Rod-Bonded Discrete Element Method

Kangrui Zhang[a], Han Yan[a], Jia-Ming Lu[b] and Bo Ren[a,*]

[a]TMCC, College of Computer Science, Nankai University, China
[b]Tsinghua University, China

ARTICLE INFO

Keywords:
discrete element method
physically based animation
fracture simulation
elastic rods

ABSTRACT

The Bonded Discrete Element Method(BDEM) has raised interests in the graphics community in recent years because of its good performance in fracture simulations. However, current explicit BDEM usually needs to work under very small time steps to avoid numerical instability. We propose a new BDEM, namely Rod-BDEM (RBDEM), which uses Cosserat energy and yields integrable forces and torques. We further derive a novel Cosserat rod discretization method to effectively represent the three-dimensional topological connections between discrete elements. Then, a complete implicit BDEM system integrating the appropriate fracture model and contact model is constructed using the implicit Euler integration scheme. Our method allows high Young's modulus and larger time steps in elastic deformation, breaking, cracking, and impacting, achieving up to 8 times speed up of the total simulation.

1. Introduction

Bonded Discrete Element Method(BDEM) is a simple and effective method for modeling the deformation of solids including granular matter, rocks and cementitious material[43], and is especially good at modeling the fracture of solids. As a particle method, it does not make a continuum assumption, i.e each particle does not represent a small continuum, but directly represents a quasi-rigid body. Due to the discontinuity of the BDEM system, it is naturally used for fracture simulation, capable of simulating crack generation and multiple fractures efficiently. Bonds are used to connect adjacent particles to simulate elastic behavior, and the failure of bonds is controlled by a local fracture model. In the graphics community, Lu et al. [30] reformulated BDEM by using quaternions to represent orientation, and avoided additional storage of incremental force and torque. They simulated a wide range of materials including elastic bodies, thin shell structures, anisotropic materials and textile. However, conventionally, BDEM uses explicit integration methods, such as explicit Euler or velocity verlet, which need to use very small time steps to ensure numerical stability.

A common implicit integration scheme is the implicit Euler method that allows larger time steps and is widely used in different methods such as mass-spring system[4], Finite Element Method(FEM)[49] and Material Point Method(MPM)[21]. For conservative internal forces, the equivalent energy optimization formulation can be derived by the implicit Euler method[34], and then iteratively solved by nonlinear optimization methods. A few changes can be introduced to this process to construct linear systems which are easier to solve[29, 7, 36]. However, it is not straightforward to apply implicit Euler method to BDEM in [30] because their torques can not be expressed as negative gradients of energy. They calculated the twist torque from the difference in orientation between bonded particles, resulting in an asymmetric torque derivative matrix in orientation level. Therefore, some more complex iterative techniques must be employed, such as GMRES, BiCGSTAB. In addition, even though we construct a linear system in angular velocity level following Baraff et al.[4], it will still lead to the asymmetry of the coefficient matrix.

In this paper, we present a novel implicit-friendly BDEM by redesigning the bond model between discrete elements with the Cosserat rod, which yields a symmetric energy Hessian matrix. We derive an energy optimization formulation for the implicit BDEM where the orientation of discrete elements are represented in quaternion forms. Inspired by the one-dimensional continuum Cosserat rod discretization model, we derive a new Cosserat rod discretization representation, which can efficiently handle the three-dimensional topological connection between discrete elements. Large time-step simulations are then achieved by jointly solving the optimal position $\mathbf{x}$ and orientation $\mathbf{q}$ of the particles, which can be two orders of magnitude larger than those in traditional explicit BDEM, achieving up to 8 times speed up of the total simulation. We quantitatively verify through experiments that our proposed method can produce consistent results with previous work. A wide range of phenomena such as elastic deformation, breaking, cracking, and impacting can be stably and efficiently simulated by our approach.

Our main contributions are:

- An explicit BDEM solver with the Cosserat energy that produces analytically integrable elastic forces and torques, and symmetric torque derivatives.
- A new Cosserat rod discretization method can efficiently represent the three-dimensional topological connection between particles.
- A complete implicit BDEM system is constructed integrating appropriate fracture model and contact model. The solver can simulate faster at high Young's modulus than previous explicit methods.

*Corresponding author
ORCID(s):

2. Related Works

2.1. Discrete Element Method

Discrete Element Method(DEM) was proposed by [14] to simulate rock and granular material. It has been widely used in various fields in the past decades, such as granular material mechanics, rock and soil mechanics, mining engineering, food industry, etc. DEM also inspired many works in graphics to model fluid-solid coupling[46, 17, 60], granular materials[5, 65] and solid material[30]. DEM does not make a continuum assumption, and each of its elements is a quasi-rigid body. Considering this natural discontinuity, BDEM was developed by Potyondy and Cundall [43] to capture the fracture of solids. They connected discrete elements using parallel bond model, allowing the simulation of deformable objects. Some improvements were introduced into BDEM, such as the use of beam models. It connects the centers of two discrete elements using a homogeneous, cylindrical beam with no volume or mass. Some classical beam models are used, such as: Euler-Bernoulli beam[9, 1] and Timoshenko beam[8]. The rod model we adopt can be considered as a nonlinear generalisation of the Timoshenko beam[26]. Lu et al.[30] However, to the best of our knowledge, there is no work focused on the implicit integration of BDEM. In DEM, almost all existing implicit methods only consider implicit update of the linear velocity and not the angular velocity[48, 57], which limits the time step size. Very recently, some implicit integral based DEM simulators considering angular velocity have been proposed. IKari et al.[22] used Taylor expansion to obtain a linear implicit DEM. Klerk et al.[15] used discrete Lagrange-d'Alembert principle to solve an optimization problem of the DEM.

2.2. Rod simulation

Rod simulation has a long history in graphics, mainly including Kirchhoff rods and Cosserat rods. The Kirchhoff rod can simulate the bending and twisting effects between two Lagrangian particles with a smaller DOF after discretization, and is widely used in slender rod and hair simulation[6, 17]. In contrast, the Cosserat rod can additionally simulate shearing effect, making it perform well on short bonds. The Cosserat rod was first introduced to the graphics community by Pai[40] for virtual surgery. Spillman et al. used quaternions to represent orientation in [54], discretize the extensible Cosserat rod and solving it through FEM in explicit integration. Some fast simulation methods developed by the graphics community were applied to the simulation of rods, by introducing the quaternion-based orientation in the corresponding methods, such as Position Based Dynamics(PBD)[24] and Projective Dynamics(PD)[53]. They calculate the corresponding constraints or energies according to [25, 26]. Some methods of discretizing the Cosserat rod also use the implicit scheme. Wen et al.[62] used rh-Adaptive Discretization for sharp contacts. Zhao et al.[66] used the orientation of axis-angle representation to solve the problem of quaternion representation with redundant degrees of freedom. They further transformed the Cosserat rod into an inextensible Kirchhoff rod using a compact representation that satisfies inextensible and unshearble constraints. However, in this paper, we employ discrete elements rather than rod elements to model volumetric solids. The Cosserat rods are utilized solely as virtual bonds to ensure correct elastic response between the bonded particles.

2.3. Fracture simulation

Solid deformation and fracture have been studied for many years since the work of Terzopoulos and Fleischer[55]. There are a large number of methods focus on fracture simulation with good results in graphics. According to the discretization method, there are mainly mesh-based methods and particle-based methods, including FEM[39, 38, 42, 32], Extended FEM(XFEM)[23, 12], Boundary Element Method(BEM)[18, 19, 67], peridynamics[50, 51, 27, 20, 10, 31], MPM[64, 59, 63, 16] and Moving Least Squares(MLS)[41, 33]. However, particle-based methods, including BDEM, are attracting increasing attention because they work well in simulating large topological changes and debris generated by the impact. Here we mainly discuss particle-based methods.

In the BDEM mentioned before, Lu et al.[30] successfully simulated the brittle fracture of elastic body, thin-shell structures, anisotropic material and textile. Peridynamics constructs its kinetic equations through the integral formula. Brittle[27, 10] and ductile fracture[10] have been well simulated through peridynamics in animation. Peridynamics can be divided into bond-based[50, 51] and state-based[52] peridynamics. The force between neighboring two particles in state-based peridynamics is determined by all the particles in their neighborhood. In bond-based peridynamics, the force is determined only by the two bonded-particles. BDEM has a close relationship with bond-based peridynamics. Specifically, BDEM takes particle rotation into account so that bending and twisting can be modeled by bonds between adjacent particles. Therefore, BDEM requires significantly fewer bonds than peridynamics and can directly model slender objects like ropes. Apart from that, BDEM has better scaling consistency than bond-based peridynamics according to Lu et al.[30].

Oriented particles was proposed by another related work[35], which used particles with orientation to efficiently simulate the deformation of solids. However, it is a shape matching method that lacks physical meaning, and the orientation particles are coupled with shape matching instead of orientation constraints.

As a hybrid method, MPM is also effective in simulating fracture in a range of materials. Regardless of numerical fracture, the key concept of MPM fracture simulation is to establish a phase-field equation for dynamic material damage evolution and discretize it using MPM. However, MPM fracture simulation have the problem that the fracture surface is not sharp[64]. Some methods have been proposed to solve it, including surface reconstruction while calculating stress[16] or using Lagrangian force model[59, 63].

3. Rod-Bonded Discrete Element Method

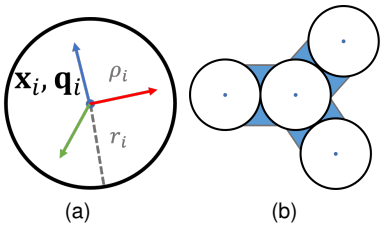


Figure 1: (a) shows the discrete element representation, where the discrete element is a sphere described by $\mathbf{x}_i$, orientation $\mathbf{q}_i$, radius r_i and density ρ_i. (b) shows the topological connection between discrete elements, where one particle can be connected to multiple particles by blue bonds.

Discrete element in 3D space can generally be considered as a small sphere i with six Degrees of Freedom (DOF) of position and orientation. BDEM describes a solid material as a collection of discrete elements connected by bonds. As shown in Fig. 1b, one element can connect multiple elements through bonds. These virtual bonds persist after being initialized between two adjacent discrete elements. There are two forms of interaction between discrete elements, either connected by a bond, or in contact without a bond. These two interactions do not occur at the same time to ensure correct elastic and granular properties in different parts of the material. In general, BDEM will calculate forces and torques based on the state of the connected particles, relative translation and rotation[30, 43], and then update the particle's position and orientation according to the Newton-Euler equation.

However, as mentioned before, the twist torque in [30] produces an asymmetric torque derivative matrix. On the contrary, we hope to find a local constitutive model that yields a symmetric Hessian matrix while accurately simulating the complex deformations caused by stretching, shearing, bending and twisting effects between bonded particles. The Geometrically exact Cosserat Rod[25, 26] can meet the above requirements, so we replace the bond in the BDEM local constitutive model with the Cosserat rod and finally propose the Rod-Bonded Discrete Element Method(RBDEM).

In the following subsections, we first briefly recap the governing equations in Sec. 3.1, then we derive an optimization formulation including inertia and potential from the semi-implicit Euler integration scheme in sec. 3.2. Inspired by the one-dimensional continuum Cosserat rod model[26], we derive a novel Cosserat rod discretization method that efficiently handles 3D topological connections, which is described in Sec. 3.3. Fracture and contact models are then described in Sec. 3.4 and Sec. 3.5, respectively.

