DiffClothAI: **Differentiable Cloth Simulation** with **I**ntersection-free Frictional Contact and Differentiable Two-Way Coupling with **A**rticulated Rigid Bodies

Xinyuan Yu[1*], Siheng Zhao[1*], Siyuan Luo[2], Gang Yang[1] and Lin Shao[1]

***Abstract*— Differentiable Simulations have recently proven useful for various robotic manipulation tasks, including cloth manipulation. In robotic cloth simulation, it is crucial to maintain intersection-free properties. We present *DiffClothAI*, a differentiable cloth simulation with intersection-free friction contact and two-way coupling with articulated rigid bodies. *DiffClothAI* integrates the *Project Dynamics* and *Incremental Potential Contact* coherently and proposes an effective method to derive gradients in the Cloth Simulation. It also establishes the differentiable coupling mechanism between articulated rigid bodies and cloth. We conduct a comprehensive evaluation of *DiffClothAI*'s effectiveness and accuracy and perform a variety of experiments in downstream robotic manipulation tasks. Supplemental materials and videos are available on our project webpage at https://sites.google.com/view/diffsimcloth.**

I. Introduction

Physics-based simulation methods for cloth have demonstrated remarkable success in various domains, including computer graphics and robotics. Recently, a number of differentiable physics simulations, including rigid bodies [1], [2], soft bodies [3], [4], [5], [6], [7], cloth [8], [9], [10], articulated bodies [11], [12], [13], and fluids [14], [15], [16], [17], have been proposed to solve system identification, control, and learning problems. However, most existing differentiable cloth simulations result in penetration issues, especially in contact-rich manipulation tasks. Small violations of contact constraints lead to penetration between objects and self-penetration, severely compromising simulation accuracy and stability. It is essential to address this challenge for contact-rich cloth manipulation tasks. Moreover, manipulating solid objects with robotic arms is a common scenario, but extending this capability to other objects like elastic materials or fabrics is challenging due to the different physical models and motion solvers required between elastics and solids.

Incremental Potential Contact (*IPC*) [18] introduced smoothed, local barrier functions to resolve contact-response to maintain the intersection-free status between 3D meshes. Huang *et al.* [19] proposed a differentiable solver for time-dependent deformation simulation by using finite element discretization coupled with the *IPC* contact model to solve PDE and ODE problems. They adopt *IPC* to ensure minimal separation between objects and implement based on the open-source PolyFEM library [20]. For coupling between rigid bodies and cloth, one approach is to perform a rigid approximation of cloth at the contact point [21], and this method lacks the coupling from cloth to the rigid body. Another way is to represent the cloth and rigid bodies system in a unified particle system [22].

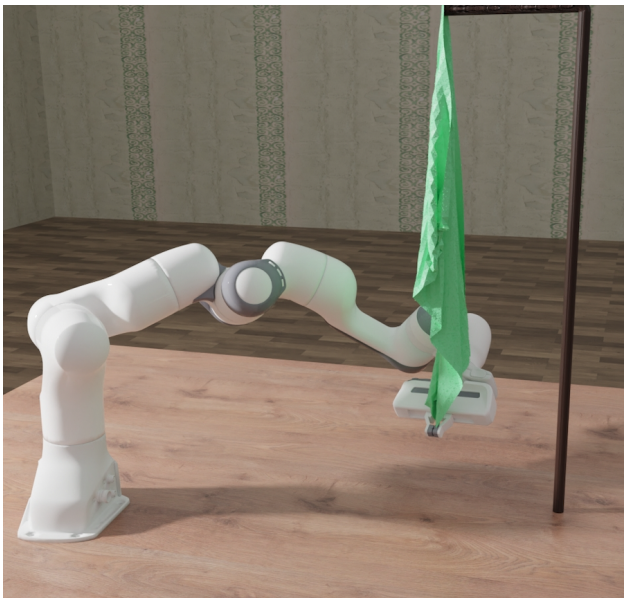

Fig. 1. A Franka robot is grasping a towel.

Our differentiable simulator for cloth and articulated rigid bodies provides intersection-free friction contact and differentiable coupling. We remove the coarse collision detection and response by vertex-vertex checking and processing, and we construct the differentiable IPC method with friction. This method is robust and maintains the intersection-free characteristic in our simulator. We adopt the adjacent sphere approximation of the cloth mesh vertices to deal with the coupling problem, and the contact force is transmitted to the cloth and rigid bodies at the contact position. We make the whole coupling process differentiable. As a result, we implement a robust and efficient intersection-free cloth and articulated rigid bodies bidirectional coupling differentiable simulator.

We provide an open-source implementation of our differ-

*The authors contributed equally.

[1] Xinyuan Yu, Siheng Zhao, Gang Yang and Lin Shao are with the Department of Computer Science, National University of Singapore, Singapore. yxy1106@gmail.com, zhaosiheng@u.nus.edu, yg.matinal@gmail.com and linshao@nus.edu.sg

[2] Siyuan Luo is with the Department of Computer Science, Xi'an Jiaotong University, China. luosiyuan2002@gmail.com

This work is in part supported by a startup grant from National University of Singapore.

entiable physics simulator, which we call *DiffClothAI*. Code and documentation will be released at our project website.

In summary, we make the following contributions:

- We present a differentiable formulation for cloth simulation. The method is efficient and intersection-free.
- We develop a differentiable coupling mechanism between cloth and articulated rigid bodies.
- We demonstrate the efficacy of the method for various tasks. We show that our method can efficiently compute derivations and maintain intersection-free properties.

II. Related Work

In this section, we review related literature on key components in our approach, including contact modeling for cloth simulation, and differentiable physics simulation. We describe how we are different from previous work.

A. Differentiable Cloth Simulation

Cloth simulation has been an active research topic with a broad range of applications in computer animation, garment design, and robotics. In recent years, an increasing number of differentiable physics simulation engines support cloth simulation, providing the gradient information to facilitate cloth-related applications [8], [23], [24]. Du *et al.* [7] developed an efficient differentiable soft-body simulator called *Differentiable Projective Dynamics* (*DiffPD*) based on the *Projective Dynamics* [25] with implicit time integration. *DiffCloth* [10] extends *DiffPD* with dry frictional contact [26]. However, *DiffCloth* only works with the vertices-vertices collision detection and response, resulting in severe intersection issues for contact-rich manipulation tasks. Qiao *et al.* [9] proposed a differentiable cloth simulation (a differentiable version of ARCSim [27]) that supports the coupling of deformable objects and rigid bodies but not with articulated bodies. Stuyck [28] proposed a differentiable simulation called *DiffXPBD* by making the Extended Position Based Dynamics (XPBD) formulation differentiable. We develop a differentiable simulator *DiffClothAI* with support for intersection-free frictional contact and differentiable two-way coupling with articulated rigid bodies.

