well-established models [41]. For a recent survey on deformation/damage modeling based on peridynamics, please refer to [11].

2.2. Hyperelasticity solver

In the classical theory, finite element method (FEM) is the most widely used method to simulate deformable objects. Early researchers in computer graphics have mainly focused on simulating linear elastic models due to its simplicity [7, 24, 21]. However, for materials in the real world, the linear elastic models usually can not accurately describe the observed elastic behavior. A hyperelastic model is an alternative option to describe the nonlinear stress–strain relationship. To capture nonlinear elastic behaviors, Wu et al. [45] proposed adaptive nonlinear FEM with mass lumping to achieve real time simulation of a 3D liver. However, they only use an explicit method to take time integration as the implicit method is computationally intensive and may have poor convergence. Irving et al. [10] presented an algorithm for the finite element simulation of elastoplastic solids with explicit integration for the elastic forces. To alleviate the stringent time step restrictions imposed by the explicit integration method, Teran et al. [40] proposed to solve nonlinear elastic materials using a modified Newton-Raphson algorithm, which is computationally intensive. Xu et al. [46] proposed to formulate the strain energy density

function using three principle stretches and solve nonlinear elasticity with an implicit integration method, allowing larger timestep and more stable performance compared to the explicit method [10].

Different with force based method, position based dynamic [25, 3] solves motion and dynamic problem by reformulating generalized constraint with particles' position. This constraint projection based method is generalized to projective dynamic method [4]. Projective dynamic method decouples hyperelasticity energy density as a rest state manifold constraints and distance measure, allowed constraints independently being carried out. Wang and Yang [43] further proposed a new gradient descent method using Jacobi preconditioning and Chebyshev acceleration to accelerate the convergence rate base on GPU for nonlinear elasticity solving, obtaining real time simulation of a variety of different hyperelastic materials. Rahul et al. [26] extended projective dynamics to model a broad range of objective functions including nonlinear models and hard constraints. After showing that projective dynamics can be interpreted as a quasi-Newton method, Liu et al. [18] proposed a quasi-Newton solver equipped with a line search to simulate hyperelastic materials. To accelerate the convergence rate, Peng et al. [30] proposed to apply Anderson acceleration to speed up the convergence of a local-global solver. Due to the similarity of our method to a fixed-point method, the Anderson acceleration can directly applied to our method to help accelerate the convergence rate.

2.3. Contact handling

For shell-like models, solving contact and self-collision fastly and robustly still remains to be a great challenge. Provot [32] provided a collision handling framework which treats contact zone as a rigid body and gived a continuous collision detection algorithm for triangle mesh based discretization. Bridson et al. [5] combined impulses method and penalty method which introduce impulse for penetrated triangle pair to separate them away and introduce repulsion force to remove close proximity for potential penetration. Previous work also treated self-collision as a constraint problem [29, 4]. Recently, Tang et al. [39] developed a GPU-friendly impact zone solver by solving CCD culled constrained optimization problem with augmented Lagrangian method. Wu et al. [44] further developed a repulsion method on GPU, which allows large time step and be free of Lagrangian multiplier, so that large but inexact iterative step is kept safe. Li et al. developed a set of contact solver based on Incremental Potential Contact algorithm [15, 16], which handle contact in a penalty way by introducing contact barrier energy.

3. Overview

For the sake of completeness, we first present a brief introduction to projective dynamics at the beginning of this section, and then give more details on how to model hyperelastic membranes with projective peridynamics. With an implicit Euler time integration for a deformable object, its variational form can be derived as [4]

$$\arg\min_{\mathbf{y}} \frac{1}{2h^2} \left\|\mathbf{M}\left(\mathbf{y}-\mathbf{y}^*\right)\right\|_F^2 + \Psi\left(\mathbf{y}\right), \tag{2}$$

where h is the time step size, $\mathbf{y}$ denotes the position of all vertices (we use $\mathbf{y}$ instead of $\mathbf{x}$ to make it consistent with the description of peridynamics), $\mathbf{M}$ is the mass matrix, $\mathbf{y}^* = \mathbf{y}^t + h\dot{\mathbf{y}}^t + h^2\mathbf{M}^{-1}\mathbf{f}^{ext}$ is the intermediate position calculated from the previous position $\mathbf{y}^t$, velocity $\dot{\mathbf{y}}^t$ and the external force $\mathbf{f}^{ext}$. Ψ represents a summation of elastic potentials that can account for deformations of stretching, bending, shearing, etc. In modeling hyperelastic materials, Ψ may contain nonlinear terms, therefore the time-consuming Newton-type methods are typically used to solve the nonlinear optimization problem. To improve the performance, projective dynamics proposes to separate a nonlinear objective function into the summation of a quadratic part and non-quadratic constraints $C_i(\mathbf{y}) = 0$. The optimization of Eq. (2) can then be solved iteratively in a local-global alternating manner, i.e., the quadratic part is solved with a global iterative solver while all non-quadratic constraints are solved in a local manner.

Let us consider a hyperelastic material model that can handle contact and self-collision, and its total potential energy is formulated as

$$\arg\min_{\mathbf{y}} \frac{1}{2h^2} \left\|\mathbf{M}\left(\mathbf{y}-\mathbf{y}^*\right)\right\|_F^2 + \Psi\left(\mathbf{y}\right) + B\left(\mathbf{y}\right), \tag{3}$$

where $B\left(\mathbf{y}\right)$ represents the barrier potential used to penalize the collision [15]

$$B(d) = \begin{cases} -(d-\hat{d})^2 \log(\frac{d}{\hat{d}}), & 0 < d_k < \hat{d} \\ 0, & d \geq \hat{d} \end{cases}. \tag{4}$$

Here $\hat{d}$ is a user-defined threshold to prevent interpenetration, d is a function of $\mathbf{y}$ representing the closest distance for each pair of primitives. Our question is how to solve the above optimization problem with a projective method? Since both $\Psi\left(\mathbf{y}\right)$ and $B\left(\mathbf{y}\right)$ are not quadratic, both of these two terms should be solved in a local step. However, simply combining $\Psi\left(\mathbf{y}\right)$ and $B\left(\mathbf{y}\right)$ together in the local step does not produce physically correct results. Lan et al.[13] has already identified two major challenges include the sticking and the jittering problems. To address these challenges, we

propose to reformulate Eq. (3) as

$$\arg\min_{\mathbf{y}} E(\mathbf{y},\dot{\mathbf{y}}) + \Psi(\mathbf{y}) + \sum_i \frac{w_i}{2} \|\mathbf{A}_i\mathbf{S}_i\mathbf{y} - \mathbf{B}_i\mathbf{z}\|^2, \quad s.t.\ B(\mathbf{z}) = 0, \tag{5}$$

where $\mathbf{z}$ represent the auxiliary variables, $\mathbf{A}_i$ and $\mathbf{B}_i$ represent the coefficient matrices and $\mathbf{S}_i$ is the constant selection matrix. A local-global strategy is also used to solve the above constrained optimization problem. However, the difficulty lies in how to solve the global step if the objective function in Eq. (5) is not quadratic anymore. Our solution is to propose a substitution-type method that can both solve nonlinear optimization efficiently and fit into the peridynamics framework well.

Algorithm 1 describes an overview of our local-global strategy in simulating hyperelastic membranes with contact. Here, Line 7∼8 represent one iteration of solving material hyperelasticity in the global step while Line 11 represents contact handling solved in the local step. More details on how to discretize Eq. 5 are given in the following context.

Algorithm 1: Projective Peridynamics

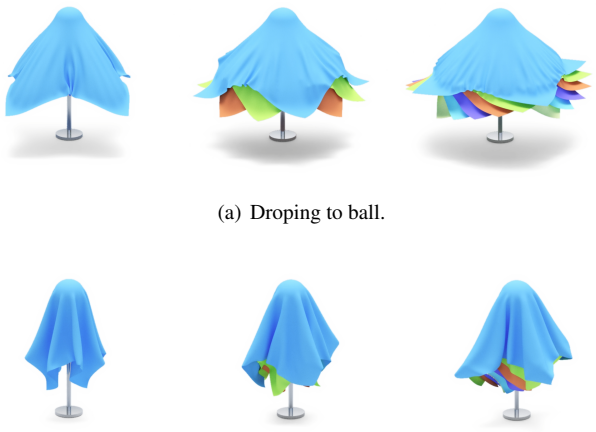
1 Input $\mathbf{y}^t$,$\mathbf{v}^t$,$\xi, h, \varepsilon, \hat{d}, s_0, k_b$
2 $\mathbf{y}^{k=0} \leftarrow \mathbf{y}^t + h\mathbf{v}^*$
3 $\mathbf{z}^{k=0} \leftarrow \mathbf{y}^t$
4 //We typically set $eps = 1e^{-4}$
5 **while** $\max_i\left(\left\|\mathbf{y}_i^k - \mathbf{y}_i^{k-1}\right\|_2\right) > eps$ *and* $k \leq$ *max_iter* **do**
6 **foreach** *vertex* i **do**
7 Calculate $\mathbf{A}_i^k, \mathbf{s}_i^k, \mathbf{s}_i^t$ //Eq. (25)
8 $\mathbf{y}_i^{k+1}$ = Jacobi($\mathbf{A}_i^k, \mathbf{s}_i^k, \mathbf{s}_i^t$) //Eq. (24)
9 **end**
10 Find active contact pairs.
11 $\mathbf{z}^{k+1}$ = Project($\mathbf{y}^{k+1}, \mathbf{z}^k, \xi, h, \varepsilon, \hat{d}$) //Algorithm 2
12 $\mathbf{y}^{k+1} \leftarrow \mathbf{z}^{k+1}$
13 **end**
14 $\mathbf{y}^{t+1} \leftarrow \mathbf{y}^k$