3.1. Governing Equations

DEM simulates the behavior of a large number of discrete elements as they interact with each other and environment. For an individual discrete element i in DEM, its governing equation is relatively simple, which is the Newton-Euler equation of rigid body motion:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i \tag{1}$$

$$\frac{d\mathbf{v}i}{dt} = \frac{\mathbf{F}i}{m_i} \tag{2}$$

$$\frac{d\mathbf{q}_i}{dt} = \frac{1}{2}\begin{bmatrix}0\\ \boldsymbol{\omega}_i\end{bmatrix}\mathbf{q}_i \tag{3}$$

$$\frac{d\boldsymbol{\omega}_i}{dt} = \mathbf{I}_i^{-1}\mathbf{T}_i. \tag{4}$$

where $\mathbf{x}_i$ and $\mathbf{q}_i$ denote the position and the orientation represented by the quaternion, $\mathbf{F}_i$ and $\mathbf{T}_i$ are the resultant force and torque of the discrete element i, m_i and $\mathbf{I}_i$ represents mass and moment of inertia, and $\mathbf{v}_i$ and $\boldsymbol{\omega}_i$ are linear velocity and angular velocity. Note that $\boldsymbol{\omega}_i$ and $\mathbf{T}_i$ are all in world frame. Since the discrete elements here are all spheres, their moment of inertia is an exchangeable diagonal matrix and does not change with time,

$$\mathbf{I}_i = \begin{bmatrix} I_1 & & \\ & I_2 & \\ & & I_3 \end{bmatrix}, \tag{5}$$

where $I_1 = I_2 = I_3 = \frac{2}{5}mr^2$. For other non-spherical shapes, the cross product term $\mathbf{I}_i^{-1}((\mathbf{I}_i\boldsymbol{\omega}_i) \times \boldsymbol{\omega}_i)$ needs to be added to the right-hand side of Eqn. (4) like [3]. In this work, we adopt a same fixed radius r for every particle. The multiplication of two quaternions in Eqn. (3) is performed using the default quaternion multiplication law.

3.2. Semi-Implicit Euler Method

We begin with Eqns. (1, 2) for the translational motions. An implicit Euler intergration scheme for the discrete linear velocity update is as follow:

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} + h\mathbf{v}_i^{(n+1)}, \tag{6}$$

$$\mathbf{v}_i^{(n+1)} = \mathbf{v}_i^{(n)} + h\mathbf{M}_i^{-1}\mathbf{F}_i^{(n+1)}, \tag{7}$$

where $\mathbf{M}_i$ is the diagonal mass matrix, and h is the constant simulation time step.

This system is equivalent to an optimization problem as discussed by Martin et al.[34]:

$$\min_{\mathbf{x}_i^{(n+1)}} \frac{1}{2h^2}\sum_i \left\|\mathbf{x}_i^{(n+1)} - \mathbf{s}_i^{(n)}\right\|_{\mathbf{M}_i} + E^{(n+1)}, \tag{8}$$

where $\mathbf{s}_i^{(n)} = \mathbf{x}_i^{(n)} + h\mathbf{v}_i^{(n)}$. The first term of Eqn. (8) is the linear inertial potential, which attracts $\mathbf{x}_i^{(n+1)}$ to $\mathbf{s}_i^{(n)}$. The second term is the elastic potential with respect to $\mathbf{x}$.

To obtain a similar energy optimization function that is consistent with the form of Eqn. (8) for rotation updates, we begin with following semi-implicit Euler integration scheme for angular velocity of the particle i:

$$\mathbf{q}_i^{(n+1)} = \mathbf{q}_i^{(n)} + \frac{h}{2}\begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n+1)}\end{bmatrix}\mathbf{q}_i^{(n)}, \tag{9}$$

$$\boldsymbol{\omega}_i^{(n+1)} = \boldsymbol{\omega}_i^{(n)} + h\mathbf{I}_i^{-1}\mathbf{T}_i^{(n+1)}. \tag{10}$$

We prove in the supplemental document that this system is equivalent to an optimization problem as follows:

$$\min_{\mathbf{q}^{(n+1)}} \frac{2}{h^2}\sum_i \left\|\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)}\right\|_{\widetilde{\mathbf{I}}_i} + E^{(n+1)}, \tag{11}$$

where $\bar{\mathbf{q}}_i^{(n)}$ and $\mathbf{u}_i^{(n)}$ only depend on previously computed states, $\widetilde{\mathbf{I}}_i$ is defined as $\widetilde{\mathbf{I}}_i = \begin{bmatrix} 0 & \\ & \mathbf{I}_i \end{bmatrix}$. $\mathbf{u}_i$ is defined as $\mathbf{u}_i^{(n)} = \mathbf{q}_I + \frac{h}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_i^{(n)} \end{bmatrix}$, where $\mathbf{q}_I$ is an identity quaternion. The second term is the elastic potential, which is the sum of all potentials associated with the orientation constraints. Using Eqn. (9), the first term can become as follows, $\frac{1}{2}(\boldsymbol{\omega}_i^{n+1} - \boldsymbol{\omega}_i^n)^T\mathbf{I}_i(\boldsymbol{\omega}_i^{n+1} - \boldsymbol{\omega}_i^n)$, which is the rotational inertial potential.

Since the two optimization problems of Eqn. (8,11) find stationary points about $\mathbf{x}$ and $\mathbf{q}$ respectively, we can combine the two equations to solve them jointly,

$$\begin{aligned} E^{Total} = &\min_{\mathbf{x}^{(n+1)},\mathbf{q}^{(n+1)}} \frac{1}{2h^2}\sum_i \left\|\mathbf{x}_i^{(n+1)} - \mathbf{s}_i^{(n)}\right\|_{\mathbf{M}_i} \\ &+ \frac{2}{h^2}\sum_i \left\|\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)}\right\|_{\widetilde{\mathbf{I}}_i} + E^{(n+1)}. \end{aligned} \tag{12}$$

Intuitively, Eqn. (12) describes a balance between linear inertial potential, rotational inertial potential, and elastic potentials. Its Hessian matrix is the second order derivative of the energy, which must be symmetric. The elastic potential $E^{(n+1)}$ on the right-hand side of Eqn. (12) contains the energy of our newly designed bond and the energy generated by some constraints, including: attachment constraints, boundary constraints, and unit quaternion constraints. For the convenience of description, we also include the contact potential and the gravitational potential into E. For simplicity and speed, we use a linear implicit scheme to solve Eqn. (12). We also use the Gauss-Newton method to approximate the Hessian of the Cosserat energy. We will discuss the numerical solution of Eqn. (12) in Sec.4.2. As we discuss in the supplemental document, the torques in [30] is not integrable, since their torque derivative matrix in orientation level is asymmetric. So the implicit integral of [30] could not be converted into the optimization formulation like Eqn. (12).

3.3. Rod As a Bond

As mentioned above, BDEM in [30] results in an asymmetric Hessian matrix. The Cosserat rod can not only lead to a symmetrical forces and torques derivative matrix, but also model various deformations between bonded discrete elements. In order to make the discretized model efficiently handle the 3D topological connections between discrete elements, we derive a new discretization method of the Cosserat rod.

Cosserat theory models rods with arc length parametrized smooth ***centerline***, $\mathbf{r}(s) : [0, l] \to \mathbb{R}^3$. Every point on the centerline has an orthonormal basis $\{\mathbf{d}_1(s), \mathbf{d}_2(s), \mathbf{d}_3(s)\}$ as shown in Fig. 2, which is also called directors. $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ represent the orthonormal basis in the fixed material frame. Here the unit quaternion $\mathbf{q}(s)$ is commonly used to represent rotation. The directors could be represented by the rotation of the basis vectors as follows $\begin{bmatrix} 0 \\ \mathbf{d}_k \end{bmatrix} = \mathbf{q}\begin{bmatrix} 0 \\ \mathbf{e}_k \end{bmatrix}\bar{\mathbf{q}}$, where $\bar{\mathbf{q}}$ indicates the conjugate of $\mathbf{q}$. The unit quaternion $\mathbf{q}(s)$ varies smoothly with s in rotation space $SO(3)$ and so do the directors. The cross section of the rod is circular and we assume no deformation of the cross section.

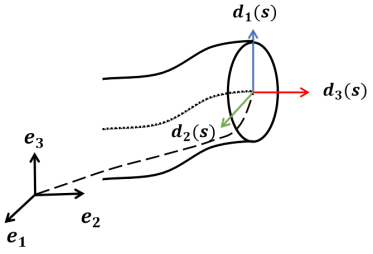


Figure 2: Diagram of the directors in Cosserat rod along the centerline.

It is to be noted that the Cosserat rod can be viewed as a generalization of the Kirchhoff rod. For the Kirchhoff rod, $\mathbf{d}_3(s)$ is strictly along the tangential direction of centerline, i.e. $\mathbf{d}_3(s) = \partial_s\mathbf{r}$, while $\mathbf{d}_1(s), \mathbf{d}_2(s)$ are along the normal plane of centerline at s. This so-called unshearable assumption is also used in beam theory such as Euler-Bernoulli beam. For the Cosserat rod, there is no such assumption. The movement of $\mathbf{d}_3(s)$ is controlled by the "soft constraint" and is limited near the tangent of the centerline, the corresponding normal plane is no longer always perpendicular to the centerline. The benefit of using the Cosserat rod is that shear deformation is taken into account by this soft constraint, allowing us to correctly simulate deformation of bonds, especially short bonds in BDEM[11].

In the following subsections, we will demonstrate how we efficiently extend the above Cosserat model into a 3D topological discretization for BDEM. We recap the elastic potentials from [26] in sec.3.3.1, and then introduce our enhanced discretization for handling 3D topological connections in sec.3.3.2.

3.3.1. Continuous Potential

Stretch and shear potential in Cosserat rods is defined using linear elastic constitutive law[26]

$$E^{SE} = \frac{1}{2}\int_0^l \boldsymbol{\Gamma}^T C^{\Gamma}\boldsymbol{\Gamma} ds, \tag{13}$$

where C^{Γ} is a matrix we will introduce in Eqn.(17) later, $\boldsymbol{\Gamma}$ is the strain measure, defined in material frame as

$$\boldsymbol{\Gamma} = R(\mathbf{q})^T\partial_s\mathbf{r} - \mathbf{e}_3. \tag{14}$$

$\partial_s\mathbf{r}$ is the extensible tangent vector on the centerline. We have $\left\|\partial_s\mathbf{r}^0\right\| = 1$ for any s in the initial configuration.

$R(\mathbf{q})^T\partial_s\mathbf{r}$ means transform $\partial_s\mathbf{r}$ from the world frame to the material frame. This strain measures the deviation of the curve tangent vector from director $\mathbf{e}_3$, including angle deviation(shearing) and length deviation(stretching or compressing). We have $\mathbf{\Gamma} = 0$ in the initial configuration, i.e. $R(\mathbf{q}^0)^T\partial_s\mathbf{r}^0 = \mathbf{e}_3$, which means that at the initial moment, the centerline is perpendicular to the cross section for any s.

Bend and twist potential is defined in a similar form:

$$E^{BT} = \frac{1}{2}\int_0^l \mathbf{\Omega}^T C^{\Omega}\mathbf{\Omega} ds, \tag{15}$$

where $\mathbf{\Omega}$ is the material curvature vector, also called the Darboux vector. The Darboux vector describes the amount by which the basis changes when the curve parameter s changes. A similar definition is angular velocity $\boldsymbol{\omega}$, which describes the amount by which the basis changes when the time changes. It can be defined in material frame as:

$$\begin{bmatrix} 0 \\ \mathbf{\Omega} \end{bmatrix} = 2\bar{\mathbf{q}}\partial_s\mathbf{q}, \tag{16}$$

which is formally consistent with the angular velocity[24].