B. Coupling with Articulated Rigid Bodies

Coupling cloth with articulated rigid bodies is a research area with diverse applications in robotics. Macklin *et al.* [22] introduced a unified dynamics framework based on the Position Based Dynamics (PBD) [29] solver, which is efficient but not accurate. Bai *et al.* [21] developed a simulator that enables one-way coupling from rigid bodies to cloth, by using rigid cloth patches approximation. Chen *et al.* [30] introduced a simulation framework for multibody dynamics using Lagrangian nodal displacements. *DiffClothAI* supports two-way coupling between cloth and articulated rigid bodies and makes the whole pipeline differentiable.

III. Preliminary

In this section, we briefly review the project dynamics cloth simulation method with dry frictional contact and its differentiable version [10], the core component of incremental potential contact [18], and the implicit function theorem.

A. Project Dynamics with Dry Frictional Contact

The cloth is considered as a nodal system with m 3D nodes. Denote $x(t), v(t) \in \mathcal{R}^{3m}$ as the positions and velocities of all nodes at time t. Denote h as the time step, and M is the positive diagonal mass matrix of size $3m \times 3m$. $f_{int}, f_{ext} \in \mathcal{R}^{3m}$ are the internal and external forces at each node. The cloth internal forces f_{int} are calculated based on the cloth material model described in [25].

1) Forward Simulation: The standard implicit time-stepping paradigm is given as below

$$\begin{aligned} x_{n+1} &= x_n + hv_{n+1} \\ v_{n+1} &= v_n + hM^{-1}[f_{int}(x_{n+1}) + f_{ext}] \end{aligned} \tag{1}$$

The above implicit time-stepping formulation is equivalent to finding a saddle point of the following energy minimization problem [31].

$$\min_x \frac{1}{2h^2}(x - \hat{x})^T M(x - \hat{x}) + W(x) \tag{2}$$

where $\hat{x} = x_n + hv_n + h^2 M^{-1} f_{ext}$. The potential energy term W defines the internal force f_{int} such that $f_{int} = -\nabla W(x_{n+1})$

Ly *et al.* [32] extended the standard PD framework to work with non-penetration collision and Coulomb friction **assuming the contacts happen only to nodes**. At time step t, a collision detection algorithm is called to identify the cloth's nodes in the collision. For each node in the collision, the Signorini-Coulomb law is applied to ensure its local force r_j and velocity u_j to satisfy one of the following equations, where μ is the frictional coefficient and r_j, u_j are node j's contact force and velocity represented in the local contact frame spanned by the tangential plane and contact normal on the contact surface. The notation $j|T, j|N$ denote the tangential and normal directions of the local frame.

$$\begin{aligned} \text{(Take off)}\ & r_j = 0, u_{j|N} > 0 \\ \text{(Stick)}\ & \|r_{j|T}\| < \mu r_{j|N}, u_j = 0 \\ \text{(Slip)}\ & \|r_{j|T}\| = \mu r_{j|N}, u_{j|N} = 0, \\ & r_{j|T} \parallel u_{j|T}, r_{j|T} \cdot u_{j|T} \leq 0 \end{aligned} \tag{3}$$

Denote the set of $(r_j, u_j) \in \mathcal{C}^j$ satisfying the above conditions.

The implicit time-stepping integration with dry frictional contact is described as below.

$$\begin{cases} x_{n+1} = x_n + hv_{n+1} \\ v_{n+1} = v_n + hM^{-1}[f_{int}(x_{n+1}) + f_{ext} + J_n^T(x_n, v_n) r_{n+1}] \\ u_{n+1} = J_n(x_n, v_n) v_{n+1} \\ (r_{n+1,j}, u_{n+1,j}) \in \mathcal{C}_n^j, \forall j \in \mathcal{I}_n \end{cases} \tag{4}$$

Here we add the subscript n in the definitions of the contact set $\mathcal{I}$ and the contact condition $\mathcal{C}^j$ to specify the time step. and the Jacobian matrix J_n of size $3|\mathcal{I}_n| \times 3m$.

2) Backward Differentiation: We define a nonlinear vector function $C_n^j(r_{n+1,j}, u_{n+1,j})$ and represent the constraint set $C_n^j(r_{n+1,j}, u_{n+1,j}) = 0$. For more detailed description, we refer to *DiffCloth* [10].

$$\begin{cases} v_{n+1} = v_n + hM^{-1}[f_{int}(x_{n+1}) + f_{ext} + J_n^T(x_n, v_n)r_{n+1}] \\ C_n(r_{n+1}, J_n v_{n+1}) = 0 \end{cases} \tag{5}$$

Differentiating both sides of the above equation with respect to v_n gives the following result.

$$\begin{aligned} \frac{\partial v_{n+1}}{\partial v_n} &= I + hM^{-1}[-h\nabla^2 W(x_n + hv_{n+1}) + J_n^T \frac{\partial r_{n+1}}{\partial v_{n+1}}]\frac{\partial v_{n+1}}{\partial v_n} \\ \frac{\partial v_{n+1}}{\partial v_n} &= [I + h^2 M^{-1}\nabla^2 W(x_n + hv_{n+1}) - hM^{-1}J_n^T \frac{\partial r_{n+1}}{\partial v_{n+1}}]^{-1} \\ &= [M + h^2\nabla^2 W(x_n + hv_{n+1}) - hJ_n^T \frac{\partial r_{n+1}}{\partial v_{n+1}}]^{-1} M \end{aligned} \tag{6}$$

Denote

$$\Delta R = [hJ_n^T \frac{\partial r_{n+1}}{\partial v_{n+1}}]^T \tag{7}$$

Backpropagation from v_{n+1} to v_n can be derived with the same adjoint method.

$$\frac{\partial L}{\partial v_n} = M[M + h^2\nabla^2 W(x + hv_{n+1}) - \Delta R]^{-1}\frac{\partial L}{\partial v_{n+1}}. \tag{8}$$

The backpropagation relationship between $\frac{\partial L}{\partial x_{n+1}}$ and $\frac{\partial L}{\partial x_n}$ can be easily derived based on $\frac{\partial L}{\partial v_{n+1}}$ and $\frac{\partial L}{\partial v_n}$.

B. Incremental Potential Contact

IPC introduced the frictional barrier incremental potential denoted below

$$E(x, x^t, v^t) + \kappa \sum_{k\in C} b(d_k(x)) + h^2 \sum_{k\in C} D_k(x) \tag{9}$$

Here C contains all non-incident point-triangle and all non-adjacent edge-edge pairs in the mesh. κ is the parameter controlling the stiffness of the barrier function. $E(x, x^t, v^t)$ is an incremental potential over valid $x \in \mathcal{R}^{3m}$, given nodes' positions x_t and velocities v_t. The IPC under the implicit Euler formulation is given below

$$E(x, x^t, v^t) = \frac{1}{2}(x-\hat{x})^T M(x-\hat{x}) - h^2 x^T f + h^2\Psi(x) \tag{10}$$

$b(d_k(x))$ is the barrier function term and $D_k(x)$ is the friction potential term. These two terms are discussed here.