3.1. Basic theory of peridynamics

In the peridynamic model, let $\mathbf{y}(\mathbf{x},t)$ be the deformed position at time t of the vertex $\mathbf{x}$ in the reference configuration of a region $\mathcal{B}$. By applying an implicit integration, the deformation of a vertex i from time t to $t+1$ is written as

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + h\mathbf{v}_i^{t+1}, \qquad \mathbf{v}_i^{t+1} = \mathbf{v}_i^* + h\mathbf{M}_i^{-1}\mathbf{f}_i^{t+1}, \tag{6}$$

in which $\mathbf{v}_i$ is the velocity, $\mathbf{v}_i^*$ is the intermediate velocity calculated as $\mathbf{v}_i^* = \mathbf{v}_i^t + h\mathbf{M}_i^{-1}\mathbf{f}_i^{ext}$, $\mathbf{M}_i$ is the mass matrix (i.e., $\mathbf{M}_i = m_i\mathbf{I}$), h is the time step, and $\mathbf{f}_i$ is the total internal force exerted on vertex i. Unlike FEM, the force $\mathbf{f}_i$ in



(a) Droping to ball.

(b) Static frame.

Figure 1. Muti-layer elastic cloths drop over default gravity. Single layer(left), three layers(middle) and six layers(right) situation are being tested.

peridynamics is modeled as an integration over a neighborhood of $\mathbf{x}_i$. More specifically, $\mathbf{f}_i$ can be descretized as

$$\mathbf{f}_i = V_i \sum_j \left\{\underline{\mathbf{T}}_i\langle\mathbf{x}_j - \mathbf{x}_i\rangle - \underline{\mathbf{T}}_j\langle\mathbf{x}_i - \mathbf{x}_j\rangle\right\} V_j, \tag{7}$$

in which $\underline{\mathbf{T}}$ is the *force state* describing the interaction force between $\mathbf{x}_i$ and $\mathbf{x}_j$, and V is the volume. Silling and Lehoucq [37] demonstrate that if the deformation is sufficiently smooth, the peridynamic stress tensor converges to a Piola-Kirchhoff stress tensor that is a function only of the local deformation gradient tensor, as in the classical theory. Therefore, suppose we have an expression for a stress tensor in the classical theory, a peridynamic constitutive model that is consistent with the model is derived as

$$\underline{\mathbf{T}}_i\langle\mathbf{x}_j - \mathbf{x}_i\rangle = \omega_{ij}\mathbf{P}_i\mathbf{K}_i^{-1}(\mathbf{x}_j - \mathbf{x}_i), \tag{8}$$

where $\mathbf{P}$ represents the first Piola-Kirchhoff stress which is a function of the deformation gradient tensor $\mathbf{F}$

$$\mathbf{F} = \left(\sum_j \omega_{ij}(\mathbf{y}_j - \mathbf{y}_i)(\mathbf{x}_j - \mathbf{x}_i)^T\right)\mathbf{K}_i^{-1}, \tag{9}$$

ω is a *scalar state* acting as a weighting function, $\mathbf{K}_i$ is *the shape tensor* calculated as

$$\mathbf{K}_i = \sum_j \omega_{ij}(\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T. \tag{10}$$

By combining Eq. (6) and (7), we obtain the following system

$$m_i\left(\mathbf{y}_i^{t+1} - \mathbf{y}_i^t - h\mathbf{v}_i^*\right) = h^2V_i\sum_j\left\{\underline{\mathbf{T}}_i^{t+1}\langle\mathbf{x}_j - \mathbf{x}_i\rangle - \underline{\mathbf{T}}_j^{t+1}\langle\mathbf{x}_i - \mathbf{x}_j\rangle\right\}V_j. \tag{11}$$

3.2. Kirchhoff-Love plate formulation

In simulating membranes that are modeled with only one layer of triangles, it can be noted from Eq. (10) that the shape tensor is not guaranteed to be positive definite. In other words, the standard form of peridynamics cannot handle degenerate configurations for codimensional materials as the singular shape tensor may cause the simulation to blow up. To address this limitation, we propose to reformulate both the shape tensor and deformation tensor based on the Kirchhoff–Love plate theory [33]. The Kirchhoff–Love plate theory is a mechanical model and a set of assumption used to represent a three-dimensional thin plate (or membrane) in two dimensional reference (named as *mid-surface* in the following discussion, which is a reference surface laid in middle between upper and lower surfaces of the plate). Three kinematic assumptions are used in this theory, i.e., straight lines normal to the mid-surface of the undeformed plate remain **straight**, **normal to the mid-surface**, and **unstretched** after deformation. The length of this normal between mid-surface and plate upper or lower surface is modeled as thickness. Therefore, we are able to extend the basic theory of peridynamics to describe codimensional structures.

We introduce a virtual bond that is normal to both the undeformed and deformed mid-surfaces, denoted as $\mathbf{x}_i^{\perp}$ and $\mathbf{y}_i^{\perp}$, respectively

$$\mathbf{x}_i^{\perp} = \xi \text{norm}\left(\frac{\sum_J \theta_J \mathbf{n}_J}{\sum_J \theta_J}\right), \mathbf{y}_i^{\perp} = \xi \text{norm}\left(\frac{\sum_J \theta_J^* \mathbf{n}_J^*}{\sum_J \theta_J^*}\right), \tag{12}$$

where ξ represents the thickness of the membrane measured from surface to the mid-surface; J is the index of all neighboring triangles; θ_J is inner angle of triangle J with apex of vertex i; $\mathbf{n}_J$ is the triangle normal, and norm$(\cdot)$ is used to normalize a vector. The superscript $*$ for θ and $\mathbf{n}$ is used to denote quantities in the deformed configuration. With virtual bonds introduced, the shape tensor $\mathbf{K}$ can be defined as

$$\mathbf{K}_i = \sum_j \omega_{ij}(\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T + \mathbf{x}_i^{\perp} \otimes \mathbf{x}_i^{\perp}, \tag{13}$$

where $\otimes$ represents the Kronecker product. Similarly, the deformation gradient tensor $\mathbf{F}$ can be reformulated as

$$\mathbf{F}_i = \left(\sum_j \omega_{ij}(\mathbf{y}_j - \mathbf{y}_i)(\mathbf{x}_j - \mathbf{x}_i)^T + \mathbf{y}_i^{\perp} \otimes \mathbf{x}_i^{\perp}\right) \mathbf{K}_i^{-1}. \tag{14}$$

Note $\mathbf{x}_i^{\perp}$ is defined as the orthogonal complement of all neighboring bonds $\langle \mathbf{x}_j - \mathbf{x}_i \rangle$, the shape tensor $\mathbf{K}_i$ is guaranteed to be nonsingular. As a result, the value of the deformation gradient $\mathbf{F}_i$ can be correctly calculated from Eq. (14).

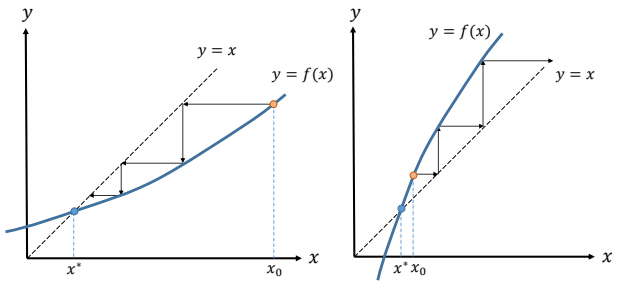


Figure 2. An illustration of the standard successive substitution method.

After extending peridynamics to model membranes, we consider to solve Eq. (11). Note $\mathbf{T}$ is a nonlinear function of the deformation gradient tensor in hyperelastic models, how to solve above nonlinear system is a real challenging task. In parallel computing, the Jacobi method is most commonly used due to its well adaptation to GPU. Unfortunately, the Jacobi method can only be used to solve a system of linear equations. In the following context, we develop a semi-implicit successive substitution method to solve Eq. (11).