For a cylindrical bond, C^{Γ} and C^{Ω} are respectively the following diagonal matrices[26],

$$C^{\Gamma} = \begin{bmatrix} \mathcal{G}A_1 & & \\ & \mathcal{G}A_2 & \\ & & \mathcal{E}S \end{bmatrix} \text{ and } C^{\Omega} = \begin{bmatrix} \mathcal{E}I_1^r & & \\ & \mathcal{E}I_2^r & \\ & & \mathcal{G}J^r \end{bmatrix}, \tag{17}$$

where $S = \pi r^2$, $I_1^r = I_2^r = \frac{\pi r^4}{4}$, $J^r = \frac{\pi r^4}{2}$ are the area, moments of inertia and polar moment of the rod cross section, and $A_1 = A_2 = \kappa S$ are the effective cross section areas, where κ is the Timoshenko shear correction factor depends on the geometry of the beam and its Poisson's ratio[13]. For simplicity, we directly use the approximation $\kappa = 5/6$ for circular cross section, following the convention of [61]. Here the cross section is symmetric, so we set $I^r = I_1^r = I_2^r$ and $A = A_1 = A_2$. $\mathcal{E}$ and $\mathcal{G}$ are the Young's modulus and shear modulus of the rod material, respectively.

3.3.2. Discretization

Take a small segment of rod as a rod element. The classic Cosserat rod usually uses the staggered grid approach to discretize the rod[54, 58, 24, 53, 66]. It would discretize the orientation to the middle of rod element and discretize the position to the ends of the rod element. Our discretization method is different. Each rod element is defined b y two points $\{\mathbf{x}_i', \mathbf{x}_j'\}$ and two quaternions $\{\mathbf{q}_i', \mathbf{q}_j'\}$, which represent the position and orientation of both ends of the rod element. We rigidly connect the two ends of the rod to the discrete elements as shown in Fig. 3, so that the rod element can be totally represented by two connected discrete elements. We will directly use $\{\mathbf{x}_i, \mathbf{x}_j\}$ and $\{\mathbf{q}_i, \mathbf{q}_j\}$ to represent rod element below. Besides, the radius of the bond cross section is set with $min(r_i, r_j)$.

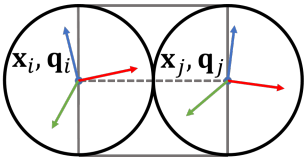


Figure 3: A Cosserat rod element is rigidly connected to discrete elements. $\mathbf{x}$, $\mathbf{q}$ at both ends of the rod element are controlled by the corresponding discrete elements.

There are two reasons for choosing this form of discretization. On the one hand, the new discretization method naturally fits in BDEM where both ends of the bond need to be controlled. On the other hand, note that we have to compute the Darboux vector $\mathbf{\Omega}$ and bend and twist potential E^{BT} between the two rod elements using the previous staggered approach. This is no problem for 1D rod simulations, but for 3D solid simulations, where a particle is connected to multiple particles and has multiple bonds like in Fig. 1b, this will heavily reduce efficiency. For example, when a particle is connected to n particles ($n > 1$), the staggered grid requires us to calculate $\frac{1}{2}n * (n-1)$ number of $\mathbf{\Omega}$ and E^{BT}, and also store the corresponding number of initial Darboux vectors $\mathbf{\Omega^0}$. On the contrary, the new discretization allows us to calculate bending and twisting on a single bond, only needing to calculate n number of $\mathbf{\Omega}$ and E^{BT}, avoiding redundant calculation time.

Using the finite difference method, since the position and orientation are at both ends of the rod element, the resulting discrete strain, curvature, and energy are all at the midpoint of the rod element. The discretization of Eqn. (13) is defined as:

$$E_{ij}^{SE} = \frac{l}{2}\mathbf{\Gamma}_{ij}^T C^{\Gamma}\mathbf{\Gamma}_{ij}, \tag{18}$$

where l is the initial length of the rod element. Discrete strain measure is defined as follows:

$$\mathbf{\Gamma}_{ij} = \frac{1}{l}(\mathbf{R}(\mathbf{q}_{ij})\mathbf{R}(\mathbf{q}_{ij}^0))^T(\mathbf{x}_j - \mathbf{x}_i) - \mathbf{e}_3, \tag{19}$$

where $\mathbf{q}_{ij}^0$ is initial element orientation satisfies initial unstrained state,

$$\frac{1}{l}\mathbf{R}(\mathbf{q}_{ij}^0)^T(\mathbf{x}_j^0 - \mathbf{x}_i^0) = \mathbf{e}_3, \tag{20}$$

$\mathbf{q}_{ij}$ is defined at the midpoint of the rod element. In contrast to the widely used linear interpolation for quaternions[54, 24, 53, 66], we use normalized linear interpolation to ensure that $\mathbf{q}_{ij}$ is a unit quaternion[45],

$$\mathbf{q}_m = \frac{1}{2}(\mathbf{q}_i + \mathbf{q}_j) \text{ and } \mathbf{q}_{ij} = Normalize(\mathbf{q}_m), \tag{21}$$

where $Normalize(\cdot)$ represents normalization operation.

E_{ij}^{BT} and $\mathbf{\Omega}_{ij}$ are also situated on the midpoint. The discretization of Eqn. (15) is defined as:

$$E_{ij}^{BT} = \frac{l}{2}\mathbf{\Omega}_{ij}^T C^{\Omega}\mathbf{\Omega}_{ij}. \tag{22}$$

Discrete Darboux vector is defined as follow:

$$\mathbf{\Omega}_{ij} = \frac{2}{l}\mathfrak{I}(\bar{\mathbf{q}}_{ij}(\mathbf{q}_j - \mathbf{q}_i)), \tag{23}$$

where $\mathfrak{I}(\cdot)$ denotes the imaginary part of the quaternion product.

3.4. Fracture Model

We use the stress-based fracture model in BDEM[43]. This fracture model does not consider the stress state inside the solid, but calculates the maximum normal stress and shear stress on the cross section of the bonds. According to the distribution of stresses on the cross section of the bonds, they all reach the maximum value at the outermost layer of the bond[8],

$$\sigma = -\frac{F^n}{A} + \frac{|\mathbf{T}^b|r}{I^r}, \tag{24}$$

$$\tau = \frac{4|\mathbf{F}^s|}{3S} + \frac{|\mathbf{T}^t|r}{J^r}. \tag{25}$$

The bond will fail if the maximum normal stress and shear stress exceed the corresponding strength, i.e. $\sigma > \sigma_c$, or $\tau > \tau_c$. The directions of the normal force $\mathbf{F}^n$, the shear force $\mathbf{F}^s$, the bend torque $\mathbf{T}^b$ and the twist torque $\mathbf{T}^t$ are shown in Fig. 4, and their calculation will be explained later.

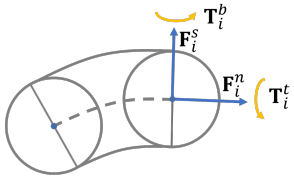


Figure 4: Forces and torques are decomposed into directions along the bond and directions vertical to the bond.

The resultant force and torque are obtained by calculating the negative gradient of the Cosserat potentials by

$$\mathbf{F}_i^{SE} = -\nabla_{\mathbf{x}_i} E_{ij}^{SE} = -l(\frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i})^T C^{\Gamma} \mathbf{\Gamma}_{ij}, \tag{26}$$

$$\widetilde{\mathbf{T}}_i^{BT} = -\nabla_{\mathbf{q}_i} E_{ij}^{BT} = -l(\frac{\partial \mathbf{\Omega}_{ij}}{\partial \mathbf{q}_i})^T C^{\Omega} \mathbf{\Omega}_{ij}. \tag{27}$$

The derivatives of the strain measure Γ and the Darboux vector Ω are given in the supplemental document. According to [47], torque parameter $\widetilde{\mathbf{T}}_i \in \mathbb{R}^4$ could be transformed into Euclidean torque $\mathbf{T}_i \in \mathbb{R}^3$:

$$\begin{bmatrix} 0 \\ \mathbf{T}_i \end{bmatrix} = \frac{1}{2}\widetilde{\mathbf{T}}_i \bar{\mathbf{q}}_i, \tag{28}$$

where we use $\widetilde{(\cdot)}$ to represent the 4d vector dual to the 3d vector.

Now, let us see how to decompose force and torque in Eqns. (26, 27). Since Γ_{ij} and Ω_{ij} are all in material frame, which means it already contains the normal and tangential decomposition in world frame. The third components of Γ_{ij} and Ω_{ij} are along the bond, and the remaining two components are vertical to the bond. When we calculate the force or torque in a certain direction, we could simply set the stiffness in the vertical direction of C^{Γ} or C^{Ω} to 0, and get $C^{\Gamma,n}, C^{\Gamma,s}, C^{\Omega,n}, C^{\Omega,s}$ as follow:

$$\begin{bmatrix} 0 & & \\ & 0 & \\ & & \mathcal{E}S \end{bmatrix}, \begin{bmatrix} \mathcal{G}A & & \\ & \mathcal{G}A & \\ & & 0 \end{bmatrix}, \begin{bmatrix} 0 & & \\ & 0 & \\ & & \mathcal{G}J^r \end{bmatrix}, \begin{bmatrix} \mathcal{E}I^r & & \\ & \mathcal{E}I^r & \\ & & 0 \end{bmatrix}$$

Substituting these stiffness matrices into Eqns. (26, 27), the following force and torque decompositions are obtained directly:

$$\begin{aligned} &\mathbf{F}_i^n = -C^{\Gamma,n}\mathbf{\Gamma}_{ij}, \mathbf{F}_i^s = -l(\frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i})^T C^{\Gamma,s}\mathbf{\Gamma}_{ij}, \\ &\widetilde{\mathbf{T}}_i^t = -l(\frac{\partial \mathbf{\Omega}_{ij}}{\partial \mathbf{q}_i})^T C^{\Omega,t}\mathbf{\Omega}_{ij}, \widetilde{\mathbf{T}}_i^b = -l(\frac{\partial \mathbf{\Omega}_{ij}}{\partial \mathbf{q}_i})^T C^{\Omega,b}\mathbf{\Omega}_{ij}. \end{aligned} \tag{29}$$

$\mathbf{F}^n$ is calculated in the material frame ($l\frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i}^T$ could be regarded as a transformation from the material frame to the world frame). Because F^n in Eqn. (24) we need is a signed scalar, the third component of $\mathbf{F}^n$ in the material frame can be conveniently taken as F_n. We further convert $\widetilde{\mathbf{T}}_i$ to 3D torque $\mathbf{T}_i$ by Eqn. (28). Since the torques on particle i and j are not equal, we use the maximum torque $|\mathbf{T}^b| = max(|\mathbf{T}_i^b|, |\mathbf{T}_j^b|)$, $|\mathbf{T}^t| = max(|\mathbf{T}_i^t|, |\mathbf{T}_j^t|)$ as in [8] to determine whether the bond is broken.

We note that although we adopt [43] for the fracture model by default, our rod-bond framework also has the flexibility to allow for other kinds of fracture models. An example is given in Fig. 13, where we use the fracture model in [2] to simulate the Hertzian cone in glass.

3.5. Contact Model

In BDEM, discrete elements are quasi-rigid spheres without deformation, and they can overlap slightly when in contact. The BDEM solver detects these overlaps between non-bonded particles, and once an overlap is found, it applies a pair of equal and opposite repulsive forces to the two overlapping particles to resolve the contact. As a penalty method, BDEM classically uses penalty forces include Hookean contact force[30, 15], or the Hertzian contact force[8]. We use $\mathbf{d}_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|$ to represent the distance between two particles, and use $\hat{\mathbf{d}}_{ij} = \mathbf{r}_i + \mathbf{r}_j$ to represent the threshold at which two particles just do not penetrate each other. The Hookean contact force is in the following form, $\mathbf{F}_{ij}^c = k_c(\hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij})\mathbf{n}$, where $\mathbf{n} = \left(\mathbf{x}_i - \mathbf{x}_j\right)/\mathbf{d}_{ij}$. The corresponding contact potential follows $E_{ij}^c \sim k_c(\hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij})^2$. The Hertzian contact force is $\mathbf{F}_{ij}^c = k_c(\hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij})^{1.5}\mathbf{n}$.The corresponding contact potential follows $E_{ij}^c \sim k_c(\hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij})^{2.5}$.