1) Barrier function: IPC introduced the smoothly clamped barriers, as shown below.

$$b(d, \hat{d}) = \begin{cases} -(d-\hat{d})^2 ln(\frac{d}{\hat{d}}), 0 < d < \hat{d} \\ 0, d \geq \hat{d} \end{cases} \tag{11}$$

The barrier function is C^2 at the clamping threshold, and it is zero beyond the $\hat{d}$. Our contact forces λ_k are then given by barrier derivatives $\lambda_k = -\frac{\kappa}{h^2}\frac{\partial b}{\partial d_k}$. The barrier function guarantees $\frac{\partial b}{\partial d_k} < 0$. Each barrier function's Hessian is given below

$$\mathbf{H}(b) = \frac{\partial^2 b}{\partial d^2}\nabla_x d(\nabla_x d)^T + \frac{\partial b}{\partial d}\nabla_x^2 d \tag{12}$$

2) Friction potential: For each primitive pair k, at current state x, a consistently oriented sliding basis $T_k(x)$ is constructed so that $u_k = T_k(x)^T(x - x^t) \in \mathcal{R}^2$ gives the local relative sliding displacement at contact k. Considering sliding friction and static friction together, we express the friction force as

$$F_k(x) = \mu\lambda_k T_k(x) f(\|u_k\|)s(u_k) \tag{13}$$

When $\|u_k\| > 0$, $s(u_k) = \frac{u_k}{\|u_k\|}$ and $f(\|u_k\|)$ = 1. When $\|u_k\| = 0$, $s(u_k)$ takes any 2D unit value and $f(\|u_k\|) \in [0, 1]$. This function can be smoothed and f is approximately replaced by

$$f_1(y) = \begin{cases} -\frac{y^2}{\epsilon_v^2 h^2} + \frac{2y}{\epsilon_v h}, & y \in [0, h\epsilon_v) \\ 1, & y \leq h\epsilon_v \end{cases} \tag{14}$$

The friction potential is defined by integration below

$$D_k(x) = \mu\lambda_k^n f_0(\|u_k\|) \tag{15}$$

where f_0 is given by the relation $f_0' = f_1$ and $f_0(\epsilon_v h) = \epsilon_v h$ so that $F_k(x) = -\nabla_x D_k(x)$. The Hessian function is given by

$$\begin{aligned} \mathbf{H}(D_k(x)) =& \mu\lambda_k^n T_k^n (\frac{f_1'(\|u_k\|)\|u_k\| - f_1(\|u_k\|)}{\|u_k\|^3} u_k u_k^T \\ &+ \frac{f_1(\|u_k\|)}{\|u_k\|} I_2) T_K^{n\,T}. \end{aligned} \tag{16}$$

C. Implicit Function Theorem

We consider an unconstrained optimization problem.

$$\min_z c(z; \theta) \tag{17}$$

where c is the cost function, $z \in \mathcal{R}^a$ is the optimization variable, $\theta \in \mathcal{R}^b$ is the problem parameters. Denote the optimal value of $z^*(\theta) = \arg\min c(z(\theta))$.

Denote the f as the gradient function of c at the optimal value. The implicit function $f : \mathcal{R}^a \times \mathcal{R}^b \to \mathcal{R}^a$ is defined as:

$$f(z^*, \theta) = 0 \tag{18}$$

for an optimal point $z^* \in \mathcal{R}^a$ and problem parameters $\theta \in \mathcal{R}^b$.

We can approximate Eqn. III-C with a first-order Taylor series results in

$$\frac{\partial f}{\partial z}\delta z + \frac{\partial f}{\partial \theta}\delta\theta = 0 \tag{19}$$

The above equation should be re-arranged to solve the sensitivities of the solution with respect to the problem parameters.

$$\frac{\partial z*}{\partial \theta} = -(\frac{\partial f}{\partial z})^{-1}\frac{\partial f}{\partial \theta} \tag{20}$$

IV. Differentiable Simulation Design

Forward cloth simulation pipeline updates velocities and positions at each discrete timestep by solving constraints and collision. The Projective Dynamics method (PD) is a stable method for implicit time integration for the cloth system, and positional constraints on the vertices control the morphological changes of the cloth. However, the vertex-vertex contact and friction in Projective Dynamics is not sufficient. Incremental Potential Contact (IPC) keep the cloth intersection-free and provides features such as friction calculation. We integrate IPC with the Projective Dynamics method to solve the constraints and collision in the simulation system, and we also provide backward gradients calculation for the whole pipeline. In addition, we provide a differentiable bidirectionally coupled rigid cloth system based on our differentiable cloth simulator.

A. Forward Simulation

We denote $\mathcal{F}^D$ as the forward step function from *DiffCloth*, which takes the nodes position and velocity x_n, v_n at time step n and outputs the velocity at the next time step v^D_{n+1} under the fry frictional contact. The positions c^D_{n+1} at next time is calculated based on v^D_{n+1}.

$$\begin{aligned} v^D_{n+1} &= \mathcal{F}^D(x_n, v_n) \\ x^D_{n+1} &= x_n + h v^D_{n+1} \end{aligned} \tag{21}$$

However, *DiffCloth* only considers vertices-vertices collision detection and response. The intersection, including self-intersection, might happen at the position x^D_{n+1}.

We design an objective function denoted as $P(x; x^D_{n+1})$ as below

$$\begin{aligned} P(x, x^D_{n+1}) =& \|x - x^D_{n+1}\|^2 + \kappa \sum_{k\in C} b(d_k(x)) \\ &+ \gamma h^2 \sum_{k\in C} D_k(x) \\ =& \|x - x^D_{n+1}\|^2 + \kappa B_d(x) + \gamma D_f(x) \\ B_d(x) =& \sum_{k\in C} b(d_k(x)) \\ D_f(x) =& h^2 \sum_{k\in C} D_k(x) \end{aligned} \tag{22}$$

We aim to find the suitable x_{n+1} near the calculated x^D_{n+1} based on Diffcloth while maintaining the intersection-free property.

$$x_{n+1} = \arg\min P(x; x^D_{n+1}) \tag{23}$$

We start from the x_n, which is the intersection-free nodes' position. The optimization is solved through Newton-type barrier solver Alg.1. For termination of this solver, we check the infinity norm of the Newton search direction scaled by time step and reach an accuracy satisfying $\frac{1}{h}\|H^{-1}\nabla_x P(x, \hat{d}, \hat{C})\|_\infty < \epsilon$. Solution accuracy is then directly defined by $h\epsilon$ in physical units of position, which measures quadratically approximated distance to local optimal. More detailed explanations are put the supplementary on our website.