4. Semi-implicit successive substitution method

In a substitution-type method, if we would like to find the roots for a nonlinear equation $f(x) = 0$, we usually start with initial guesses for all of the unknowns and then loop around the equation to obtain "better" approximations. To take successive iterations, we construct a function $g(x)$ satisfying

$$x = g(x) \iff f(x) = 0, \tag{15}$$

known as the fixed-point problem. Finding a function satisfying Eq. (15) is easy, e.g., we can just set $g(x) = f(x) - x$. However, the difficulty is how to guarantee the successive substitutions $x^{k+1} = g(x^k)$ will converge to the root with the chosen $g(x)$. Conditions for convergence of successive substitution method provides us a sufficient convergence criteria

$$|g'(x)| < 1. \tag{16}$$

From Figure 2, it can be noted that if the convergence criteria is strictly satisfied, x^k will finally converge to root as k increases. Otherwise, x^k may diverge. In the following context, we will talk about how to reformulate Eq. (11) into a formula that can satisfy the convergence criterion in Eq. (16). Motivated by the projective peridynamics [9], we separate the *force state* $\underline{\mathbf{T}}_i \langle \mathbf{x}_j - \mathbf{x}_i \rangle$ into two parts

$$\underline{\mathbf{T}}_i \langle \mathbf{x}_j - \mathbf{x}_i \rangle = \underline{\mathbf{T}}_i^{+} \langle \mathbf{x}_j - \mathbf{x}_i \rangle + \underline{\mathbf{T}}_i^{-} \langle \mathbf{x}_j - \mathbf{x}_i \rangle, \tag{17}$$

where $\underline{\mathbf{T}}_i^{+} \langle \mathbf{x}_j - \mathbf{x}_i \rangle$ will be further formulated as a function of $\mathbf{y}_j - \mathbf{y}_i$ while $\underline{\mathbf{T}}_i^{-} \langle \mathbf{x}_j - \mathbf{x}_i \rangle$ still remain as a function of $\mathbf{x}_j - \mathbf{x}_i$. The superscript $+$ indicates the coefficients of

$\mathbf{x}_j - \mathbf{x}_i$ are guaranteed to be positive while the superscript $-$ indicates the coefficients of $\mathbf{x}_j - \mathbf{x}_i$ are guaranteed to be negative. For isotropic materials, Teran et al. [40] pointed out that the first Piola-Kirchhoff stress tensor can be factorized into the following form

$$\mathbf{P}_i = \mathbf{U}_i \hat{\mathbf{P}}_i \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T, \tag{18}$$

where $\hat{\mathbf{F}}_i$ is the diagonal matrix obtained from the standard singular value decomposition of $\mathbf{F}_i = \mathbf{U}_i \hat{\mathbf{F}}_i \mathbf{V}_i^T$, $\hat{\mathbf{P}}_i$ is a function of $\hat{\mathbf{F}}_i$ which is also a diagonal matrix. By splitting $\mathbf{P}_i$ into two components, we have

$$\mathbf{P}_i = \mathbf{U}_i \hat{\mathbf{P}}_i^+ \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T + \mathbf{U}_i \hat{\mathbf{P}}_i^- \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T, \tag{19}$$

where $\hat{\mathbf{P}}_i^+$ represents the positive component of $\hat{\mathbf{P}}_i$ while $\hat{\mathbf{P}}_i^-$ the negative component. With the above separation, the positive part of the *force state* can be reformulated as

$$\begin{aligned} \underline{\mathbf{T}}_i^+ \langle \mathbf{x}_j - \mathbf{x}_i \rangle &= \omega_{ij} \mathbf{P}_i^+ {\mathbf{K}_i}^{-1} \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ &= \omega_{ij} \mathbf{U}_i \hat{\mathbf{P}}_i^+ \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T {\mathbf{K}_i}^{-1} \mathbf{V}_i \hat{\mathbf{F}}_i^{-1} \mathbf{U}_i^T \\ &\quad \cdot (\mathbf{y}_j - \mathbf{y}_i) . \end{aligned} \tag{20}$$

by applying an approximation $\mathbf{y}_j - \mathbf{y}_i \approx \mathbf{F}_i \cdot (\mathbf{x}_j - \mathbf{x}_i)$ during the derivation. In a similar way, the negative part of the *force state* can be reformulated as

$$\begin{aligned} \underline{\mathbf{T}}_i^- \langle \mathbf{x}_j - \mathbf{x}_i \rangle &= \omega_{ij} \mathbf{P}_i^- {\mathbf{K}_i}^{-1} \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ &= \omega_{ij} \mathbf{U}_i \hat{\mathbf{P}}_i^- \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T {\mathbf{K}_i}^{-1} \cdot (\mathbf{x}_j - \mathbf{x}_i) . \end{aligned} \tag{21}$$

To further simply the *force state*, we assume the shape tensor $\mathbf{K}_i$ in Eq. (20) and Eq. (21) is isotropic and its value can be calculated analytically according to the following formula

$$\mathbf{K}_i = \int_{\mathcal{H}_i} (\mathbf{x}_j - \mathbf{x}_i) (\mathbf{x}_j - \mathbf{x}_i)^T dV_j = \frac{16}{15} \pi {r_i}^5 \mathbf{I}, \tag{22}$$

where $\mathcal{H}_i$ represents a spherical neighborhood of radius r_i centered at $\mathbf{x}_i$ and $\mathbf{I}$ represents the identity matrix. By substituting the analytical solution of $\mathbf{K}_i$ into both Eq. (20) and Eq. (21), a simplified form of the *force state* can finally be defined as

$$\begin{aligned} \underline{\mathbf{T}}_i \langle \mathbf{x}_j - \mathbf{x}_i \rangle = \kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^+ \left(\hat{\mathbf{F}}_i\right) \hat{\mathbf{F}}_i^{-1} \mathbf{U}_i^T \cdot (\mathbf{y}_j - \mathbf{y}_i) + \\ \kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^- \left(\hat{\mathbf{F}}_i\right) \mathbf{V}_i^T \cdot (\mathbf{x}_j - \mathbf{x}_i) , \end{aligned} \tag{23}$$

where $\kappa_i = \frac{16}{15} \pi {r_i}^5$. Note when Eq. (23) is solved with our semi-implicit successive substitution method, we make a special treatment to handle the first term of RHS in an implicit way while the second term in an explicit way. Substituting Eq. (23) into Eq. (11), the semi-implicit governing equation for a nonlinear system can finally be defined as

$$\mathbf{y}_i^{k+1} = \left(m_i \mathbf{I} + \mathbf{A}_i^k\right)^{-1} \left(\sum_j \mathbf{A}_{ij}^k \mathbf{y}_j^k + \mathbf{s}_i^k + \mathbf{s}_i^t \right), \tag{24}$$

where

$$\begin{aligned} \mathbf{A}_{ij} &= h^2 V_i \left(\kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^+ \hat{\mathbf{F}}_i^{-1} \mathbf{U}_i^T + \kappa_j \mathbf{U}_j \hat{\mathbf{P}}_j^+ \hat{\mathbf{F}}_j^{-1} \mathbf{U}_j^T \right) V_j \\ \mathbf{s}_i^k &= h^2 V_i \sum\nolimits_j \left(\kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^- \mathbf{V}_i^T + \kappa_j \mathbf{U}_j \hat{\mathbf{P}}_j^- \mathbf{V}_j^T \right) V_j \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{s}_i^t &= m_i \left(\mathbf{y}_i^t + h \mathbf{v}_i^* \right) \\ \mathbf{A}_i &= \sum\nolimits_j \mathbf{A}_{ij} . \end{aligned} \tag{25}$$

Please note Eq. (24) is now in a form of $x^{k+1} = g(x^k)$, which can be easily solved with successive substitutions. The question is whether $\mathbf{y}_i^{k+1}$ will converge to a global solution as k increases? By checking the convergent criterion, we have

$$\left\| \left(m_i \mathbf{I} + \mathbf{A}_i^k\right)^{-1} \sum_j \mathbf{A}_{ij}^k \right\| < 1, \quad k = 0, 1, \cdots . \tag{26}$$

Therefore, if we do not update the coefficients in Eq. (24) at each iteration, $\mathbf{y}^{k+1}$ will absolutely converge to a root of Eq. (24). However, without simultaneously updating the coefficients in Eq. (24), the roots are not guaranteed to converge to the solution of the nonlinear system in Eq. (11).