In implementation, these well-proven penalty forces are both integrated under explicit methods. When we use implicit schemes for integration to simulate fractures in

large time step, the results of these forces are not satisfactory. Specifically, if the bond is compressed, or the particles are close together when the bond is initialized, the overlap between the two bonded particles will be large before fracture. Then when k_c is large, the particles will suddenly be subjected to a huge penalty force after fracture. The momentum of this particle is so high that its other bonds are broken, leading to catastrophic results in fracture simulation[56]. When k_c is small, tunneling artifacts may occur between two colliding objects.

Here, We use smooth log-barrier function as penalty potential[28]. The contact potential follows:

$$E_{ij}^{c} = \frac{k_c \pi r}{2}(\hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij})^2 \ln\left(\frac{\mathbf{d}_{ij}}{\hat{\mathbf{d}}_{ij}}\right), \tag{30}$$

where $\frac{\pi r}{2}$ is a coefficient from [30]. Compared with the Hookean and Hertzian contact potential, Eqn. (30) can give a smaller force when the two particles overlap little, and alleviate the problem of crushing caused by excessive contact force when the fracture occurs. At the same time, as the overlap between the two particles increases, the penalty force can grow to infinity to avoid penetration.

4. Implementation

4.1. Initial Configuration

There are many ways to initialize particles in solid packing in BDEM. Using hexagonal closest packing with the same element size in mesh is the most efficient way. This method can ensure that the particles are just in contact with each other at the initial moment without penetration, and has a good scaling consistency[30]. We use this packing method in Figs. 5,7,9. However, this packing produces unexpected anisotropy[1], requires random bond strength to produce more natural cracks, and can not represent smooth surfaces. In other figures, random close packing, which is more common in engineering[43, 1], is used. Random close packing can be generated by dynamically simulating contact forces between particles and between particles and boundaries. Both DEM(Figs. 10,12,13) and SPH(Fig. 11) simulations can generate packing with small particle overlaps , which is sufficient to handle fracture simulations in BDEM.

During the initialization procedure of bonds, each particle is connected to other particles within the range of $\delta_0 r$ through the generated bond. The length l of the bond is the distance between the two particles at this time. We set $\delta_0 = 1.1$ for all cases. We further calculate ${\mathbf{q}^0}_{ij}$ through Eqn. (20) and store it on the bond.

4.2. Simulation Workflow

Gravity, attachment and boundary are also handled as energies. The first derivative of the gravitational potential is gravity, and the second derivative is 0. The attachment constraint is a zero rest-length spring added between the current position and the target position, which could be fixed or moved under our control. The boundary constraint is also a zero rest length spring, but its orientation is always perpendicular to the normal of the boundary. For simplicity, its stiffness is set to be as large as the contact stiffness k_c . An additional requirement is that the quaternion must be unit length. We use the following constraint:

$$E_i^{\mathbf{q}} = \frac{1}{2} k_q (\|\mathbf{q}_i\| - 1)^2, \tag{31}$$

where we set $k_q = 1$ and explicitly normalize $\mathbf{q}_i$ as post-processing after solving the linear system.

We include all the energies in Eqns. (18,22,30,31), attachment, boundary constraint and gravitational potential into our total energy function Eqn. (12) which is now aware of elastic response, contact, attachment and grivity. For simplicity and speed, we approach this optimization problem using a linear implicit scheme (one-step Newton iteration). We approximate the Hessian of the Cosserat energy using the Gauss-Newton method[37], i.e., ignoring the terms containing the second derivatives of the strain Γ and the darboux vector Ω. It can be seen from the last two rows of Table 3 that under this approximation, the macroscopic elastic response of our implicitly-integrated method is basically consistent with that of our explicitly-integrated method. We give the derivative of the Cosserat energy and its approximate Hessian matrix in the supplemental document. Preconditioned conjugate gradient(PCG) with jacobi preconditioner is used to solve one Gauss-Newton iteration. Its convergence threshold is set to 2e-5. The overall algorithm of RBDEM to simulate fracture is in Algorithm 1.

Algorithm 1 Overall Simulation Workflow

function SOLVE($\mathbf{x}_i^{(n)}, \mathbf{q}_i^{(n)}, \mathbf{v}_i^{(n)}, \boldsymbol{\omega}_i^{(n)}$)
  $\mathbf{x}_i^* = \mathbf{x}_i^{(n)} + h\mathbf{v}_i^{(n)}$
  $\mathbf{q}_i^* = Normalize(\mathbf{q}_i^{(n)} + \frac{h}{2}\begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n)}\end{bmatrix}\mathbf{q}_i^{(n)})$
  Compute$\nabla E^{Total}, \nabla^2 E^{Total}$ (Eqn.(12), Sec.4.2)
  $\Delta\mathbf{x}_i, \Delta\mathbf{q}_i = PCGSolve(\nabla E^{Total}, \nabla^2 E^{Total})$
  $\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^* + \Delta\mathbf{x}_i$
  $\mathbf{q}_i^{(n+1)} = Normalize(\mathbf{q}_i^* + \Delta\mathbf{q}_i)$
  $\sigma, \tau = Stress(\mathbf{x}_i^{(n+1)}, \mathbf{q}_i^{(n+1)})$ (Eqn. (24,25))
  check breakage (Sec 3.4)
  $\mathbf{v}_i^{(n+1)} = \frac{1}{h}(\mathbf{x}_i^{(n+1)} - \mathbf{x}_i^{(n)})$
  $\begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n+1)}\end{bmatrix} = \frac{2}{h}(\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{q}_I)$
  Return $\mathbf{x}_i^{(n+1)}, \mathbf{q}_i^{(n+1)}, \mathbf{v}_i^{(n+1)}, \boldsymbol{\omega}_i^{(n+1)}$
end function

We use GPU parallel computing technology to accelerate simulations. Computing interactions and collisions between particles, as well as PCG when implicitly solving Eqn.(12), are performed in parallel. A uniform 3D grid is used to perform spatial neighbourhood particle searches for both explicit and implicit RBDEM.

Experiment	Δt	N	r	ρ	$\mathcal{E}$	$\mathcal{G}$	k_c	σ_c	τ_c	s/frame
Spring	5e-3	2.4K	0.001	2000	7e6	2.8e6	1e6	$+\infty$	$+\infty$	0.50
Cantilever Beam	2e-2	50.6k	0.00125	1000	5e4/5e5	2e4/2e5	1e5	$+\infty$	$+\infty$	0.93/1.16
Beam Bending	2e-4	28.1k	0.00125	3000	1e8	3.3e7	1e7	$1e7$	$7e6$	15.62
Beam Twisting	2e-4	52.5k	0.0025	2600	1e7	3.3e6	1e6	4e6	1.5e6	18.58
Bunny	1e-4	25.8k	0.00125	2700	1e8	4e7	1e7	7.6e5	2e7	21.06
Thin Plate	1e-4	24.2k	0.002	5000	1e9	4e8	5e7	5e6	2e7	17.75
Thick Plate	5e-5	129.5k	0.002	5000	1e8	4e7	5e7	1e6	4e6	14.06
Hertzian Cone	3e-5	62.7k	0.002	5000	1e9	4e7	1e9	-	-	11.30

Table 1
Parameters Setting and performance

Experiment	Δt	max Δt_{exp}	s/frame$_{exp}$	speedup
Spring	5e-3	2e-5	3.76	7.52
Cantilever Beam	2e-2	1e-4/4e-5	3.45/9.41	3.71/8.40
Beam Bending	2e-4	4e-6	30.45	1.95
Bunny	1e-4	4e-6	28.19	1.33
Thin Plate	1e-4	2e-6	58.40	3.29
Thick Plate	1e-4	8e-6	5.90	0.52

Table 2
Performance comparison of our implicit RBDEM with the explicit BDEM.

4.3. Parameter Discussion

The Timoshenko shear correction factor ($\kappa = 5/6$) in Eqn. (17) could correct the effective cross section area, and is used in most of our experiments unless otherwise mentioned. When this factor is not used ($\kappa = 1$), our model produces an elastic response consistent with the explicit BDEM, which will be explained in Sec. 5.2.

5. Results

We implement our experiments with CUDA. Our simulations run on a NVIDIA GEFORCE 1080 Ti GPU. Model parameters and simulation performance are summarized in Table 1. In the cases where only elastic deformation occurs, the frame in the video corresponds to about 0.02s of the simulation. For the cases where fractures occurs, one frame corresponds to about 0.01s of the simulation since these cases involve a lot of fast motion. Especially for the cases of the Hertzian cone(Figs. 12 and 13), because the collision or crack growth occurs extremely fast, one frame corresponds to 0.001s of the simulation.

We also provide a performance comparison of our implicit RBDEM with explicit BDEM[30] using the velocity verlet. We simulate with the largest time step that the explicit model can tolerate. The time step and seconds per frame of the simulation are summarized in Table 2. As shown in Table2, in most cases, Δt_{exp} under explicit integration is limited to very small by Young's modulus. The reason why the implicit version can simulate faster than the explicit version is that it can simulate stably at larger time steps. It is worth noting that a large number of collisions limits the time step of implicit integration, resulting in poor speedup or slower speed than explicit integration like Thick Plate in Table 2. This is due to the inability of the linear implicit scheme to handle high-speed collisions. However, the implicit system generated by the RBDEM model can indeed enlarge the time step of the simulation, and obtain faster results in most scenarios.

5.1. Deformation

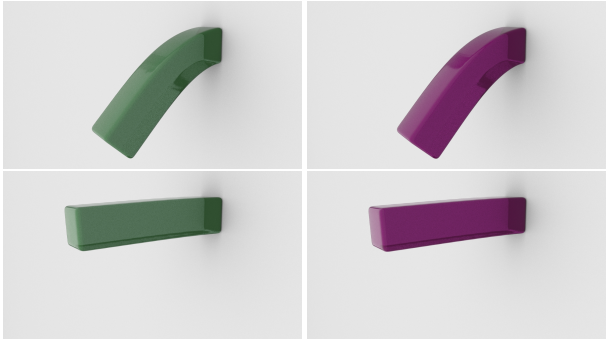
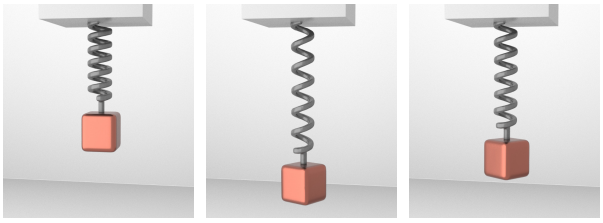


Figure 5: Simulating the cantilever beam using implicit RBDEM(green) and BDEM(purple). Using different Young's modulus: 5e4 (top), and 5e5 (bottom). Explicit BDEM will fail at the same time step, and a time step of table2 size must be set to be stable.

Fig. 5 shows a cantilever beam sagging under gravity. We simulate and produce similar visual effects using implicit RBDEM (green) and BDEM (purple), respectively. Both simulations of $E = 5e4$ (the first row) and $E = 5e5$ (the second row) can be accelerated by implicit RBDEM. As shown in Table 1 and Table 2, the speedup ratio is larger when $E = 5e5$, because the maximum time step that the explicit method can tolerate decreases significantly when the Young's modulus is larger, which makes the explicit method more time-consuming. In this comparison, in order to make the vibration amplitudes during the sag simulated by the two models visually similar, we set $\kappa = 1$ in RBDEM and add a small velocity damping in BDEM because the implicit method automatically introduces numerical damping.