Algorithm 1 Barrier Aware Newton Solver

procedure BarrierAwareProjectedNewton(x^t, ϵ)
  $x \leftarrow x^t$
  $\hat{C} \leftarrow$ ComputeConstraintSet($x, \hat{d}$)
  $E_{prev} \leftarrow P(x; x^D, \hat{d}, \hat{C})$
  $x_{prev} \leftarrow x$
  do
    $H \leftarrow \nabla^2_x P(x, \hat{d}, \hat{C}))$
    $p \leftarrow -H^{-1}\nabla_x P(x, \hat{d}, \hat{C})$
    $\alpha \leftarrow \min(1, \text{StepSizeUpperBound}(x, p, \hat{C})$
    do
      $\alpha \leftarrow \alpha/2$
      $x \leftarrow x_{prev} + \alpha p$
      $\hat{C} \leftarrow$ ComputeConstraintSet($x, \hat{d}$)
    while $P(x, \hat{d}, \hat{C}) > E_{prev}$
    $E_{prev} \leftarrow P(x, \hat{d}, \hat{C})$
    $x_{prev} - x$
  while $\frac{1}{h}\|p\|_\infty > \epsilon$
  return x
end procedure

Once we find the nodes' position at next time x_{n+1}, Our simulation calculates the updated nodes' velocities as below.

$$v_{n+1} = \frac{1}{h}(x_{n+1} - x_n) \tag{24}$$

where the calculated x_{n+1} and v_{n+1} are fed into the Diffcloth forward function to get the next velocities and positions v^D_{n+2}, x^D_{n+2}.

$$\begin{aligned} v^D_{n+2} &= \mathcal{F}^D(x_{n+1}, v_{n+1}) \\ x^D_{n+2} &= x_{n+1} + h v^D_{n+2} \end{aligned} \tag{25}$$

B. Backward Gradients

The gradient

$$\frac{\partial L}{\partial x_n} = \frac{\partial L}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial x^D_{n+1}} \frac{\partial x^D_{n+1}}{\partial x_n} \tag{26}$$

The gradient $\frac{\partial L}{\partial x_{n+1}}$ is given and the gradient $\frac{\partial x^D_{n+1}}{\partial x_n}$ is provided by the *DiffCloth* simulation. Here we describe how to derive the term $\frac{\partial x_{n+1}}{\partial x^D_{n+1}}$.

Potential function $P(x, x^D_{n+1})$ at the optimal value x^*_{n+1} satisfies the optimal condition

$$\begin{aligned} &\nabla_x P(x^*_{n+1}, x^D_{n+1}) \\ =& (x^*_{n+1} - x^D_{n+1}) + \nabla B_d(x^*_{n+1}) + \nabla D_f(x^*_{n+1}) \\ =& 0 \end{aligned} \tag{27}$$

With the implicit function theorem, let $f(x^*_{n+1}; x^D_{n+1}) = \nabla_x P(x^*_{n+1}, x^D_{n+1}) = 0$. We get the following equations.

$$\begin{aligned} &\mathbf{I}\delta x^*_{n+1} - \mathbf{I}\delta x^D_{n+1} + \mathbf{H}(B_d)\delta x^*_{n+1} + \mathbf{H}(D_f)\delta x^*_{n+1} = 0 \\ &(\mathbf{I} + \mathbf{H}(B_d) + \mathbf{H}(D_f))\delta x^*_{n+1} = \mathbf{I}\delta x^D_{n+1} \end{aligned} \tag{28}$$

Here $\mathbf{I}$ is the identify matrix, $\mathbf{H}(B_d)$ and $\mathbf{H}(D_f)$ are Hessian functions.

It is easy to derive

$$\frac{\partial x^*_{n+1}}{\partial x^C_{n+1}} = (\mathbf{I} + \mathbf{H}(B_d) + \mathbf{H}(D_f))^{-1} \tag{29}$$

Here $\mathbf{H}(B_d)+\mathbf{H}(D_f)$ are available when we find the x^*_{n+1} through Newton method. So we can calculate the backward gradient from the $n+1$ time step to the n time step.

C. Coupling between Articulated Objects and Cloth

We denote the state and action of cloth at timestep n as s^D_n and a^D_n. The forward simulation for cloth is defined as $\mathcal{F}^D$, with $s^D_{n+1} = \mathcal{F}^D(s^D_n, a^D_n)$. We denote the state and action of the articulated rigid body to be s^R_n and a^R_n. The forward simulation for articulated rigid bodies is defined as $\mathcal{F}^R$, with $s^R_{n+1} = \mathcal{F}^R(s^R_n, a^R_n)$. With the differentiable operations inside each simulation, we can gather their corresponding gradient as follow:

$$\begin{matrix} \frac{\partial s^D_{n+1}}{\partial s^D_n}, \frac{\partial s^D_{n+1}}{\partial a^D_n} \\ \frac{\partial s^R_{n+1}}{\partial s^R_n}, \frac{\partial s^R_{n+1}}{\partial a^R_n} \end{matrix} \tag{30}$$

While the forward and backward simulation of cloth and articulated rigid body can be conducted respectively, we can't simulate the contact between these two different types of objects and get the gradient of one type w.r.t another type. So the coupled differentiable simulation needs to be proposed to tackle this problem. We define the state and action of cloth and articulated rigid bodies as $s_n = (s^D_n, s^R_n)$, $a_n = (a^D_n, a^R_n)$, and the coupled forward simulation function as:

$$s_{n+1} = (s^D_{n+1}, s^R_{n+1}) = \mathcal{F}(s_n, a_n) \tag{31}$$

Specifically, the unified simulation $\mathcal{F}$ has two components denoted as $\mathcal{F}_1$ and $\mathcal{F}_2$:

$$s^D_{n+1} = \mathcal{F}_1(s_n, a_n) \quad s^R_{n+1} = \mathcal{F}_2(s_n, a_n) \tag{32}$$

We describe $\mathcal{F}_1$ in subsection of *Rigid2Cloth* and $\mathcal{F}_2$ in the subsection of *Cloth2Rigid*. We can also calculate their gradient:

$$\begin{matrix} \frac{\partial s^D_{n+1}}{\partial s^R_n}, \frac{\partial s^D_{n+1}}{\partial a^R_n} \\ \frac{\partial s^R_{n+1}}{\partial s^D_n}, \frac{\partial s^R_{n+1}}{\partial a^D_n} \end{matrix} \tag{33}$$

1) Rigid2Cloth: Here we describe how to control articulated rigid bodies to interact with the cloth. In forward simulation, we first calculate articulated rigid bodies' next state $s^R_{n+1} = \mathcal{F}^R(s^R_n, a^R_n; r^R_n)$ based on differentiable simulation for articulated rigid bodies. Here r^R_n is the external force (collision impulse from the cloth). Given the articulated object at next time, we calculate the cloth next state $s^D_{n+1} = \mathcal{F}^D(s^D_n, a^D_n; s^R_{n+1})$. In the process, the collision impact between articulated rigid bodies and cloth is calculated on cloth simulation denoted as r^R_{n+1}. This collision impact r_{n+1} is not included in the backward computation graph. So we modify the forward function of cloth simulation to be changed from $\mathcal{F}^D(s^D_n, a^D_n)$ to $\mathcal{F}^{D'}(s^D_n, a^D_n, r^R_{n+1})$. It turns out that the $s^D_{n+1} = \mathcal{F}_1(s_n, a_n) = \mathcal{F}^{D'}(s^D_n, a^D_n, r^R_{n+1})$. We plot the relationship within $[s^D_{n+1}, s^R_{n+1}, s^D_n, s^R_n, a^R_n, a^D_n]$ during *Rigid2Cloth* in Fig. 2. More detailed explanations are put in the supplementary on our website.