4.1. Convergence and Performance

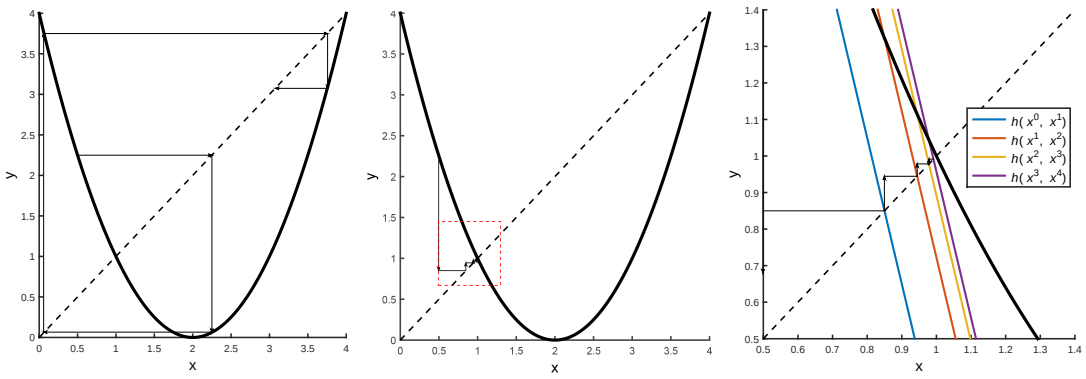
To demonstrate how our semi-implicit successive substitution method works, we first consider a simple one dimensional spring whose governing equation is written as $x_0 = (x_0 - x_1 - 1)^2$ in a position-based manner. Here, x_0 and x_1 represent the one-dimensional coordinates of two spring ends. By fixing one end, e.g., setting $x_1 = 1$, the governing equation for the other spring end is written as

$$x_0 = (x_0 - 2)^2 \equiv g(x_0) . \tag{27}$$

Starting with an initial guess of $x_0 = 0.5$, it can be noticed from Figure 3(a) that the standard successive substitution method fails to find a converged solution while Figure 3(b) shows our method is able to find the converged solution. In our semi-implicit successive substitution method, the definition of $g(x_0)$ at k-th iteration is first converted into a linearized semi-implicit equation

$$h\left(x_0^k, x_0^{k+1}\right) = -4x_0^{k+1} + \left(x_0^k\right)^2 + 4. \tag{28}$$

Therefore, solving $x_0^{k+1} = -4x_0^{k+1} + \left(x_0^k\right)^2 + 4$ is equivalent to find an intersection point between $y = x$ and $y = -4x + \left(x_0^k\right)^2 + 4$, as was demonstrated in Figure 3(c).



(a) (b) (c)

Figure 3. Convergence illustration on how to find the intersection point between $y = x$ and $y = (x-2)^2$. Starting with an initial guess of $x = 0.5$, (a) the standard successive substitution method fails to converge while (b) our semi-implicit successive substitution method succeeds in finding the solution. Refer to (c) for a snapshot of the convergence routine of our method.

Notice when the linearized function $h\left(x_0^k, x_0^{k+1}\right)$ is updated at each iteration, the sequence x_0^k is guaranteed to converge to one of the solution of Eq. (27).

Now let us consider a general dynamics problem and define its governing equations as the following multivariable functions

$$\mathbf{x} = \mathbf{g}\left(\mathbf{x}\right). \tag{29}$$

To take one iteration, $\mathbf{g}\left(\mathbf{x}\right)$ is first converted into a linearized semi-implicit equation $\mathbf{h}(\mathbf{x}^{k+1}, \mathbf{x}^k)$ at the beginning of each iteration. Then, the intersection point $\mathbf{x}^{k+1}$ between the two hyperplanes $\mathbf{y} = \mathbf{x}$ and $\mathbf{y} = h(\mathbf{x}, \mathbf{x}^k)$ is calculated by solving the linear system of equation $\mathbf{x} = h(\mathbf{x}, \mathbf{x}^k)$. As the iteration number increases, the sequence $\mathbf{x}^{k+1}$ is expected to converge to a global solution of the dynamics problem defined in Eq. (29). However, for an arbitrary function of $\mathbf{g}(\mathbf{x})$, the semi-implicit successive substitution method still suffers from the **overshooting problem**, as was demonstrated in Figure 4. Note as $\mathbf{x}^k$ approaches the global solution $\mathbf{x}^*$, the gradient magnitude of linearized semi-implicit equation $\mathbf{h}\left(\mathbf{x}_0^k, \mathbf{x}_0^{k+1}\right)$ can finally become larger than the gradient magnitude of the original function. As a result, further substitution iterations may cause the sequence $\mathbf{x}^k$ to oscillate around the global solution.

To address the overshooting problem, our solution is to adjust the step length as follows

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \alpha_i \left(\mathbf{x}_i^{k+1} - \mathbf{x}^k{}_i\right), \tag{30}$$

where $\alpha_i \in [0, 1]$ represents the step length for vertex i. In the standard gradient descent methods, a backtracking line search is typically applied to find the feasible step size. Starting from an initial guess for α, e.g., $\alpha_0 = 1$, the value of α will be halved until the Wolfe or Goldstein conditions are reached. The problem with this standard strategy is that

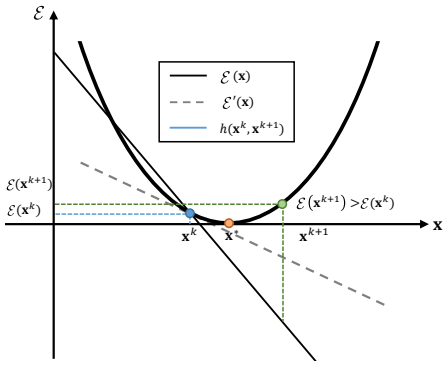


Figure 4. Demonstration of the overshooting problem. As the sequence $\mathbf{x}^k$ approaches the local minima of the energy function, the gradient magnitude of the linearized semi-implicit equation $\mathbf{h}(\mathbf{x}_0^k, \mathbf{x}_0^{k+1})$ can become larger than the gradient magnitude of the energy function, which could possibly cause the sequence $\mathbf{x}^k$ to oscillate around the global solution.

it can waste a lot of iterations during the backtracking process [43]. In our work, we propose a new step length search method by only taking account of the current deformation state. Given $\mathbf{x}^k$, we first reformulate the energy density function $\mathcal{E}\left(\mathbf{x}\right)$ into a first-order Taylor polynomial defined as

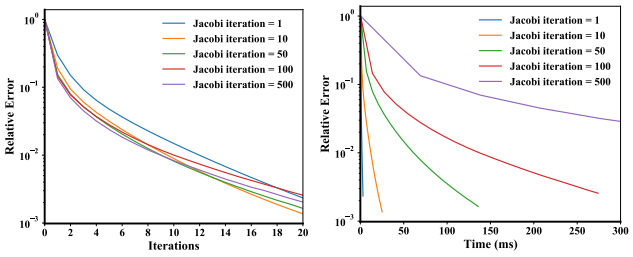
$$\mathcal{E}\left(\mathbf{x}\right) = \mathcal{E}\left(\mathbf{x}_i^k\right) + \alpha \frac{\partial \mathcal{E}_i}{\partial \mathbf{x}_i}\left(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\right). \tag{31}$$

Since the new energy density function is guaranteed to be non negative, an upper limit for α_i can naturally be defined as

$$\alpha_i = \min\left[-\frac{\mathcal{E}\left(\mathbf{x}_i^k\right)}{\frac{\partial \mathcal{E}_i}{\partial \mathbf{x}_i} \cdot \left(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\right)}, 1\right]. \tag{32}$$

Note α_i can be calculated independently for each vertex, which makes our method quite suitable for GPU implementation. In addition, our method requires no iterative strat-

egy to adjust the step length because the upper limit of α_i has already provides a sufficient condition to guarantee the convergence. In a practical implementation, the value of the denominator can be zero due to the rounding errors resulted from floating-point calculations. To avoid being divided by zero, we simply set $\alpha_i = 1$ if the denominator value is smaller than a predefined threshold, e.g., 10^{-6} for single precision operations.



(a) (b)

Figure 5. The convergence of our method with different Jacobi iteration numbers in solving Eq. (24). By using more Jacobi iterations, the semi-implicit successive substitution can accelerate the convergence rate. However, the total computational cost is increased.

The full procedure of our method is now quite similar to the Newton–Raphson method except that our method takes one additional operation to adjust the step length. In addition, our method imposes no restriction on the accuracy in solving the linear system of equations $\mathbf{x} = \mathbf{h}(\mathbf{x}, \mathbf{x}^k)$. Due to our step length adjustment strategy, our method is flexible enough to solve the linearized semi-implicit equation at arbitrary accuracy. Figure 5(a) demonstrates how the accuracy in solving the linearized semi-implicit equation can affect the global convergence rates. Generally speaking, the more accurate we solve the linearized semi-implicit equation, the faster convergence rate can be achieved. However, there exists a performance balance between the global iteration number and the inner iteration number. Figure 5(b) shows that increasing the iteration number in solving the linearized semi-implicit equation can significantly increase the total computational cost. Therefore, in our practical implementation, we usually take only one Jacobi iteration to solve $\mathbf{x} = h(\mathbf{x}, \mathbf{x}^k)$ and the new sequence is found to converge to the global solution well.