In Fig. 6, the spring oscillates up and down between Fig. 6a and Fig. 6b due to the mutual conversion of elastic potential, kinetic energy and gravitational potential, and finally stops at a middle position Fig. 6c. This case is composed entirely of discrete elements which shows that RBDEM can directly handle the simulation where coupling of rods (springs) and volumetric objects is included.



(a) (b) (c)

Figure 6: A spring dynamometer that oscillates up and down under gravity.

	$\mathcal{E}_{micro}$	σ_{micro}	$\mathcal{E}_{macro}$	σ_{macro}	Δt	s/frame
BDEM	1.0e8	$1.0e7$	$1.11e8$	$1.63e7$	4e-6	30.45
Ours_{exp} ($\kappa = 1$)	1.0e8	$1.0e7$	$1.11e8$	$1.63e7$	4e-6	32.10
Ours_{exp}	1.0e8	$1.0e7$	$1.07e8$	$1.55e7$	4e-6	31.95
Ours_{imp}	1.0e8	$1.0e7$	$1.07e8$	$1.52e7$	2e-4	15.62

Table 3
All cases use the same fracture model and contact model described in Sec.3.4, 3.5. Ours_{exp} mean RBDEM in explicit integration, while Ours_{imp} mean in implicit integration. The other microscopic parameter settings are in Table 1.

5.2. Bending and Twisting

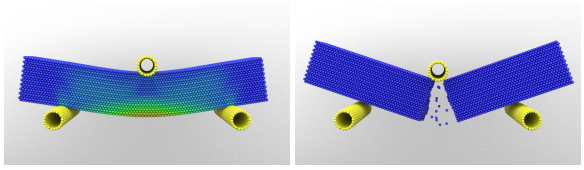


Figure 7: Three points bending test. The particle color is visualized using the tensile stress magnitude in the connected bonds.

The three points bending test, as shown in the Fig. 7, refers to the test to measure the macroscopic Young's modulus and macroscopic flexural strength of a piece of material. As the cylindrical indenter presses down, we can measure the corresponding load-displacement diagram. According to the formula from beam theory in [30], the macroscopic Young's modulus $\mathcal{E}_{macro}$ and flexural strength σ_{macro} of the material can be obtained through the slope of the load-displacement curve and the maximum load before fracture, and the calculation results are shown in Table 3. We use the method of least squares to fit the slopes of the measured points. In the case of explicit integration without adding the Timoshenko shear correction factor ($\kappa = 1$), our model can obtain almost the same $\mathcal{E}_{macro}$ and σ_{macro} compared with BDEM. With the help of the Timoshenko shear correction factor ($\kappa = 5/6$), the measured macroscopic parameters differ from BDEM but are closer to the corresponding microscopic parameters $\mathcal{E}_{micro}$ and σ_{micro}. Finally, we use an implicit integration of the RBDEM with a larger time step, and the comparison shows that our implicit RBDEM produces consistent results at a faster speed. Note that we calculate higher σ_{macro} in the first row of table 3 than [30], because our fracture model does not consider compression, which delays the occurrence of fracture. Fig. 8 shows bending experiments similar to Fig. 7 with a hole or random close packing of different discrete element radii.

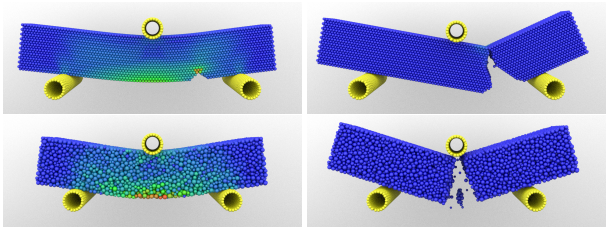


Figure 8: Bending experiments with a pre-set hole or random close packing of different discrete element radii.

The first row of Fig. 9 shows a beam that gradually twists to fracture. The second and third rows show the results of explicit RBDEM and BDEM simulations at the same computation time, respectively. It can be observed that the object shapes and stress distributions during the twisting process of the three simulators are similar. However, for explicit integration, regardless of whether the rod model is used or not, the stress distribution on the fragments is discontinuous, and many unexpected explosive particles are generated. When one bond breaks, the two connected particles will lose the elastic forces exerted by the bond and gain equal and opposite contact forces.The adjacent bonds need to be further deformed to maintain the equilibrium state of the two particles, causing the adjacent bonds that are already close to the threshold (σ_c ,τ_c) to break. For implicit integration, since it involves a global linear system with respect to position, the deformation will be more evenly distributed to the surrounding bonds instead of just adjacent bonds. The explosive particles are still present when we reduce k_c and Δt in explicit integration.

5.3. Impacting

Fig. 10 shows the fracture of a thin plate by the impact of a ball. We provide a comparison with MPM in the second row of Fig. 10 to illustrate that our method can produce better visual effects in brittle fracture. NACC model in CDMPM[64] is used with the same number of particles. We find that it is difficult to produce brittle fracture using CDMPM under feasible parameters when the Young's modulus is large. We search hard for parameters to raise Young's modulus, and finally set it to 5e3 (significantly lower than 1e9), which is sufficient to produce fragments similar to RBDEM. However, as is shown in Fig. 10, the fracture surface simulated by MPM is not sharp, while the fracture surface of RBDEM is rigid and clear. Increasing the number of sampled particles in MPM does not solve this problem, and ultimately may rely on complex reconstruction methods such as in [16].

Fig. 11 shows simulation of a breakable bunny sculpture impact a rigid wall. The bunny breaks after hitting the wall,

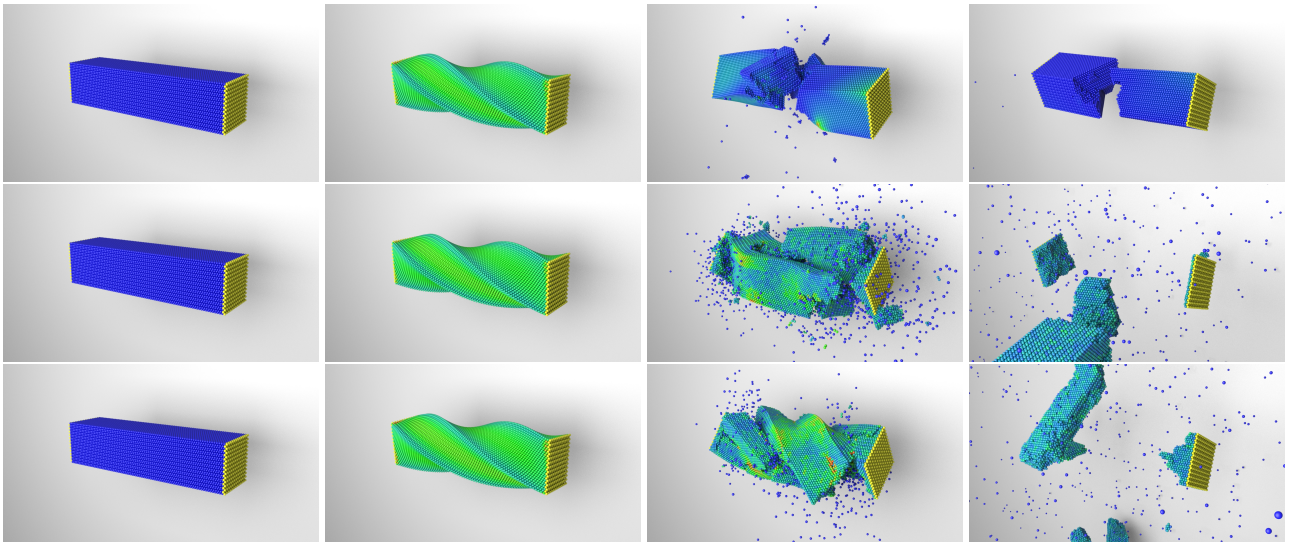


Figure 9: Both ends of the beam are initially fixed, after which they rotate in opposite directions with a fixed angular velocity of 2 rad/s. Using implicit RBDEM (top), explicit RBDEM (middle) and explicit BDEM (bottom), respectively.

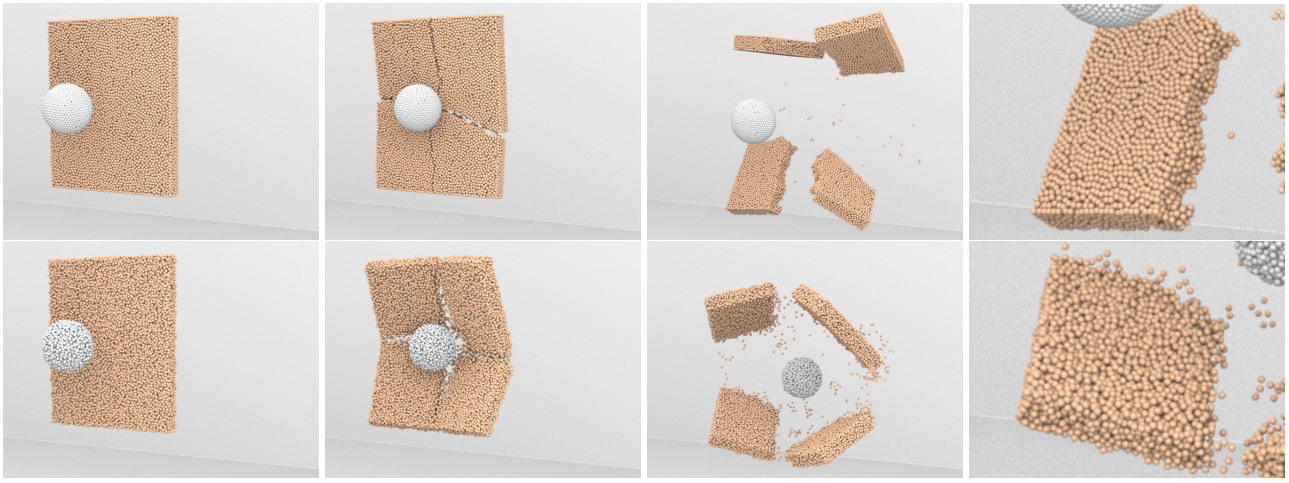


Figure 10: A thin plate produce a brittle fracture effect with implicit RBDEM(top), and MPM(bottom) after being impacted by the ball. The initial speed of the ball is $5m/s$.

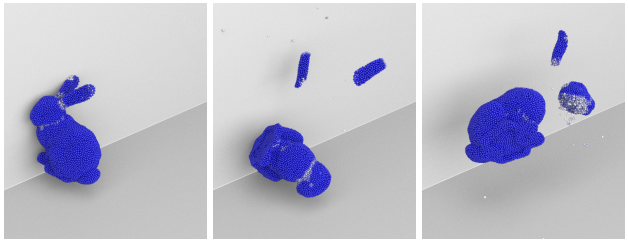


Figure 11: A bunny sculpture breaks after hitting the wall at a speed of 3m/s.

and the rest of it breaks again after contacting the ground. Our method can handle fractures in complex geometries.

5.4. Hertzian Cone

Fig. 12 shows the fracture of a thick plate after being hit by a spherical indenter moving at a constant velocity $10m/s$, and the bottom of the thick plate is fixed. From the cross section of the thick plate shown in the first row of Fig. 12, it can be seen that debris correctly flies out from the impacted side, while the Hertzian Cone crack develops on the other side.