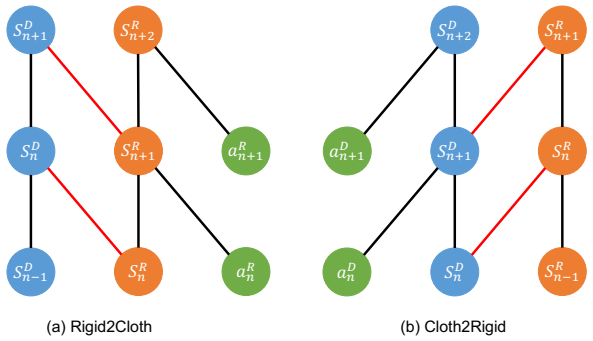

Fig. 2. Computation Graph of Coupled Simulation. We use blue nodes to denote the state of Cloth Simulation, orange nodes to denote the state of Articulated Rigid-body Simulation, and green nodes to denote actions. A line is connected between two nodes if and only if we directly use one state or action to compute the resulting state, with the black line standing for the computation inside one single simulation and the red line for the computation between Cloth Simulation and Articulated Rigid-body Simulation.

The s^D_{n+1} is determined by s^D_n, a^D_n and s^R_{n+1} (due to the r^R_{n+1}). In other word, the s^D_t is always affected by s^R_t, where $t \in [1, 2, 3..., T]$, indicating the gradient $\frac{s^D_{n+1}}{s^R_{n+1}}$ and $\frac{s^D_n}{s^R_n}$ exist. In backward simulation, we discuss how to calculate the gradients:

$$\frac{\partial s^D_{n+1}}{\partial s^R_n} = \frac{\partial s^D_{n+1}}{\partial s^D_n}\frac{\partial s^D_n}{\partial s^R_n} + \frac{\partial s^D_{n+1}}{\partial s^R_{n+1}}\frac{\partial s^R_{n+1}}{\partial s^R_n} \tag{34}$$

$$\frac{\partial s^D_{n+1}}{\partial a^R_n} = \frac{\partial s^D_{n+1}}{\partial s^R_{n+1}}\frac{\partial s^R_{n+1}}{\partial a^R_n} \tag{35}$$

Then we discuss how to calculate the $\frac{\partial s^D_n}{\partial s^R_n}$ term in the supplementary material in our website due to page limit at website.

2) Cloth2Rigid: Here we describe how to control cloth to interact with articulated rigid bodies. We first calculate the cloth's next state $s^D_{n+1} = \mathcal{F}^D(s^D_n, a^D_n; s^R_n)$ based on the cloth simulation along with the articulated rigid bodies state s^R_n. The collision between s^D_{n+1} and s^R_n is detected and we derive the collision impulse/response between the cloth and articulated rigid bodies denoted as r^D_n. Then the articulated rigid bodies' next state is calculated based on its current state, action, and the external force exerted by the cloth $s^R_{n+1} = \mathcal{F}^R(s^R_n, a^R_n, r^D_n)$.

We shown the relationship within $[s^D_{n+1}, s^R_{n+1}, s^D_n, s^R_n, a^R_n, a^D_n]$ during *Cloth2Rigid* in Fig. 2. More detailed explanations are put in the supplementary.

The s^R_{n+1} is determined by s^R_n, a^R_n and s^D_{n+1} (due to the r^D). In other word, the s^R_t is always affected by s^D_t, where $t \in [1, 2, 3..., T]$, indicating the gradient $\frac{s^R_{n+1}}{s^D_{n+1}}$ and $\frac{s^D_n}{s^R_n}$ exist. The gradients are calculated below:

$$\frac{\partial s^R_{n+1}}{\partial s^D_n} = \frac{\partial s^R_{n+1}}{\partial s^R_n}\frac{\partial s^R_n}{\partial s^D_n} + \frac{\partial s^R_{n+1}}{\partial s^D_{n+1}}\frac{\partial s^D_{n+1}}{\partial s^D_n} \tag{36}$$

$$\frac{\partial s_{n+1}^R}{\partial a_n^D} = \frac{\partial s_{n+1}^R}{\partial s_{n+1}^D} \frac{\partial s_{n+1}^D}{\partial a_n^D} \tag{37}$$

Then we discuss how to calculate the $\frac{\partial s_n^R}{\partial s_n^D}$ term in the supplementary material at our website due to page limit.

V. Experiments

Our experiments evaluate the following questions: 1) Can our differentiable simulation produce intersection-free simulation results for contact-rich manipulation tasks? 2) How accurate is our differential simulation gradient calculation? 3) How effective is our proposed differentiable coupling mechanism between cloth and articulated rigid bodies?

A. Implementation

In this work, we develop our differentiable simulation based on the open-source implementation of *DiffCloth* [10], *IPC* [18], and *Nimble* [11]. For intersection-free contact, the previous points' position and velocity, together with the projective dynamics' output is fed into the IPC. Then, the IPC optimizes the value of the objective function as Algorithm 1 describes; For coupling between Cloth and Articulated rigid body, we use DiffCloth and Nimble as the base of our cloth simulation and rigid body simulation, respectively. By taking these two simulators' states as input, calculating their contacts, and then putting the interaction results back into simulators, we can get the consequent next step states. Gradients are also calculated correspondingly.

B. Checking Intersection-free Quality

1) Self-Intersection: We demonstrate the intersection-free quality of our algorithm using a tie-wearing experiment. The tie is about 1.4 meters in length and easily bent. When wearing a tie, the tie easily gets intertwined. Thus, its shape is hard to be estimated by the simulator. In our experiment, the tie is in a complex form. Our target is to move control points of the tie to some specific positions.

We compare the simulation results based on our DiffClothAI with the origin DiffCloth. Besides we also perform ablation experiments to demonstrate the effectiveness of the component in our simulation. Specifically, we remove the term of the frictional potential term, which is the $D_f(x)$ in our defined potential function.

One step of the simulation result of tie-wearing is shown as Fig. 3(Large resolution images/animations are available on our website). In the above experiments, we can conclude that our DiffClothAI algorithm is able to avoid intersections in the simulation process. However, using DiffClothAI without the friction potential, the simulation process gets stuck easily. With the help of the friction potential, the simulation process can move forward.

2) Cloth and Rigid Body: In this experiment, we pull a piece of cloth downward onto a rigid cube. We aim to demonstrate that our algorithm outperforms DiffCloth in avoiding deformable cloth and rigid body intersections. The comparison result is shown in the Fig. 4.