5. Hyperelastic modeling of membranes

In this section, we will give details on how to decompose the first Piola-Kirchhoff stress tensor for hyperplastic materials. According to the continuum theory, the strain-energy density function $\Psi(I_1, I_2, I_3)$ of an isotropic hyperelastic material is defined by the three invariants of the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$:

$$\begin{aligned} I_1 &= \mathrm{trace}(\mathbf{C}) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \mathbf{C} : \mathbf{C} = \lambda_1^4 + \lambda_2^4 + \lambda_3^4 \\ I_3 &= \det(\mathbf{C}) = \lambda_1^2\lambda_2^2\lambda_3^2 \,. \end{aligned} \tag{33}$$

Following [10, 46], the diagonal first Piola-Kirchhoff stress tensor can be derived as

$$\hat{\mathbf{P}} = \mathrm{diag}\left(\frac{\partial \Psi}{\partial \lambda_1}, \frac{\partial \Psi}{\partial \lambda_2}, \frac{\partial \Psi}{\partial \lambda_3}\right). \tag{34}$$

Taking the compressible neo-Hookean material for example, its strain energy density function is formulated as

$$\Psi = s_0 \left(I_1 - 3 - \ln I_3\right) + s_1 \left(\sqrt{I_3} - 1\right)^2, \tag{35}$$

we have

$$\frac{\partial \Psi}{\partial \lambda_m} = 2s_0\lambda_m + 2s_1\frac{I_3}{\lambda_m} - 2\left(\frac{s_0}{\lambda_m} + s_1\frac{{I_3}^{1/2}}{\lambda_m}\right). \tag{36}$$

Therefore, we succeed in decomposing $\hat{\mathbf{P}}$ into

$$\begin{aligned} \hat{\mathbf{P}}^+_{m,m} &= 2s_0\lambda_m + 2s_1\frac{I_3}{\lambda_m} \\ \hat{\mathbf{P}}^-_{m,m} &= -2\left(\frac{s_0}{\lambda_m} + s_1\frac{{I_3}^{1/2}}{\lambda_m}\right), \end{aligned} \tag{37}$$

which absolutely obeys the decomposition principle as well.

However, if we try to model hyperelasticity for shell-like objects with standard hyperelastic energy density functions, several difficulties can be encountered. For example, since normal principle stretch used to calculate the third invariant I_3 can be relatively small for codimensional materials, the force arisen from the second term in Eq. (34) can be quite sensitive to the deformation. Besides, the Poisson effect makes it hard to control the stretching and bending for membranes independently. Motivated by the common way to decouple stretching and bending strain for membranes [38, 8, 2, 12], we propose to use two specific energy models that can be well adapted into our framework and control stretching and bending separately.

5.1. Stretching

To better modeling stretching for membranes, we propose to use the following user-defined hyperelasticity model using principle of strain λ_i:

$$\Psi = s_0 \left(\mathcal{A}_3(\lambda_1) + \mathcal{A}_3(\lambda_2) + \mathcal{A}_3(\lambda_3)\right), \tag{38}$$

where $\mathcal{A}_3$ is a special case of $\mathcal{A}_n$ proposed by [47]

$$\mathcal{A}_n = \frac{1}{n}\left(\frac{s^{n+1} - 1}{n+1} + \frac{s^{-n+1} - 1}{n-1}\right). \tag{39}$$

The corresponding $\hat{\mathbf{P}}$ can be decomposed into two parts as followed:

$$\begin{aligned}\hat{\mathbf{P}}^{+}_{m,m} &= s_0 \frac{{\lambda_m}^3}{3} \\ \hat{\mathbf{P}}^{-}_{m,m} &= -s_0 \frac{1}{3{\lambda_m}^3}\,.\end{aligned} \tag{40}$$

5.2. Bending

To allow modeling membrane bending independently, the bending force should disappear for configurations under the rest, pure rotational and stretching states (i.e. gradient deformation tensor is diagonal). In order to satisfy the above constraints, we extend the meshless bending model in Wu et al.[44] and make a further derivation to make the final form of the bending force compatible with our semi-implicit solver. The bending energy density function on each vertex is defined as

$$E_i^b = \frac{k_b}{2} \sum_j \left\| \mathbf{G}_i(\mathbf{x}_j - \mathbf{x}_i) \right\|_2^2, \tag{41}$$

where k_b is constant parameter for bending control, $\mathbf{G} = \mathbf{F}^T\mathbf{F}^{-1} - \mathbf{I}$ can be viewed as a bending measurement. It is easy to verify that this measurement can fit into our method as well. By taking the approximation of $\mathbf{y}_j - \mathbf{y}_i \approx \mathbf{F}_i(\mathbf{x}_j - \mathbf{x}_i)$, the bending force derived from $-\nabla_{\mathbf{y}} E_i^b$ can be written as

$$\underline{\mathbf{T}}_i^b \langle \mathbf{x}_j - \mathbf{x}_i \rangle = k_b \mathbf{F}_i^{-T}\mathbf{F}_i^{-1}(\mathbf{y}_j - \mathbf{y}_i) - k_b \mathbf{F}_i^{-1}(\mathbf{x}_j - \mathbf{x}_i). \tag{42}$$

Notice Eq. (42) resembles the force state formulation defined in Eq. (23). Therefore, both the stretching and bending energy can be efficiently solved with our semi-implicit successive substitution method.

To sum up, a global step to solve the membrane can be formulated by the Eq. (24) with $\mathbf{A}_{ij}$ and $\mathbf{s}_i^k$ defined as follows

$$\begin{aligned}\mathbf{A}_{ij} = h^2 V_i \Big(&\kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^{+} \hat{\mathbf{F}}_i^{-1} \mathbf{U}_i^T + \kappa_j \mathbf{U}_j \hat{\mathbf{P}}_j^{+} \hat{\mathbf{F}}_j^{-1} \mathbf{U}_j^T \\ &+ k_b (\mathbf{F}_i \mathbf{F}_i^T)^{-1} + k_b (\mathbf{F}_j \mathbf{F}_j^T)^{-1} \Big) V_j,\end{aligned} \tag{43}$$

$$\begin{aligned}\mathbf{s}_i^k = h^2 V_i \sum_j \Big(&\kappa_i \mathbf{U}_i \hat{\mathbf{P}}_i^{-} \mathbf{V}_i^T + \kappa_j \mathbf{U}_j \hat{\mathbf{P}}_i^{-} \mathbf{V}_j^T \\ &- k_b \mathbf{F}_i^{-1} - k_b \mathbf{F}_j^{-1} \Big) V_j \cdot (\mathbf{x}_j - \mathbf{x}_i).\end{aligned} \tag{44}$$

Figure 7 and Figure 6 demonstrate how parameter s_0 and k_b influence the deformation behavior of a hyperelastic cloth. Generally, a larger value of s_0 generate more stiff stretching behaviors, while a larger value of k_b brings more resistance to folding and buckling. Notice, the two parameters can be independently adjusted to control different deformation behaviors.

6. Contact handling

Notice the global step solve cannot guarantee vertex positions remain interpenetration-free. Therefore, a local step should be taken in order to avoid interpenetration. Given the vertex positions $\mathbf{y}^k$ after one global step, two objectives should be fulfilled after taking the local step solve. Firstly, vertex positions $\mathbf{y}$ should remain as close as to $\mathbf{y}^k$. Secondly, the constraint $B(\mathbf{z}) = 0$ is fulfilled for membrane modeling with a finite thickness of ξ. According to Eq. (5), the general formulation for the contact problem in the local step can be written as

$$\arg\min_{\mathbf{z}} \sum_i \frac{\omega_i}{2} \left\| \mathbf{B}_i \mathbf{z}_i - \mathbf{A}_i \mathbf{S}_i \mathbf{y}_i^k \right\|^2, \;\; s.t. \;\; B(\mathbf{z}) = 0. \tag{45}$$

For simplicity, we choose $\mathbf{A}_i = \mathbf{B}_i = \mathbf{S}_i = I$ and reformulate Eq. (45) as follows

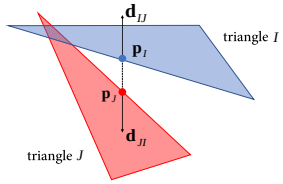
$$\arg\min_{\mathbf{z}} \sum_i \frac{1}{2} \left\| \mathbf{z}_i - \mathbf{y}_i^k \right\|^2, \;\; s.t. \;\; B(\mathbf{z}) = 0. \tag{46}$$

According to the numerical optimization theory, the above constrained optimization problem actually can be reformulated into the following unconstrained optimization problem:

$$\mathcal{B}(\mathbf{z}) = \min_{\mathbf{z}} \left(\sum_i \frac{1}{2} \left\| \mathbf{z}_i - \mathbf{y}_i^k \right\|^2 + \mu \sum_k B_k(\mathbf{z}) \right), \tag{47}$$

where μ is a constant weight. Note the first term $\frac{1}{2} \left\| \mathbf{z}_i - \mathbf{y}_i^k \right\|_2^2$ acts like a momentum potential to prevent vertices from getting too far away. The second term is used to penalize the collision. In the follow context, we will present a gradient descent method to solve the unconstrained optimization problem, as demonstrated in Algorithm 2. Note all steps in our local step solve is highly parallelizable and can be easily implemented on modern GPUs.