The Hertzian cone crack in the first row of Fig. 13(image taken from [44]) was first observed by Hertz at the end of the 19th century. With the fracture model in [43], it is possible to simulate cone-shaped cracks in geotechnical materials like Fig. 12, but Hertzian cone cracks in silica glass can not yet be simulated. André et al.[2] noted that, with the standard fracture model in BDEM simulation, the fracture would occur near the indenter and propagate through the entire material thickness in the indentation test, which is unexpected. In order to simulate the clear Hertzian cone crack under indenter impact in silica glass, they proposed a new fracture criterion based on hydrostatic stress. Our robust bond model

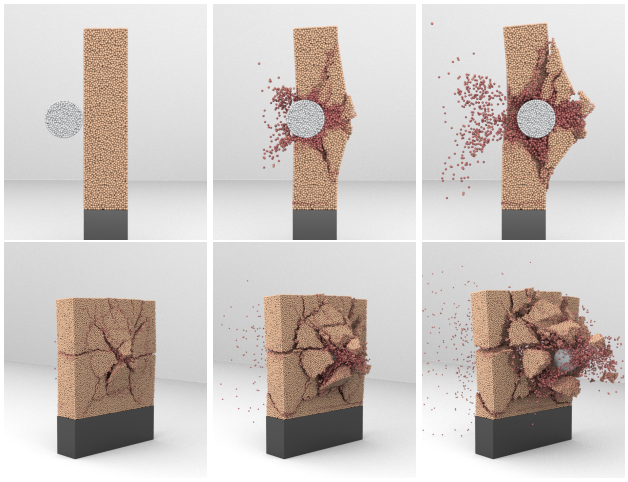


Figure 12: A thick plate is impacted by a spherical indenter, producing complex fragmentation effects with implicit RBDEM. The cross-sectional view and the full view of the simulation results are shown in the first and second rows, respectively.

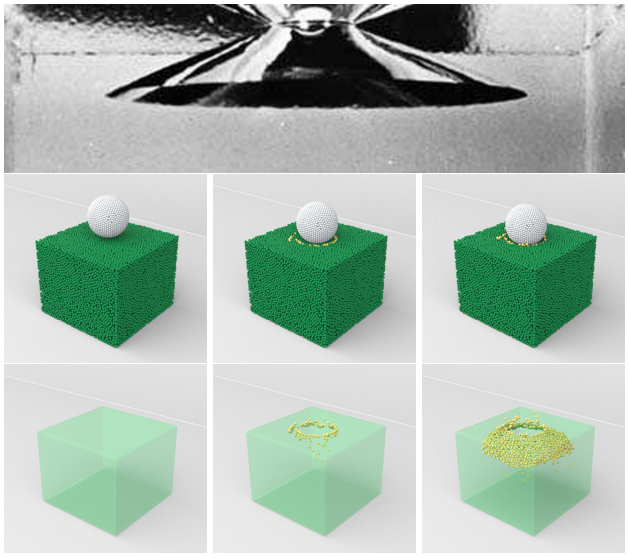


Figure 13: Photograph of the Hertzian cone crack in real world(first row) and computer simulation of the Hertzian cone crack in silica glass(second and third rows). We use transparent surfaces for better crack visualization (third row).

enables simulation of the expected Hertzian cone crack in silica glass under this fracture model.

Firstly, this model defines the Cauchy stress on discrete element i, which needs to be calculated by traversing all the bonds connected to the particle,

$$\bar{\sigma}_i = \frac{1}{4V}\sum_j (\mathbf{x}_{ij} \otimes F_i^{SE} + F_i^{SE} \otimes \mathbf{x}_{ij}), \tag{32}$$

where $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, $\otimes$ is the tensor product and V is the volume of the discrete element. The original formula in [2] uses force from the cohesive beam bond model instead of the F_i^{SE} derived from the Cosserat potential in our RBDEM. Brittle fracture occurs when the tensile hydrostatic stress (volumetric stress) calculated from the Cauchy stress is large. We calculate the tensile hydrostatic stress of each particle i by the formula $\frac{1}{3}trace(\bar{\sigma}_i)$, where $trace(\cdot)$ means to trace the matrix. Once the hydrostatic stress of the particle reaches a threshold, i.e. $\frac{1}{3}trace(\bar{\sigma}_i) > \bar{\sigma}_c$, all bonds connected to the particle are broken. In this case, we set $\bar{\sigma}_c = 4e6$.

The simulation results obtained with this fracture model is shown in Fig. 13, where a piece of silica glass is cracked by the impact of a spherical indenter. We can observe the gradual growth of the apparent cone-shaped crack inside the transparent glass in the third row of Fig. 13, where we do not visualize the indenter to fully show the Hertzian cone crack.

6. Conclusion and Future Works

In this work, we propose a new BDEM. Through the new discretization method of the Cosserat rod, we efficiently integrate the Cosserat rod into BDEM. Since our method is based on Cosserat energy, it is able to generate a symmetric Hessian matrix. By modifying the fracture model and selecting the appropriate contact potential, we construct a complete implicit integral system for BDEM. We quantitatively show that our method is able to produce results consistent with previous work. Since the implicit integration allows us to simulate with larger time steps, the implicit RBDEM is computationally faster than previous work in some scenarios.

The proposed method has some limitations and future work. The time step that explicit BDEM can tolerate increases when Young's modulus gets smaller, so in this cases implicit RBDEM may not have a speed advantage over explicit BDEM. As we point out in the results, the implicit method significantly increase the time step of the simulation,but collisions limit the further increase of the time step. Since we explicitly integrated the fracture model in the implicit simulation, when the time step is too large, the propagation of the fracture may be limited, manifested as slower propagation. We believe that both problems need to be solved by better multi-step implicit iteration involving collisions and fractures, which we consider to be future work. Furthermore, the fully Hessian(rather than Gauss-Newton approximation) and constraint-based methods may provide us better stability under large time step, and exact enforcement of unit quaternion constraints. Since our implicit method is a kind of energy optimization formulation, PBD and PD can be used to speed up the solution. Support for frictional contact allows fracture simulation in more scenarios. Support for plastic deformation, ductile fracture, and fluid-soild c oupling w ould a lso b e i nteresting future work. We would like to investigate the generalization of our method to handle non-uniform and non-spherical discrete elements, which is achievable under the theory of our implicit method.

CRediT authorship contribution statement

Kangrui Zhang: Conceptualization, Methodology, Software, Writing – Original Draft. **Han Yan**: Writing – review & editing, Visualization. **Jia-Ming Lu**: Writing – review. **Bo Ren**: Conceptualization, Methodology, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work was supported by the Natural Science Foundation of China (62272245), the Fundamental Research Funds for the Central Universities (Nankai University, No. 63233080).

References

[1] André, D., Iordanoff, I., Charles, J.l., Néauport, J., 2012. Discrete element method to simulate continuous material by using the cohesive beam model. Computer methods in applied mechanics and engineering 213, 113–125.

[2] André, D., Jebahi, M., Iordanoff, I., Charles, J.l., Néauport, J., 2013. Using the discrete element method to simulate brittle fracture in the indentation of a silica glass with a blunt indenter. Computer Methods in Applied Mechanics and Engineering 265, 136–147.

[3] Baraff, D., 1997. An introduction to physically based modeling: rigid body simulation i—unconstrained rigid body dynamics. SIGGRAPH course notes 82.

[4] Baraff, D., 1998. Large steps in cloth simulation. Proc. SIGGRAPH '98 , 43–54.

[5] Bell, N., Yu, Y., Mucha, P.J., 2005. Particle-based simulation of granular materials, in: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 77–86.

[6] Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., Grinspun, E., 2008. Discrete elastic rods, in: ACM SIGGRAPH 2008 papers, pp. 1–12.

[7] Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M., 2014. Projective dynamics: Fusing constraint projections for fast simulation. ACM transactions on graphics (TOG) 33, 1–11.

[8] Brown, N.J., Chen, J.F., Ooi, J.Y., 2014. A bond model for dem simulation of cementitious materials and deformable structures. Granular Matter 16, 299–311.

[9] Carmona, H.A., Wittel, F.K., Kun, F., Herrmann, H.J., 2008. Fragmentation processes in impact of spheres. Physical Review E 77, 051302.

[10] Chen, W., Zhu, F., Zhao, J., Li, S., Wang, G., 2018. Peridynamics-based fracture animation for elastoplastic solids, in: Computer Graphics Forum, Wiley Online Library. pp. 112–124.

[11] Chen, X., Peng, D., Morrissey, J.P., Ooi, J.Y., 2022. A comparative assessment and unification of bond models in dem simulations. Granular Matter 24, 1–20.

[12] Chitalu, F.M., Miao, Q., Subr, K., Komura, T., 2020. Displacement-correlated xfem for simulating brittle fracture, in: Computer Graphics Forum, Wiley Online Library. pp. 569–583.

[13] Cowper, G., 1966. The shear coefficient in timoshenko's beam theory. Journal of applied mechanics 33, 335–340.

[14] Cundall, P.A., Strack, O.D., 1979. A discrete numerical model for granular assemblies. geotechnique 29, 47–65.

[15] De Klerk, D.N., Shire, T., Gao, Z., McBride, A.T., Pearce, C.J., Steinmann, P., 2022. A variational integrator for the discrete element method. Journal of Computational Physics 462, 111253.

[16] Fan, L., Chitalu, F.M., Komura, T., 2022. Simulating brittle fracture with material points. ACM Transactions on Graphics (TOG) 41, 1–20.

[17] Fei, Y., Batty, C., Grinspun, E., Zheng, C., 2019. A multi-scale model for coupling strands with shear-dependent liquid. ACM Transactions on Graphics (TOG) 38, 1–20.

[18] Hahn, D., Wojtan, C., 2015. High-resolution brittle fracture simulation with boundary elements. ACM Transactions on Graphics (TOG) 34, 1–12.

[19] Hahn, D., Wojtan, C., 2016. Fast approximations for boundary element based brittle fracture simulation. ACM Transactions on Graphics (TOG) 35, 1–11.

[20] He, X., Wang, H., Wu, E., 2017. Projective peridynamics for modeling versatile elastoplastic materials. IEEE transactions on visualization and computer graphics 24, 2589–2599.

[21] Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., Jiang, C., 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. ACM Transactions on Graphics (TOG) 37, 1–14.

[22] Ikari, H., Gotoh, H., 2022. Fully implicit discrete element method for granular column collapse. Computational Particle Mechanics , 1–11.

[23] Koschier, D., Bender, J., Thuerey, N., 2017. Robust extended finite elements for complex cutting of deformables. ACM Transactions on Graphics (TOG) 36, 1–13.

[24] Kugelstadt, T., Schömer, E., 2016. Position and orientation based cosserat rods., in: Symposium on Computer Animation, pp. 169–178.

[25] Lang, H., Linn, J., 2009. Lagrangian field theory in space-time for geometrically exact Cosserat rods. volume 3. ITWM, Fraunhofer Inst. Techno-und Wirtschaftsmathematik.

[26] Lang, H., Linn, J., Arnold, M., 2011. Multi-body dynamics simulation of geometrically exact cosserat rods. Multibody System Dynamics 25, 285–312.

[27] Levine, J.A., Bargteil, A.W., Corsi, C., Tessendorf, J., Geist, R., 2014. A peridynamic perspective on spring-mass fracture., in: Symposium on Computer Animation, Citeseer. pp. 47–55.

[28] Li, M., Ferguson, Z., Schneider, T., Langlois, T.R., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M., 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Trans. Graph. 39, 49.

[29] Liu, T., Bargteil, A.W., O'Brien, J.F., Kavan, L., 2013. Fast simulation of mass-spring systems. ACM Transactions on Graphics (TOG) 32, 1–7.

[30] Lu, J.M., Li, C., Cao, G.C., Hu, S.M., 2021. Simulating fractures with bonded discrete element method. IEEE Transactions on Visualization and Computer Graphics .

[31] Madenci, E., Oterkus, E., 2013. Peridynamic theory, in: Peridynamic theory and its applications. Springer, pp. 19–43.