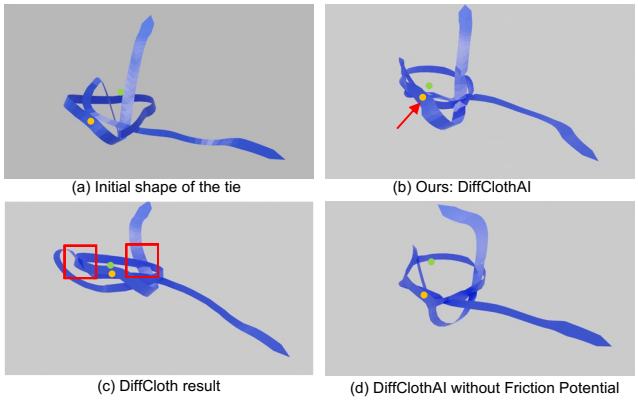

Fig. 3. The orange point describes one of the control points. The green point describes its target position. (a) is the initial shape of the tie. (b) is the result of DiffClothAI. The result is intersection-free. The control point can move toward the target position successfully. (c) is the simulation result of the tie with the original DiffCloth. We can see that interpenetration happens in the red rectangles. (d) is the simulation result of our DiffClothAI without friction potential. However, it happens that the simulator gets stuck at a specific position.

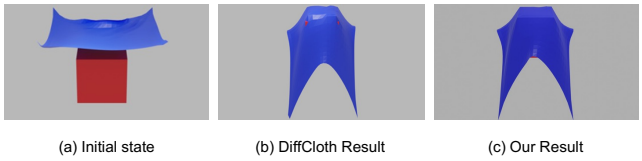

Fig. 4. Experiments results of the intersection-free quality checking between cloth and rigid body. (a) is the initial state. In (b) and (c), the cloth is stretched to a point of being severely deformed. (b) is the result of DiffCloth. We can see intersections happen around the corners of the cube. (c) is our result, intersections are successfully avoided.

In this experiment, we can see that DiffCloth may meet serious interpenetrations in the case of edge-edge collision. Our algorithm can successfully avoid such failure in even extreme cases.

C. Checking Differentiation Calculation

In this experiment, two pieces of clothes are pulled in opposite directions toward each other. Our target is to move the center of one of the clothes to a target position. A loss is calculated as the L_2 distance between the cloth center's position and the target position. The gradient is backward to the action, which is the next step position of the control points. The experiment is described in Fig. 6.

The loss curve is shown in Fig. 6 (d). The loss decreases correctly. This experiment indicates the effectiveness of our proposed differentiation calculation.

D. Checking Coupling Quality

To evaluate the quality of our coupled differentiable simulation between cloth and (articulated) rigid body, we design three contact-rich tasks and utilize the backward gradient to update trajectories. Stochastic gradient descent is used as our gradient-based optimization algorithm for all three tasks. Experiment results (optimized trajectories visualization and

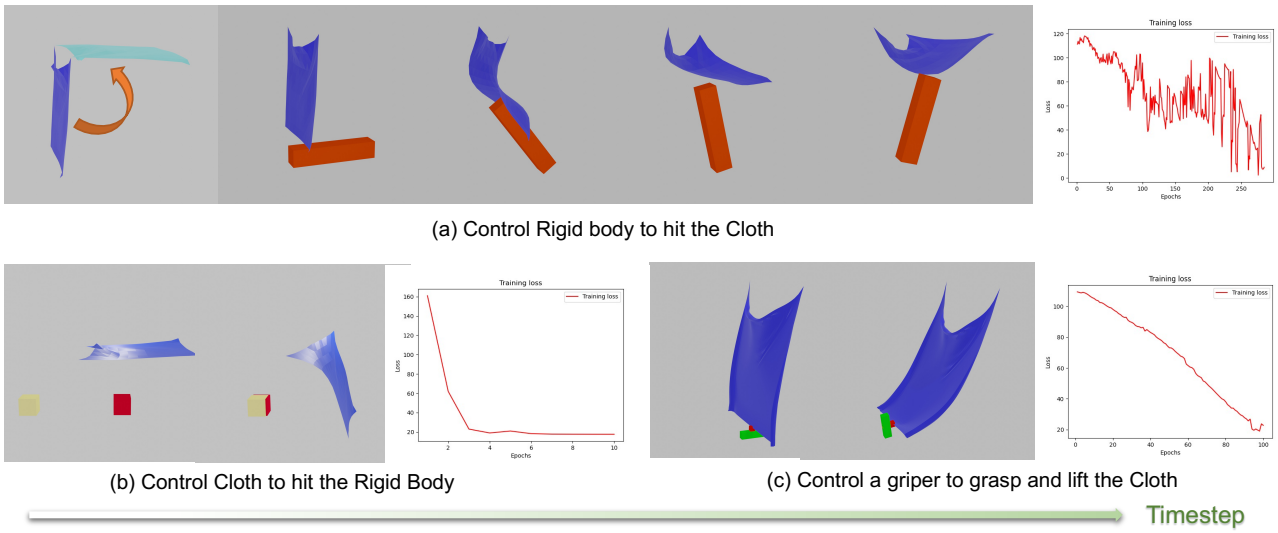

Fig. 5. Experiment results of coupled differentiable simulation. Part (a) depicts initial fabric shape in blue, while the target shape is shown in cyan. Selected frames from our optimized trajectory are also displayed. Part (b) highlights the first and last frames of the optimized trajectory, where the red cube represents the initial position and the faint yellow cube represents the target position. Part (c) provides a detailed visualization of the optimized control procedure for grasping and lifting the fabric using a two-finger gripper. Additionally, on the right side of each subfigure, their respective training loss curves with respect to traing epochs for all three tasks are plotted, enabling a comprehensive evaluation of the optimization process.

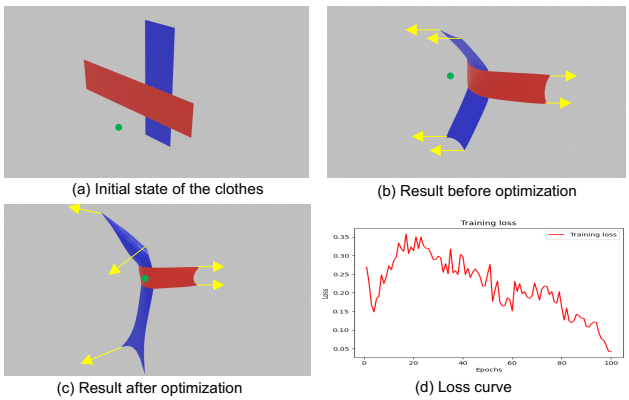

Fig. 6. Experiment results of differentiation calculation checking. Part (a) shows the initial state of two fabrics, while green point indicates the blue fabric's target center. Part (b) highlights the result before optimization. The control force added to red fabric is constant, with the yellow arrows indicate the force direction. We only optimize the force added to the blue fabric. Part (c) provides result after optimization. Part (d) shows the loss curve.

training loss) reported in Fig. 5 reveal the correctness and efficiency of our coupled differentiable simulation. The state spaces in each task is the combination of DoFs of rigid body and a 1728 DoFs square fabric. We explain the target shape and more detailed training process in the supplementary materials provided at our website.