6.1. Direction of gradient descent

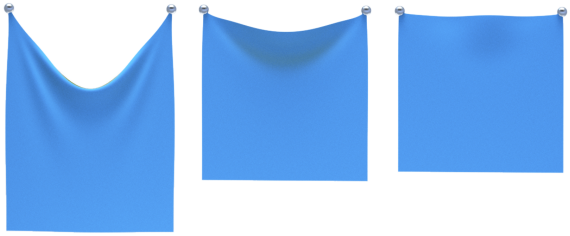


In this section, we first discuss how to define a well-chosen descent direction that can decrease the total potential energy gradually.

If we consider the collision detection between a pair of triangles, the CCD test between the two triangles actually reduces to performing 6 vertex-triangle queries and 9 edge-edge queries. Without loss of generality, let us consider the edge-edge case in the right figure to demonstrate

(a) 0× (b) 0.1× (c) 1× (d) 10×

Figure 6. Bending stiffness test. A cloth drops onto a table with some books under default gravity. The stretching stiffness s_0 is set to a constant 8000, while the bending stiffness k_b is set to $0\times, 0.1\times, 1\times$ and 10×240 from left to right.



(a) 0.1× (b) 1× (c) 10×

Figure 7. Stretching stiffness test. The cloth is dangling down under 20 times the default gravity. From left to right, the stretching stiffness s_0 is set to $0.1\times, 1\times$ and 10×500, while the bending stiffness is set to 0 for all simulations.

how to calculate the gradient descent direction. By denoting the two triangle as I and J, the distance between triangle I and J can be defined as

$$d_{IJ} = \arg\min_{\alpha_I, \alpha_J} \left(||\mathbf{p}_I(\alpha_I) - \mathbf{p}_J(\alpha_J)||_2 \right), \tag{48}$$

where $\mathbf{p}_I$ and $\mathbf{p}_J$ represent a point locating inside triangle I and J, respectively, α_I and α_J represent the barycentric coordinates. If we insert above equation into Eq. (47), the gradient descent direction for vertex i can then be calculated by taking derivative of $\mathcal{B}(\mathbf{y})$ with respect to $\mathbf{z}_i$

$$\begin{aligned} \mathbf{g}_i =& -\nabla_{\mathbf{z}_i}\mathcal{B} \\ =& \mu \sum_I \alpha_I^i \left[\frac{(d_{IJ} - \hat{d})^2}{d_{IJ}} + 2(d_{IJ} - \hat{d}) \log\left(\frac{d_{IJ}}{\hat{d}}\right) \right] \\ & \text{norm}(\mathbf{d}_{IJ}) + \mathbf{y}_i^k - \mathbf{z}_i \,, \end{aligned} \tag{49}$$

where α_I^i represents the barycentric coordinate for vertex i in triangle I.

6.2. Step length adjustment

Unlike solving traditional unconstrained optimization problem, the feasible set of $\mathbf{y}$ in our problem is further constrained by the thickness of membrane. More precisely, we require the unsigned distance between any pair of primitives to satisfy the following strict inequal condition:

$$\hat{d}_{IJ} = d_{IJ} - \xi > 0. \tag{50}$$

To fulfill above condition, one practice is to take the Additive CCD algorithm [16] to find a lower bound of time-of-impact t_{IJ} for each pair of primitives and then clamp the vertex position according to Line 4∼18 in Algorithm (2). However, directly clamping the vertex position by the time-of-impact may introduce sticking artifacts, just as pointed out by Wang et al.[42].

Therefore, we propose to rescale the step length according the time-of-impact. More precisely, the time-of-impact will be used to clamp vertex position only if it is greater than a predefined threshold. Otherwise, we use a C^1 piecewise continuous function to rescale the proximal distance d_{IJ} as follows

$$f(d) = \begin{cases} (1+\varepsilon)\xi, & d \le \xi \\ \dfrac{1}{4\varepsilon\xi}(d-\xi)^2 + (1+\varepsilon)\xi, & \xi < d \le (1+2\varepsilon)\xi \\ d, & (1+2\varepsilon)\xi < d \end{cases}, \tag{51}$$

where ε is a user-defined mininal separation multiplier (which is typically set to $1e^{-1}$ or $1e^{-2}$ in our current implementation). If we replace d_{IJ} with $f(d_{IJ})$, it can be verified that Eq. (50) stands forever given a positive mininal separation multiplier ε, i.e., $\hat{d}_{IJ} > \varepsilon\xi$. Besides, it can further be verified that the first derivative of the barrier function $B'(d)$ is bounded, as shown in Figure 8.

After we insert Eq. (50) into Eq. (49), the question becomes how to calculate a suitable step length λ that would gradually minimize the object function as follows

$$\mathcal{B}(\mathbf{z}^m + \lambda\mathbf{g}) < \mathcal{B}(\mathbf{z}^m). \tag{52}$$

A simple yet effective approach is to use a *backtracking line search* approach to reduce an initial guess until the Wolfe condition gets satisfied. However, this could waste a bunch of iterations during the backtracking process. In fact, the boundedness of $B'(d)$ allows us to guess an upper bound for the step length.

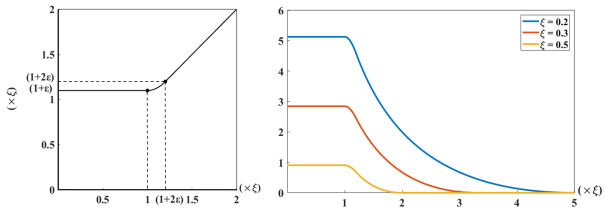
Algorithm 2: Contact Handling in the Local Step

Project($\mathbf{y}^k, \mathbf{z}^{k-1}, \xi, h, \varepsilon, \hat{d}$)
$\mathbf{z}^{m=0} \leftarrow \mathbf{y}^k$
while $\max_i \left(\left\|\mathbf{z}_i^m - \mathbf{z}_i^{m-1}\right\|_2\right) > eps$ *and* $m \leq max\_ite$ **do**
  foreach $triangle_I$ **do**
    foreach (tri_I, tri_J) *is activated,* **do**
      $t_{IJ} =$ AdditiveCCD($\text{tri}_J, \text{tri}_I, h, \mathbf{z}^m, \mathbf{z}^{k-1}$) //set $s = 0.1, t_c = 0.95$ for original ACCD algorithm
    end
    if $t_{IJ} < 1$ **then**
      Insert $(\text{tri}_I, \text{tri}_J)$ into $ContactList$
    end
    $t_I = \min_J(t_{IJ})$
  end
  foreach *vertex* i **do**
    $t_i = \min_{i \in \text{tri}_J}(t_J)$
    if $t_i < 1$ *and* $t_i >$ *threshold* **then**
      $\mathbf{z}_i^m \leftarrow (\mathbf{z}_i^m - \mathbf{z}_i^{k-1})t_i + \mathbf{z}_i^{k-1}$
    end
  end
  foreach $(tri_I, tri_J) \in ContactList$ **do**
    Calculate $\mathbf{p}_I, \mathbf{p}_J, \mathbf{d}_{IJ}$ (Section 6.1)
    Calculate α_I^i for $i \in \text{tri}_I$
    Calculate α_J^i for $i \in \text{tri}_J$
  end
  foreach *vertex* i **do**
    Calculate $\hat{d}_{IJ} = f(d_{IJ}) - \xi, \forall \text{tri}_I, i \in \text{tri}_I$ (Section 6.2)
    Calculate $\mathbf{g}_i$ (Section 6.1)
  end
  Calculate step length λ (Section 6.2)
  foreach *vertex* i **do**
    $\mathbf{z}_i^{m+1} \leftarrow \mathbf{z}_i^m + \lambda\, \mathbf{g}_i$
  end
end
return $\mathbf{z}^k \leftarrow \mathbf{z}^m$

By requiring the maximum position change at each iteration is no greater than $\varepsilon\xi$, i.e., $\max_i \|\lambda \mathbf{g}_i\| \leq \varepsilon\xi$, an upper bound for λ can easily be derived as

$$\lambda \leq \frac{\varepsilon\xi}{\mu\alpha_0 \lceil B'(d) \rceil + \hat{d}}, \tag{53}$$

where $\lceil B'(d) \rceil$ represents the upper bound of $B'(d)$, α_0 is an estimated upper bound of $\sum_I \alpha_I^i$, $\hat{d}$ is an upper bound of $\left\|\mathbf{y}_i^k - \mathbf{z}_i\right\|$.