[32] Mandal, A., Chaudhuri, P., Chaudhuri, S., 2021. Remeshing-free graph-based finite element method for ductile and brittle fracture. arXiv preprint arXiv:2103.14870 .

[33] Martin, S., Kaufmann, P., Botsch, M., Grinspun, E., Gross, M., 2010. Unified simulation of elastic rods, shells, and solids. ACM Transactions on Graphics (TOG) 29, 1–10.

[34] Martin, S., Thomaszewski, B., Grinspun, E., Gross, M., 2011. Example-based elastic materials, in: ACM SIGGRAPH 2011 papers, pp. 1–8.

[35] Müller, M., Chentanez, N., 2011. Solid simulation with oriented particles, in: ACM SIGGRAPH 2011 papers, pp. 1–10.

[36] Narain, R., Overby, M., Brown, G.E., 2016. Admm⊇ projective dynamics: fast simulation of general constitutive models., in: Symposium on Computer Animation, p. 2016.

[37] Nocedal, J., Wright, S., 2006. Numerical optimization. edition springer verlag .

[38] O'brien, J.F., Bargteil, A.W., Hodgins, J.K., 2002. Graphical modeling and animation of ductile fracture, in: Proceedings of the 29th

annual conference on Computer graphics and interactive techniques, pp. 291–294.
[39] O'brien, J.F., Hodgins, J.K., 1999. Graphical modeling and animation of brittle fracture, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 137–146.
[40] Pai, D.K., 2002. Strands: Interactive simulation of thin solids using cosserat models, in: Computer graphics forum, Wiley Online Library. pp. 347–352.
[41] Pauly, M., Keiser, R., Adams, B., Dutré, P., Gross, M., Guibas, L.J., 2005. Meshless animation of fracturing solids. ACM Transactions on Graphics (TOG) 24, 957–964.
[42] Pfaff, T., Narain, R., De Joya, J.M., O'Brien, J.F., 2014. Adaptive tearing and cracking of thin sheets. ACM Transactions on Graphics (TOG) 33, 1–9.
[43] Potyondy, D.O., Cundall, P., 2004. A bonded-particle model for rock. International journal of rock mechanics and mining sciences 41, 1329–1364.
[44] Roesler, F., 1956. Brittle fractures near equilibrium. Proceedings of the Physical society. section B 69, 981.
[45] Romero, I., 2004. The interpolation of rotations and its application to finite element models of geometrically exact rods. Computational mechanics 34, 121–133.
[46] Rungjiratananon, W., Szego, Z., Kanamori, Y., Nishita, T., 2008. Real-time animation of sand-water interaction, in: Computer Graphics Forum, Wiley Online Library. pp. 1887–1893.
[47] Schwab, A.L., Meijaard, J.P., 2006. How to draw euler angles and utilize euler parameters, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Citeseer. pp. 259–265.
[48] Shi, G.h., 1992. Discontinuous deformation analysis: a new numerical model for the statics and dynamics of deformable block structures. Engineering computations .
[49] Sifakis, E., Barbic, J., 2012. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction, in: Acm siggraph 2012 courses, pp. 1–50.
[50] Silling, S.A., Askari, E., 2004. Peridynamic modeling of impact damage, in: ASME Pressure Vessels and Piping Conference, pp. 197–205.
[51] Silling, S.A., Askari, E., 2005. A meshfree method based on the peridynamic model of solid mechanics. Computers & structures 83, 1526–1535.
[52] Silling, S.A., Epton, M., Weckner, O., Xu, J., Askari, E., 2007. Peridynamic states and constitutive modeling .
[53] Soler, C., Martin, T., Sorkine-Hornung, O., 2018. Cosserat rods with projective dynamics, in: Computer Graphics Forum, Wiley Online Library. pp. 137–147.
[54] Spillmann, J., Teschner, M., 2007. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 63–72.
[55] Terzopoulos, D., Fleischer, K., 1988. Modeling inelastic deformation: viscolelasticity, plasticity, fracture, in: Proceedings of the 15th annual conference on Computer graphics and interactive techniques, pp. 269–278.
[56] Truong, N., Yuksel, C., Watcharopas, C., Levine, J.A., Kirby, R.M., 2021. Particle merging-and-splitting. IEEE Transactions on Visualization and Computer Graphics 28, 4546–4557.
[57] Tuley, R., Danby, M., Shrimpton, J., Palmer, M., 2010. On the optimal numerical time integration for lagrangian dem within implicit flow solvers. Computers & chemical engineering 34, 886–899.
[58] Umetani, N., Schmidt, R., Stam, J., 2014. Position-based elastic rods, in: ACM SIGGRAPH 2014 Talks, pp. 1–1.
[59] Wang, S., Ding, M., Gast, T.F., Zhu, L., Gagniere, S., Jiang, C., Teran, J.M., 2019. Simulation and visualization of ductile fracture with the material point method. Proceedings of the ACM on Computer Graphics and Interactive Techniques 2, 1–20.
[60] Wang, X., Fujisawa, M., Mikawa, M., 2021. Visual simulation of soil-structure destruction with seepage flows. Proceedings of the ACM on Computer Graphics and Interactive Techniques 4, 1–18.
[61] Weeger, O., Narayanan, B., Dunn, M.L., 2019. Isogeometric shape optimization of nonlinear, curved 3d beams and beam structures. Computer Methods in Applied Mechanics and Engineering 345, 26–51.
[62] Wen, J., Chen, J., Umetani, N., Bao, H., Huang, J., 2020. Cosserat rod with rh-adaptive discretization, in: Computer Graphics Forum, Wiley Online Library. pp. 143–154.
[63] Wolper, J., Chen, Y., Li, M., Fang, Y., Qu, Z., Lu, J., Cheng, M., Jiang, C., 2020. Anisompm: Animating anisotropic damage mechanics: Supplemental document. ACM Trans. Graph 39.
[64] Wolper, J., Fang, Y., Li, M., Lu, J., Gao, M., Jiang, C., 2019. Cd-mpm: continuum damage material point methods for dynamic fracture animation. ACM Transactions on Graphics (TOG) 38, 1–15.
[65] Yue, Y., Smith, B., Chen, P.Y., Chantharayukhonthorn, M., Kamrin, K., Grinspun, E., 2018. Hybrid grains: Adaptive coupling of discrete and continuum simulations of granular media. ACM Transactions on Graphics (TOG) 37, 1–19.
[66] Zhao, C., Lin, J., Wang, T., Bao, H., Huang, J., 2022. Efficient and stable simulation of inextensible cosserat rods by a compact representation .
[67] Zhu, Y., Bridson, R., Greif, C., 2015. Simulating rigid body fracture with surface meshes. ACM Trans. Graph. 34, 150–1.

Rod-Bonded Discrete Element Method

Kangrui Zhang[a], Han Yan[a], Jia-Ming Lu[b] and Bo Ren[a,*]

[a]TMCC, College of Computer Science, Nankai University, China
[b]Tsinghua University, China

1. Derivation of Energy Function in Orientation Level

To get an integral in orientation level, we reorganize the orientation-related part of the Newton-Euler equation as follows,

$$\mathbf{q}_i^{(n+1)} - \mathbf{q}_i^{(n)} - \frac{h}{2}\begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n)}\end{bmatrix}\mathbf{q}_i^{(n)} = \frac{h^2}{2}\begin{bmatrix}0\\ \mathbf{I}_i^{-1}\mathbf{T}_i^{(n+1)}\end{bmatrix}\mathbf{q}_i^{(n)} \quad (1)$$

Multiplying both sides of the equation by $\bar{\mathbf{q}}_i^{(n)}$,

$$\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)} = \frac{h^2}{2}\begin{bmatrix}\mathbf{0} & \\ & \mathbf{I}_i^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{T}_i^{(n+1)}\end{bmatrix}, \quad (2)$$

where $\mathbf{u}_i^{(n)} = \mathbf{q}_I + \frac{h}{2}\begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n)}\end{bmatrix}$.

Substituting $\begin{bmatrix}0\\ \mathbf{T}_i\end{bmatrix} = \frac{1}{2}\widetilde{\mathbf{T}}_i\bar{\mathbf{q}}_i$ into Eqn. (2) and multiplying both sides by $\mathbf{q}_i^{(n)}$, and moving the moment of inertia to the left-hand side, we get

$$\widetilde{\mathbf{I}}_i((\mathbf{q}_i^{(n+1)} - \mathbf{u}_i^{(n)})\bar{\mathbf{q}}_i^{(n)}) = \frac{h^2}{4}\widetilde{\mathbf{T}}_i\bar{\mathbf{q}}_i \quad (3)$$

for every discrete elements.

Torque parameter $\widetilde{\mathbf{T}}_i$ is conservative about E, $\widetilde{\mathbf{T}}_i = -\nabla_{\mathbf{q}_i}E$[4]. Multiplying both sides by $\mathbf{q}_i^{(n)}$, we further have:

$$(\widetilde{\mathbf{I}}_i((\mathbf{q}_i^{(n+1)} - \mathbf{u}_i^{(n)})\bar{\mathbf{q}}_i^{(n)}))\mathbf{q}_i^{(n)} = -\frac{h^2}{4}\nabla_{\mathbf{q}_i}E. \quad (4)$$

An optimization formulation can be obtained by transposing the right-hand side of Eqn. (4) to the left and integrating:

$$\min_{\mathbf{q}_i^{(n+1)}} \frac{2}{h^2}(\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)})\widetilde{\mathbf{I}}_i(\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)}) + E^{(n+1)}. \quad (5)$$

By summing i, the optimization function can be obtained.

$$\min_{\mathbf{q}^{(n+1)}} \frac{2}{h^2}\sum_i \left\|\mathbf{q}_i^{(n+1)}\bar{\mathbf{q}}_i^{(n)} - \mathbf{u}_i^{(n)}\right\|_{\widetilde{\mathbf{I}}_i} + E^{(n+1)}. \quad (6)$$

2. Gradient and Hessian of the Cosserat Energy

We should revisit the matrix multiplication form of quaternion multiplication first.

$$\mathbf{pq} = Q(\mathbf{p})\mathbf{q} = \begin{bmatrix}\Re(\mathbf{p}) & -\Im(\mathbf{p})^T\\ \Im(\mathbf{p}) & \Re(\mathbf{p})\mathbf{1}_{3\times3} + [\Im(\mathbf{p})]^\times\end{bmatrix}\begin{bmatrix}\Re(\mathbf{q})\\ \Im(\mathbf{q})\end{bmatrix}, \quad (7)$$

where the matrix $[\mathbf{p}]^\times$ is used to represent the vector cross product as a matrix-vector product $\mathbf{p}\times\mathbf{q} = [\mathbf{p}]^\times\mathbf{q}$. Right multiplying a quaternion can also be written in the form of a matrix-vector product:

$$\mathbf{pq} = \hat{Q}(\mathbf{q})\mathbf{p} = \begin{bmatrix}\Re(\mathbf{q}) & -\Im(\mathbf{q})^T\\ \Im(\mathbf{q}) & \Re(\mathbf{q})\mathbf{1}_{3\times3} - [\Im(\mathbf{q})]^\times\end{bmatrix}\begin{bmatrix}\Re(\mathbf{p})\\ \Im(\mathbf{p})\end{bmatrix}. \quad (8)$$

According to [2], the derivative of a rotated vector $\mathbf{R}(\mathbf{q})\mathbf{p}$ w.r.t. $\mathbf{q}$ is

$$\frac{\partial(\mathbf{R}(\mathbf{q})\mathbf{p})}{\partial\mathbf{q}} = 2\hat{Q}(\mathbf{p}\bar{\mathbf{q}})_{3\times4}, \quad (9)$$

where $(\cdot)_{3\times4}$ means we only take the lower 3×4 part of the matrix.