1) Control Rigid body to hit the Cloth: This task is meant to test our forward simulation result of **Rigid2Cloth**, as well as the correctness of its corresponding gradient. We try to use a 6 DoFs (3 DoFs in translation and 3 DoFs in rotation) cuboid to hit the square fabric in a vertical rest shape into a horizontal target shape. With the loss function defined as the L_2 distance between the current shape and the target shape of the fabric, we optimize the 6 DoFs action space to minimize the objective using SGD as the optimizer.

2) Control Cloth to hit the Rigid Body: In this task, we verify both forward and backward simulation results of another coupled part **Cloth2Rigid**. Using 2 control points on the cloth (upper left corner and upper right corner), each with 3 DoFs, we manage to control the cloth to push a cube into the target position. With the loss function defined as the L_2 distance between the current position and the target position of the cube, we optimize these 2 control points to achieve the goal.

3) Control a griper to grasp and lift the Cloth: While our coupled simulation does well in the interaction between cloth and simple rigid body, as shown before, we also want to verify its effectiveness in simulating the contact between cloth and articulated rigid body, which is of great importance for robotics manipulation tasks. So in this experiment, we try to control an 8 DoFs gripper to grasp and lift a square fabric from rest shape to target shape as shown in Fig. 5 (c). This gripper is consisted of one hand and two fingers, with 6 Dofs in the hand and 2 Dofs in two fingers. With the objective function defined as the L_2 distance between the current shape and target shape of the cloth, we use SGD to optimize this 8 DoFs action space.

VI. Conclusion

In this work, we introduce *DiffClothAI*, a unified differentiable simulation framework with intersection-free friction contact and two-way coupling with articulated rigid bodies. To achieve this, *DiffClothAI* integrates the Project Dynamics and Incremental Potential Contact, offering an effective approach to compute gradients within the simulation. Additionally, we develop the differentiable coupling mechanism between articulated rigid bodies and cloth. We

conduct extensive experiments to evaluate the effectiveness and accuracy of *DiffClothAI* for various downstream robotic manipulation tasks.

REFERENCES

[1] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[2] J. Degrave, M. Hermans, J. Dambre, *et al.*, "A differentiable physics engine for deep learning in robotics," *Frontiers in neurorobotics*, p. 6, 2019.

[3] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand, "Taichi: a language for high-performance computation on spatially sparse data structures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, p. 201, 2019.

[4] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," in *2019 International conference on robotics and automation (ICRA)*, pp. 6265–6271, IEEE, 2019.

[5] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Lévesque, K. Xie, K. Erleben, *et al.*, "gradsim: Differentiable simulation for system identification and visuomotor control," *arXiv preprint arXiv:2104.02646*, 2021.

[6] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, "Add: Analytically differentiable dynamics for multi-body systems with frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.

[7] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik, "Diffpd: Differentiable projective dynamics," *ACM Trans. Graph.*, vol. 41, nov 2021.

[8] J. Liang, M. Lin, and V. Koltun, "Differentiable cloth simulation for inverse problems," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[9] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Scalable differentiable physics for learning and control," *arXiv preprint arXiv:2007.02168*, 2020.

[10] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik, "Diffcloth: Differentiable cloth simulation with dry frictional contact," *ACM Transactions on Graphics (TOG)*, 2022.

[11] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics for articulated rigid bodies with contact," *arXiv preprint arXiv:2103.16021*, 2021.

[12] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Joint optimization of robot design and motion parameters using the implicit function theorem," in *Robotics* (S. Srinivasa, N. Ayanian, N. Amato, and S. Kuindersma, eds.), Robotics: Science and Systems, (United States), MIT Press Journals, 2017. Publisher Copyright: © 2017 MIT Press Journals. All rights reserved.; 2017 Robotics: Science and Systems, RSS 2017 ; Conference date: 12-07-2017 Through 16-07-2017.

[13] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Efficient differentiable simulation of articulated bodies," in *International Conference on Machine Learning*, pp. 8661–8671, PMLR, 2021.

[14] K. Um, R. Brand, Y. R. Fei, P. Holl, and N. Thuerey, "Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6111–6122, 2020.

[15] N. Wandel, M. Weinmann, and R. Klein, "Learning incompressible fluid dynamics from scratch–towards fast, differentiable fluid models that generalize," *arXiv preprint arXiv:2006.08762*, 2020.

[16] P. Holl, V. Koltun, and N. Thuerey, "Learning to control pdes with differentiable physics," *arXiv preprint arXiv:2001.07457*, 2020.

[17] T. Takahashi, J. Liang, Y.-L. Qiao, and M. C. Lin, "Differentiable fluids with solid coupling for learning and control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 6138–6146, May 2021.

[18] 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. (SIGGRAPH)*, vol. 39, no. 4, 2020.

[19] A. Gjoka, Z. Huang, D. C. Tozoni, Z. Ferguson, T. Schneider, D. Panozzo, and D. Zorin, "Differentiable solver for time-dependent deformation problems with contact," *arXiv preprint arXiv:2205.13643*, 2022.

[20] T. Schneider, J. Dumas, X. Gao, D. Zorin, and D. Panozzo, "PolyFEM." https://polyfem.github.io/, 2019.

[21] Y. Bai and C. K. Liu, "Coupling cloth and rigid bodies for dexterous manipulation," in *Proceedings of the 7th International Conference on Motion in Games*, MIG '14, (New York, NY, USA), p. 139–145, Association for Computing Machinery, 2014.

[22] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, "Unified particle physics for real-time applications," *ACM Trans. Graph.*, vol. 33, jul 2014.

[23] Y. Qiao, J. Liang, V. Koltun, and M. Lin, "Differentiable simulation of soft multi-body systems," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17123–17135, 2021.

[24] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "Difftaichi: Differentiable programming for physical simulation," *ICLR*, 2020.

[25] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly, "Projective dynamics: Fusing constraint projections for fast simulation," *ACM transactions on graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[26] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes, "Projective dynamics with dry frictional contact," *ACM Trans. Graph.*, vol. 39, aug 2020.

[27] R. Narain, A. Samii, and J. F. O'Brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM Trans. Graph.*, vol. 31, nov 2012.

[28] T. Stuyck, "Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics," *arXiv preprint arXiv:2301.01396*, 2023.

[29] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109–118, 2007.

[30] Y. Chen, M. Li, L. Lan, H. Su, Y. Yang, and C. Jiang, "A unified newton barrier method for multibody dynamics," *ACM Trans. Graph.*, vol. 41, jul 2022.

[31] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross, "Example-based elastic materials," in *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, (New York, NY, USA), Association for Computing Machinery, 2011.

[32] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes, "Projective dynamics with dry frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 57–1, 2020.