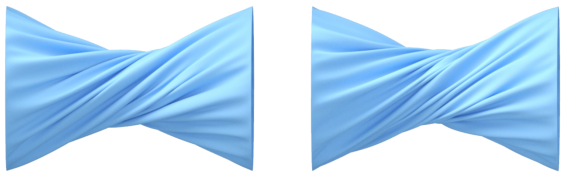
(a) $f(d)$ (b) $B'(d)$

Figure 8. (a) The diagram of refactor function in Eq. (51). In this case, we set $\varepsilon = 0.1$. (b) The corresponding barrier function derivative $B'(d)$ after adopting distance refact with $\xi = 0.2, 0.3$ and 0.5: $\hat{d}$ is set to a normalized number 1.0, denoted 1 times of primitive unit.

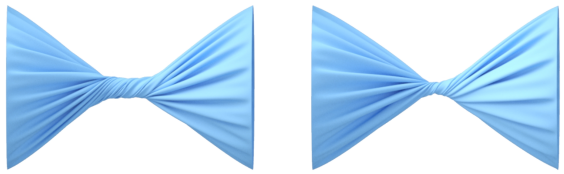
7. Results and discussion

All experiments are implemented in C++ and CUDA and are executed on 11th Gen Intel(R) Core(TM) i9-11900K CPU @3.5GHz ×8 with 32GB memory and NVIDIA RTX A4000 GPU. Table 1 shows all statistics for the experiments.

7.1. Thickness modelling



(a) Phase difference at 1.20π rad.



(b) Phase difference at 2.39π rad.

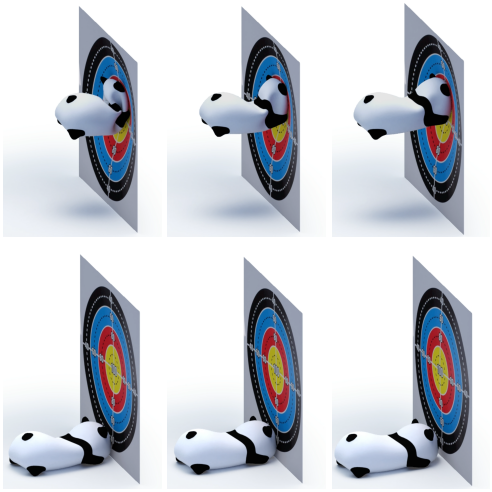
Figure 9. Elastic cylinders rotate.

In this set of experiments, an elastic cylinder is being twisted with an angular velocity $\omega = 1$ rad/s. The two ends of the cylinder are twisted in opposite directions. Figure 9 demonstrates the different central cylinder as we choose different values for the cloth thickness ξ.

7.2. Deformation, contact and separation.

In this example, we provide a set of experiments to test deformation, contact and separation for two objects. A panda-shape elastic shell is ejected to hit an elastic cloth-like target with an initial velocity $\mathbf{v} = 15$ m/s and density

$\rho = 1000$ kg/m^3. Both the panda-shape shell and cloth target are set to the same material. As the panda-shape shell hits the target, contact forces between two objects helps decelerate the speed of the shell. Then, the two objects separate and no unnatural locking artifacts can be observed. After changing the material stiffness s_0, distinctive deformation behaviors can also be observed. Notice as the material stiffness s_0 is decreased, the panda appears to be softer.



(a) $s_0 = 2000$ (b) $s_0 = 8000$ (c) $s_0 = 40000$

Figure 10. Shooting panda. The stretching stiffness s_0 is set to 2000, 8000 and 40000 from left to right while the bending stiffness is set to $0.005 \times s_0$ for all simulations.

7.3. Interactive cloth simulation

Figure 11 demonstrates an example to show interactive cloth simulation with mouse interactions. In this configuration, contact with the avatar is handled with SDF-based constraints while inner contact among the cloth is handled with our local step solver. The supplementary video shows that this complex example can be run at an interactive speed even when a large scale of primitives and contact get involved.

7.4. Comparison to descent methods [43]

To be fair, we set up an example similar to Figure 6 and compare our semi-implicit successive substitution method to the gradient descent method [43]. The compressible neo-Hookean model [28] is chosen to define its strain energy density function and the two Lamé parameters are set to $s_0 = 4.8e^7$ and $s_1 = 1.2e^7$, respectively. The timestep is set to $1ms$. The relative error at k-th iteration is defined to be $\eta = \frac{E(\mathbf{y}^k) - E(\mathbf{y}^*)}{E(\mathbf{y}^0) - E(\mathbf{y}^*)}$, where E is the total strain energy, $\mathbf{y}^0$ is the initial guess, $\mathbf{y}^k$ is the solution at k-th iteration and $\mathbf{y}^*$ is the exact solution (in our practical implementation, $\mathbf{y}^*$ is chosen to be the solution after 500 iterations are taken). Figure 12 shows statistics on the convergence and time cost at the 20-th frame for both methods. First of all, it can be noted that our method achieves a fast convergence rate at the first dozens of iterations. However, the convergence rate gradually slows down as the iteration number is further increased, just similar to a standard Jacobi iterative method. Besides, it can be noted from Figure 12(bottom) that our method takes lower computational cost in taking one iteration due to the efficient step length adjustment method. Finally, both methods can be accelerated by taking the Chebyshev method.

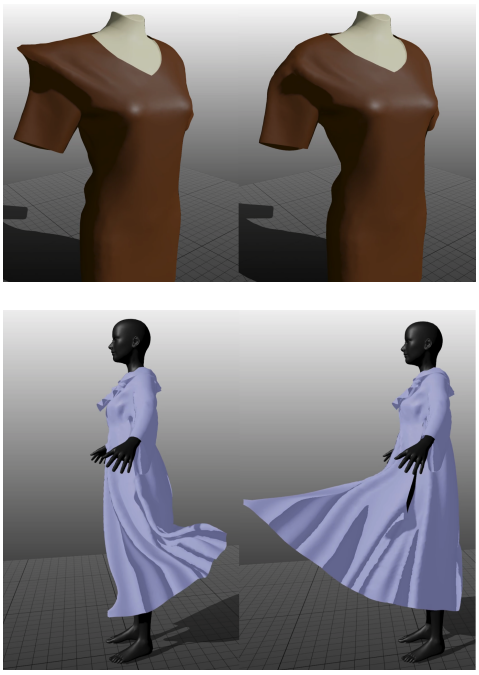


Figure 11. Interactive cloth simulation with contact. For the short skirt interaction, experiment running in averge 22 frames per seconds, while for the long dress interaction, experiment running in averge 8 frames per seconds because of more self-collision handling involved.

8. Conclusion and limitations

We present a stable and efficient semi-implicit successive substitution method for simulating hyperelastic membranes with contact based on peridynamics. Inspired by the fix-point iterative method, we separate inner elastic force into an implicit positive part and an explicit negative part to guarantee the convergence of each iteration. Overshooting problem is avoided by adopting self-adapted step length on position marching through gradient direction during iteration. Furthermore, self-collision contact will be projected as constraint and solved with gradient descend method.

For membranes modeling, we follow classical Kirchhoff–Love plate theory and adopt a vitual bond with direction of vertex normal for each particles to revise sigular shape matrix and deformation gradient matrix. We also derive two specific energy models that can be well adapted

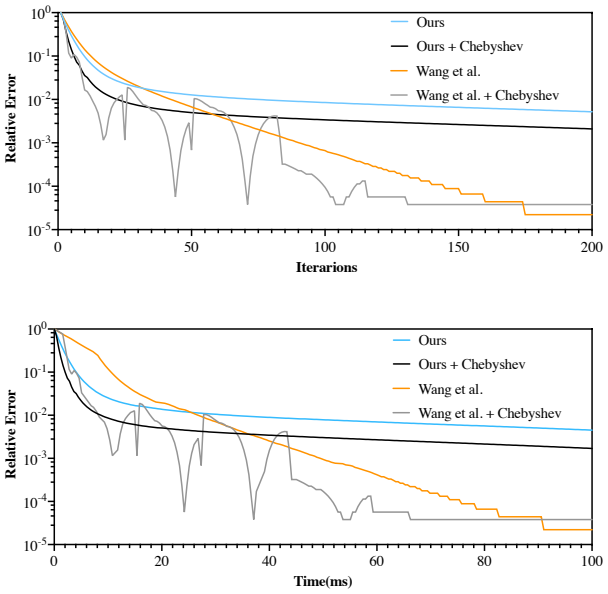


Figure 12. Comparison to Wang and Yang [43] with/without Chebyshev acceleration. Relative error over iteration step and running time is shown correspondingly.

into the framework and allow to model stretching and bending property independently for membranes material.

Our method also has several limitations. First, the convergence rate slows down in simulating a muti-layer cloth, which could possibly be caused by inexact gradient descend direction when both sides of the mid-layer generate active contact pairs. Besides, our method cannot implement strict controllable strain limits for elasticity when simulating shell-like materials. For our further work, we would first try to test with other collision detection algorithms in the local step. Then, since the proposed semi-implicit successive substitution method is general, we would like to apply it to solve constraints in other applications, e.g. rigid body dynamics. Finally, we would like to study on how to simulate other material models that involve both nonlinearity and contact.