Next, we give the specific derivatives of the strain measure and the Darboux vector. Take the derivative of $\boldsymbol{\Gamma}_{ij}$ with respect to $\mathbf{x}_i$, we get

$$\frac{\partial\boldsymbol{\Gamma}_{ij}}{\partial\mathbf{x}_i} = -\frac{1}{l}(R(\mathbf{q}_{ij})R(\mathbf{q}_{ij}^0))^T. \quad (10)$$

Take the derivative of $\boldsymbol{\Gamma}_{ij}$ w.r.t. $\mathbf{q}_i$, we get

$$\frac{\partial\boldsymbol{\Gamma}_{ij}}{\partial\mathbf{q}_i} = \frac{1}{l}R(\mathbf{q}_{ij}^0)^T\hat{Q}(\partial_s\mathbf{x}\mathbf{q}_{ij})_{3\times4}diag(1,-1,-1,-1)\frac{\partial\mathbf{q}_{ij}}{\partial\mathbf{q}_m}, \quad (11)$$

where

$$\frac{\partial\mathbf{q}_{ij}}{\partial\mathbf{q}_m} = \frac{\mathbf{1} - \mathbf{q}_{ij}\mathbf{q}_{ij}^T}{\|\mathbf{q}_m\|}. \quad (12)$$

The derivative of $\boldsymbol{\Omega}$ w.r.t. $\mathbf{q}_i$ is

$$\frac{\partial\boldsymbol{\Omega}_{ij}}{\partial\mathbf{q}_i} = -\frac{2}{l}(\frac{1}{2}\hat{Q}(\mathbf{q}_j - \mathbf{q}_i)diag(1,-1,-1,-1)\frac{\partial\mathbf{q}_{ij}}{\partial\mathbf{q}_m} - Q(\bar{\mathbf{q}}_{ij})) \quad (13)$$

The force derived from E_{SE} and the torque derived from E_{BT} have been calculated in Eqns. (Main-26, Main-27), the force derived from E_{SE} can be calculated by the following equation

$$\widetilde{\mathbf{T}}_i^{SE} = -\nabla_{\mathbf{q}_i}E_{ij}^{SE} = -(\frac{\partial E_{ij}^{SE}}{\partial\boldsymbol{\Gamma}_{ij}}\frac{\partial\boldsymbol{\Gamma}_{ij}}{\partial\mathbf{q}_i})^T = -l(\frac{\partial\boldsymbol{\Gamma}_{ij}}{\partial\mathbf{q}_i})^T C^\Gamma\boldsymbol{\Gamma}_{ij}. \quad (14)$$

It is very difficult to further derive these derivatives with respect to quaternions using the quaternion representation.

*Corresponding author
ORCID(s):

For simplicity, we use the Gauss-Newton method, using an approximate Hessian matrix to ignore the terms containing the second derivative of $\mathbf{\Gamma}$ and $\mathbf{\Omega}$. For example,

$$\begin{aligned}\mathbf{H}^{SE}_{\mathbf{x}_i,\mathbf{x}_i} &\approx l(\frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i})^T C^{\Gamma} \frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i},\\ \mathbf{H}^{SE}_{\mathbf{q}_i,\mathbf{x}_i} &\approx l(\frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{q}_i})^T C^{\Gamma} \frac{\partial \mathbf{\Gamma}_{ij}}{\partial \mathbf{x}_i}.\end{aligned} \tag{15}$$

The other blocks of the Hessian matrix are calculated similarly.

3. Asymmetry Introduced by Torques in [3]

In this section, we first construct a linear system in angular velocity level following Baraff et al.[1]. We take the derivative of the shear torque in [3] to show that the resulting coefficient matrix of the linear system is asymmetric. In a similar way, we explain that the twist torque in [3] is not integrable because its torque derivative matrix in orientation level is asymmetric.

For any discrete element i, the implicit Euler scheme for updating orientation is

$$\begin{cases}\Delta \mathbf{q}_i = \frac{1}{2} h \begin{bmatrix}0\\ \boldsymbol{\omega}_i^{(n+1)}\end{bmatrix} \mathbf{q}_i^n\\ \Delta \boldsymbol{\omega}_i = h \mathbf{I}_i^{-1} \mathbf{T}_i^{(n+1)}.\end{cases} \tag{16}$$

Here we consider a simplified case where the torque $\mathbf{T}_i$ depends only on orientation, which is sufficient to show that the coefficient matrix of the angular velocity part in the linear system is asymmetric. Applying a first-order Taylor series expansion to T_i

$$\Delta \boldsymbol{\omega}_i = h \mathbf{I}_i^{-1} (\mathbf{T}_i^{(n)} + \sum_k \frac{\partial \mathbf{T}_i}{\partial \mathbf{q}_k} \Delta \mathbf{q}_k). \tag{17}$$

Eliminating Δq from the above equation, using matrix multiplication in Eqn. (8) instead of quaternion multiplication, we obtain

$$\Delta \boldsymbol{\omega}_i = h \mathbf{I}_i^{-1} (T_i^{(n)} + \frac{h}{2} \sum_k \frac{\partial T_i}{\partial \mathbf{q}_k} \hat{Q}(\mathbf{q}_k) \begin{bmatrix}0\\ \boldsymbol{\omega}_k^{(n+1)}\end{bmatrix}). \tag{18}$$

After regrouping, each particle has the following equation

$$\begin{aligned}&(\mathbf{I}_i - \frac{h^2}{2} \frac{\partial T_i}{\partial \mathbf{q}_i} \hat{Q}(\mathbf{q}_i) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}) \Delta \boldsymbol{\omega}_i - \frac{h^2}{2} \sum_{k \neq i} \frac{\partial T_i}{\partial \mathbf{q}_k} \hat{Q}(\mathbf{q}_k) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix} \Delta \boldsymbol{\omega}_k\\ &= h(T_i^{(n)} + \frac{h}{2} \sum_k \frac{\partial T_i}{\partial \mathbf{q}_k} \hat{Q}(\mathbf{q}_k) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix} \boldsymbol{\omega}_k), i = 1, ..., m,\end{aligned} \tag{19}$$

, forming a linear system. $\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}$ is a 4×3 matrix, and its upper and lower parts are respectively a 3×3 identity matrix and a 1×3 zero vector. Defining $K_{ij} = \frac{\partial T_i}{\partial \mathbf{q}_j} \hat{Q}(\mathbf{q}_j) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}$ and $K_{ji} = \frac{\partial T_j}{\partial \mathbf{q}_i} \hat{Q}(\mathbf{q}_i) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}$, it is easy to see that K_{ij} and K_{ji} are 3×3 block matrices in the symmetrical position of the coefficient matrix(omit $\frac{h^2}{2}$).

In [3], the shear torque of the bond are obtained by calculating the rotation angle between the shear direction and the normal direction. The shear direction $\mathbf{d}_s$ is defined as

$$\begin{bmatrix}0\\ \mathbf{d}_s\end{bmatrix} = \frac{1}{2} (\mathbf{q}_i \begin{bmatrix}0\\ \mathbf{d}_0\end{bmatrix} \bar{\mathbf{q}}_i + \mathbf{q}_j \begin{bmatrix}0\\ \mathbf{d}_0\end{bmatrix} \bar{\mathbf{q}}_j). \tag{20}$$

where $\mathbf{d}_0$ is the initial bond direction. The shear torque applied to particle i is

$$\mathbf{M}_i^s = \frac{1}{2} l k_s (\mathbf{n} \times \frac{\mathbf{d}_s}{\|\mathbf{d}_s\|}), \tag{21}$$

where $\mathbf{n} = (\mathbf{x}_i - \mathbf{x}_j) / \left\|\mathbf{x}_i - \mathbf{x}_j\right\|$. The torque with respect to the opposite particle j is $\mathbf{M}_j^s = \mathbf{M}_i^s$.

Now we can substitute $\mathbf{M}^s$ into the block matrices K_{ij} and K_{ji} from the coefficient matrix in Eqn. (19)

$$\begin{aligned}K_{ij} &= \frac{\partial \mathbf{M}_i^s}{\partial \mathbf{d}_s} \frac{\partial \mathbf{d}_s}{\partial \mathbf{q}_j} \hat{Q}(\mathbf{q}_j) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix},\\ K_{ji} &= \frac{\partial \mathbf{M}_i^s}{\partial \mathbf{d}_s} \frac{\partial \mathbf{d}_s}{\partial \mathbf{q}_i} \hat{Q}(\mathbf{q}_i) \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}.\end{aligned} \tag{22}$$

It can be seen that the front part of K_{ij} and K_{ji} are both $2\frac{\partial \mathbf{M}_i^s}{\partial \mathbf{d}_s}$, while the back part is only related to $\mathbf{q}_i$ and $\mathbf{q}_j$ respectively. $\mathbf{q}_i$ and $\mathbf{q}_j$ are completely independent variables, so for any $\mathbf{q}_i$, there is a $\mathbf{q}_j$ such that $K_{ij} \neq K_{ji}^T$, i.e, the coefficient matrix is asymmetric.

For implicit integration in orientation level, we first convert the torque into the torque parameter according to Eqn. (Main-28), because the torque parameter can be considered as the first derivative of energy with respect to the quaternion. We further take derivatives of the obtained torque parameters, and find that its torque derivative matrix is asymmetric.

The torque parameter applied to the discrete element i, j has the following forms respectively,

$$2\hat{Q}(\mathbf{q}_i) \begin{bmatrix}0\\ \mathbf{T}_i\end{bmatrix}, 2\hat{Q}(\mathbf{q}_j) \begin{bmatrix}0\\ \mathbf{T}_j\end{bmatrix}. \tag{23}$$

In [3], the twist torque has the following calculation form,

$$\mathbf{M}_i^t = k_t (\theta \mathbf{a} \cdot \mathbf{n}) \mathbf{n}, \tag{24}$$

where $\mathbf{a} = \frac{\Im(\mathbf{q}_t)}{|\Im(\mathbf{q}_t)|}, \theta = 2 \arccos \Re\left(\mathbf{q}_t\right)$, and $\mathbf{q}_t = \mathbf{q}_j \bar{\mathbf{q}}_i$. And the twist torque with respect to the opposite particle j is $\mathbf{M}_j^t = -\mathbf{M}_j^t$.

Here, if we use L_{ij}, L_{ji} to represent the derivative of torque paramters in Eqn. (23), we have

$$\begin{aligned}L_{ij} &= 2\hat{Q}(\mathbf{q}_i) \begin{bmatrix}0\\ I\end{bmatrix} \frac{\partial \mathbf{M}_i^t}{\partial \mathbf{q}_t} \hat{Q}(\bar{\mathbf{q}}_i)\\ L_{ji} &= -2\hat{Q}(\mathbf{q}_j) \begin{bmatrix}0\\ I\end{bmatrix} \frac{\partial \mathbf{M}_i^t}{\partial \mathbf{q}_t} Q(q_j) diag(1, -1, -1, -1).\end{aligned} \tag{25}$$

The asymmetry of the matrix can be found like Eqn. (22), which means that the torque in [3] is not integrable in orientation level.

References

[1] Baraff, D., 1997. An introduction to physically based modeling: rigid body simulation i—unconstrained rigid body dynamics. SIGGRAPH course notes 82.

[2] Kugelstadt, T., Schömer, E., 2016. Position and orientation based cosserat rods., in: Symposium on Computer Animation, pp. 169–178.

[3] Lu, J.M., Li, C., Cao, G.C., Hu, S.M., 2021. Simulating fractures with bonded discrete element method. IEEE Transactions on Visualization and Computer Graphics .

[4] Spillmann, J., Teschner, M., 2007. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 63–72.