Supplementary Material

I. Coupling Force between Cloth and Articulated Rigid Bodies

A. The derivation of $\frac{S^R_{t+1}}{S^D_{t+1}}$

Ly *et al.* [1] extend the standard PD framework to work with non-penetration collision and Coulomb friction assuming the contacts happen only to nodes. At time step t, a collision detection algorithm is called to identify the cloth's nodes which are in collision. For each node in collision, the Signorini-Coulomb law is applied to ensure its local force r_j and velocity u_j to satisfy one of the following equations, where μ is the frictional coefficient, and r_j, u_j are node j's contact force and velocity represented in the local contact frame spanned by the tangential plane and contact normal on the contact surface. The notation $j|T, j|N$ denote the tangential and normal direction of the local frame.

$$\begin{aligned} &\text{(Take off)}\ r_j = 0, u_{j|N} > 0 \\ &\text{(Stick)}\ \|r_{j|T}\| < \mu r_{j|N}, u_j = 0 \\ &\text{(Slip)}\ \|r_{j|T}\| = \mu r_{j|N}, u_{j|N} = 0, \\ &\qquad r_{j|T} \parallel u_{j|T}, r_{j|T} \cdot u_{j|T} \leq 0 \end{aligned} \tag{1}$$

Denote the set of $(r_j, u_j) \in \mathcal{C}^j$ satisfying the above conditions.

For the forward process, we first get the state of cloth at $t+1$ by the forward function of DiffCloth:

$$F^D(s^D_t, s^R_t, a^D_t) = s^D_{t+1}$$

, where a^D_t is the action for the cloth. We then perform collision detection between s^D_{t+1} and s^R_t. We get the set of collision vertices on cloth mesh denoted as $x^C_{t+1} = \{x_{t+1,j}\}$, where $x_{t+1,j}$ is the vertex on the cloth mesh in contact with the surrounding objects. For every collision point $x_{t+1,j}$, we have a corresponding collision force $r_{t+1,j}$, calculated by equation 1. We denote the $r_{t+1,j}$ to represent the contact force exerted by the rigid body onto the vertex $x_{t+1,j}$ on cloth mesh and let $f_{t+1,j} = -r_{t+1,j}$ to be the contact force exerted onto the rigid bodies according to Newton's third law. For simplicity, we denote the $f^C_{t+1} = \{f_{t+1,j}\}$.

Now the next state of rigid bodies at time $t+1$ is derived following the below equation.

$$\mathcal{F}^R(s^R_t, a^R_t, f^C_{t+1}, x^C_{t+1}) = s^R_{t+1}$$

Here x^C_{t+1} is needed to calculate the applied torques to the rigid bodies.

For the backward process, we can get the gradient of function $\mathcal{F}^R$ from Nimble:

$$\frac{\partial s^R_{t+1}}{\partial f^C_{t+1}}, \frac{\partial s^R_{t+1}}{\partial x^C_{t+1}} \tag{2}$$

Note that s^D_{t+1} contains the vertices' positions and velocities on the cloth mesh denoted as x^D_{t+1} and v^D_{t+1}. So that

$$\frac{\partial s^R_{t+1}}{\partial s^D_{t+1}} = \frac{\partial s^R_{t+1}}{\partial x^D_{t+1}}, \frac{\partial s^R_{t+1}}{\partial v^D_{t+1}} \tag{3}$$

Note that we denote x^C_{t+1} as the set of vertices which are in contact with rigid bodies. Thus, x^C_{t+1} is the sub set of x^D_{t+1}. For the remaining vertices which are not in contact with the rigid bodies, their $\frac{\partial s^R_{t+1}}{\partial x^D_{t+1}}$ and $\frac{\partial s^R_{t+1}}{\partial f^D_{t+1}}$ are zero.

We then discuss how to calculate $\frac{\partial s^R_{t+1}}{\partial x^D_{t+1}}, \frac{\partial s^R_{t+1}}{\partial v^D_{t+1}}$ for vertices in x^C_{t+1}.

First, we can easily the results of $\frac{\partial s^R_{t+1}}{\partial x^D_{t+1}} = \frac{\partial s^R_{t+1}}{\partial x^C_{t+1}}$.

We then discuss how to derive $\frac{\partial s^R_{t+1}}{\partial v^D_{t+1}}$. According to the equation 8 in section 3.3 of Ly *et al.* [1], we have:

$$Mv_{t+1} = \tilde{b} - Cv_t + r_{t+1}h \tag{4}$$

where M is the (diagonal positive) mass-matrix, C is a positive matrix, and $\tilde{b}$ is a vector, h is the time step. For detailed explanation, we refer you to [1].

$$\begin{aligned} &\frac{\partial r_{t+1}}{\partial v_{t+1}} = \frac{M}{h} \\ &\frac{\partial s^R_{t+1}}{\partial v^D_{t+1}} = \frac{\partial s^R_{t+1}}{\partial f^C_{t+1}} \frac{\partial f^C_{t+1}}{\partial r_{t+1}} \frac{\partial r_{t+1}}{\partial v_{t+1}} = -\frac{\partial s^R_{t+1}}{\partial f^C_{t+1}} \frac{M}{h} \end{aligned} \tag{5}$$

where $\frac{\partial s^R_{t+1}}{\partial f^C_{t+1}}$ is already provided.

B. The derivation of $\frac{\partial S^D_{t+1}}{\partial S^R_{t+1}}$

The derivation of $\frac{\partial S^D_{t+1}}{\partial S^R_{t+1}}$ is almost a reverse process of derivation of $\frac{\partial S^R_{t+1}}{\partial S^D_{t+1}}$. As the same with section A, a collision detection algorithm is first called between the cloth S^D_t and rigid body S^R_{t+1}. The only interaction between S^D_{t+1} and S^R_{t+1} is the contact force r_{t+1} exerted by the rigid body onto the cloth mesh. So the calculation of $\frac{\partial S^D_{t+1}}{\partial S^R_{t+1}}$ can be simply expanded as:

$$\frac{\partial S^D_{t+1}}{\partial S^R_{t+1}} = \frac{\partial S^D_{t+1}}{\partial r_{t+1}} \frac{\partial r_{t+1}}{\partial S^R_{t+1}} \tag{6}$$

According to equation 4, we have:

$$\frac{\partial S^D_{t+1}}{\partial r_{t+1}} = M^{-1}h \tag{7}$$

As for $\frac{\partial r_{t+1}}{\partial S^R_{t+1}}$, it calculates how the small perturbations of the rigid bodies affect the contact force exerted onto the

cloth mesh. The gradient calculation of $\frac{\partial r_{t+1}}{\partial S^R_{t+1}}$ is provided by DiffCloth [2].

REFERENCES

[1] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes, "Projective dynamics with dry frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 57–1, 2020.

[2] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik, "Diffcloth: Differentiable cloth simulation with dry frictional contact," *ACM Transactions on Graphics (TOG)*, 2022.