Acknowledgement

We would like to thank anonymous reviewers for their valuable comments. The project was supported by the National Key R&D Program of China (No.2021YFB1715800), the National Natural Science Foundation of China (No.61872345), Youth Innovation Promotion Association, CAS (No.2019109).

References

[1] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966. 1

[2] D. Baraff and A. P. Witkin. Large steps in cloth simulation. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 43–54, July 1998. 8

[3] J. Bender, M. Müller, and M. Macklin. Position-based simulation methods in computer graphics. *EUROGRAPHICS Tutorial Notes*, 2015. 1, 3

[4] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph. (SIGGRAPH)*, 33(4):154:1–154:11, 2014. 1, 3

[5] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, jul 2002. 3

[6] W. Gerstle, N. Sau, and S. Silling. Peridynamic modeling of concrete structures. *Nuclear engineering and design*, 237(12-13):1250–1258, 2007. 2

[7] S. F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical report, Technical Report, Mitsubishi Electric Research Laboratories, 1997. 2

[8] E. Grinspun, A. Hirani, M. Desbrun, and P. Schröder. Discrete shells. 07 2003. 8

[9] X. He, H. Wang, and E. Wu. Projective peridynamics for modeling versatile elastoplastic materials. *IEEE transactions on visualization and computer graphics*, 2017. 1, 5

[10] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of SCA*, pages 131–140, 2004. 2, 3, 8

[11] A. Javili, R. Morasata, E. Oterkus, and S. Oterkus. Peridynamics review. *Mathematics and Mechanics of Solids*, 24(11):3714–3739, 2019. 2

[12] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans. Graph. (SIGGRAPH)*, 36(4), 2017. 8

[13] L. Lan, G. Ma, Y. Yang, C. Zheng, M. Li, and C. Jiang. Penetration-free projective dynamics on the gpu. *ACM Trans. Graph.*, 41(4), jul 2022. 3

[14] R. B. Lehoucq and S. A. Silling. Force flux and the peridynamic stress tensor. *Journal of the Mechanics and Physics of Solids*, 56(4):1566–1577, 2008. 2

[15] M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman. Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.*, 39(4), jul 2020. 2, 3

[16] M. Li, D. Kaufman, and C. Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics*, 40, 08 2021. 3, 10

[17] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. Fast simulation of mass-spring systems. *ACM Trans. Graph. (SIGGRAPH Asia)*, 32(6):214:1–214:7, Nov. 2013. 1

[18] T. Liu, S. Bouaziz, and L. Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph.*, 36(3), May 2017. 1, 3

[19] M. Macklin and M. Muller. A constraint-based formulation of stable neo-hookean materials. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and*

Name	#Vertex, #Face	Max, Min mesh size	Position	h	s_0	k_b ($\times s_0$)	ξ (mm)	$\hat{d}$ ($\times \xi$)	g ($\times$default)
Cloths over Ball	(single layer) 40.0K, 79.2K	0.0121, 0.0085	Figure 1 left	1e-3	12000	0.1	1.5	4	1.0
			Figure 1 middle	1e-3	12000	0.1	1.5	4	1.0
			Figure 1 right	1e-3	12000	0.1	1.5	4	1.0
Cloth Drop	14.4K, 28.3K	0.0118, 0.0084	Figure 7 left	1e-4	50	0	2.4	4	20.0
			Figure 7 middle	1e-4	500	0	2.4	4	20.0
			Figure 7 right	1e-4	5000	0	2.4	4	20.0
Antependium Drop on Table	40.0K, 79.2K	0.0071, 0.0050	Figure 6 upper left	1e-3	8000	0	1.4	4	1.0
			Figure 6 upper right	1e-3	8000	0.003	1.4	4	1.0
			Figure 6 bottom left	1e-3	8000	0.03	1.4	4	1.0
			Figure 6 bottom right	1e-3	8000	0.3	1.4	4	1.0
Twisting Cylinder	0.32M, 0.64M	0.0108, 0.0075	Figure 9 left	1e-3	1200	0.1	3.2	4	0
			Figure 9 right	1e-3	1200	0.1	1.6	4	0
Shooting Panda	27.8K, 54.9K	0.0407, 0.0084	Figure 10 left	5e-4	2000	0.005	6.1	4	1.0
			Figure 10 middle	5e-4	8000	0.005	6.1	4	1.0
			Figure 10 right	5e-4	40000	0.005	6.1	4	1.0
Short Skirt	14.3K, 28.3K	0.0393, 0.0076	Figure 11 top	1e-3	12000	0.01	3.9	4	1.0
Long Dress	32.8K, 64.8K	0.0242, 0.0003	Figure 11 bottom	2e-3	12000	0.01	2.4	4	0.1

Table 1. Statistics and simulation parameters of all examples.

Games, MIG '21, New York, NY, USA, 2021. Association for Computing Machinery. 1

[20] E. Madenci and E. Oterkus. *Peridynamic Theory and Its Applications*, volume 10.1007/978-1-4614-8465-3. 2014. 2

[21] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. (SIGGRAPH)*, 30(4):37:1–37:12, July 2011. 2

[22] P. Mott and C. Roland. Limits to poisson's ratio in isotropic materials—general result for arbitrary deformation. *Physica Scripta*, 87(5):055404, 2013. 2

[23] M. Müller, N. Chentanez, T. Kim, and M. Macklin. Strain based dynamics. In *Proceedings of SCA*, pages 21–23, 2014. 1

[24] M. Müller and M. Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, GI '04, page 239–246, Waterloo, CAN, 2004. Canadian Human-Computer Communications Society. 2

[25] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, Apr. 2007. 1, 3

[26] R. Narain, M. Overby, and G. E. Brown. Admm $\supseteq$ projective dynamics: fast simulation of general constitutive models. In *Proceedings of SCA*, pages 21–28, 2016. 3

[27] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006. 2

[28] R. Ogden. *Non-Linear Elastic Deformations*. Dover Civil and Mechanical Engineering. Dover Publications, 2013. 12

[29] M. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. Implicit contact handling for deformable objects. *Comput. Graph. Forum*, 28:559–568, 04 2009. 3

[30] Y. Peng, B. Deng, J. Zhang, F. Geng, W. Qin, and L. Liu. Anderson acceleration for geometry optimization and physics simulation. *ACM Trans. Graph.*, 37(4):1–14, 2018. 3

[31] S.-D. Poisson. *Mémoire sur l'équilibre et le mouvement des corps élastiques*. F. Didot, 1828. 2

[32] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation*, 1997. 3

[33] J. N. Reddy. *Theory and analysis of elastic plates and shells*. CRC press, 2006. 5

[34] S. A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175–209, 2000. 2

[35] S. A. Silling and E. Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17-18):1526–1535, 2005. 2

[36] S. A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184, 2007. 2

[37] S. A. Silling and R. B. Lehoucq. Convergence of peridynamics to classical elasticity theory. *Journal of Elasticity*, 93(1):13–37, 2008. 2, 4

[38] R. Tamstorf and E. Grinspun. Discrete bending forces and their jacobians. *Graphical Models*, 75(6):362–370, 2013. 8

[39] M. Tang, T. Wang, Z. Liu, R. Tong, and D. Manocha. I-Cloth: Incremental collision handling for GPU-based interactive cloth simulation. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)*, 37(6):204:1–10, November 2018. 3

[40] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, page 181–190, New York, NY, USA, 2005. Association for Computing Machinery. 2, 6

[41] M. Tupek, J. Rimoli, and R. Radovitzky. An approach for incorporating classical continuum damage models in state-

based peridynamics. *Computer methods in applied mechanics and engineering*, 263:20–26, 2013. 2

[42] B. Wang, Z. Ferguson, T. Schneider, X. Jiang, M. Attene, and D. Panozzo. A large scale benchmark and an inclusion-based algorithm for continuous collision detection. *ACM Transactions on Graphics*, 40(5), Oct. 2021. 10

[43] H. Wang and Y. Yang. Descent methods for elastic body simulation on the gpu. *ACM Transactions on Graphics (TOG)*, 35(6):212, 2016. 2, 3, 7, 12, 13

[44] L. Wu, B. Wu, Y. Yang, and H. Wang. A safe and fast repulsion method for gpu-based cloth self collisions. *ACM Trans. Graph.*, 40(1), dec 2020. 3, 9

[45] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Computer Graphics Forum*, volume 20, pages 349–358. Wiley Online Library, 2001. 2

[46] H. Xu, F. Sin, Y. Zhu, and J. Barbič. Nonlinear material design using principal stretches. *ACM Trans. Graph.*, 34(4), July 2015. 2, 8

[47] L. Xu, X. He, W. Chen, S. Li, and G. Wang. Reformulating hyperelastic materials with peridynamic modeling. *Computer Graphics Forum*, 37(7):121–130, 2018. 8