Peridynamics-Based Fracture Animation for Elastoplastic Solids

Wei Chen¹, Fei Zhu¹, Jing Zhao²,³, Sheng Li¹ and Guoping Wang¹

¹Department of Computer Science, Peking University, Beijing, China
{suitmyself, feizhu, lisheng, wgp}@pku.edu.cn
²School of Information Science and Engineering, Yanshan University, Hebei Province, China
zhaojing@ysu.edu.cn
³Key Laboratory of Computer Virtual Technology and System Integration, Hebei Province, China

Abstract

In this paper, we exploit the use of peridynamics theory for graphical animation of material deformation and fracture. We present a new meshless framework for elastoplastic constitutive modelling that contrasts with previous approaches in graphics. Our peridynamics-based elastoplasticity model represents deformation behaviours of materials with high realism. We validate the model by varying the material properties and performing comparisons with finite element method (FEM) simulations. The integral-based nature of peridynamics makes it trivial to model material discontinuities, which outweighs differential-based methods in both accuracy and ease of implementation. We propose a simple strategy to model fracture in the setting of peridynamics discretization. We demonstrate that the fracture criterion combined with our elastoplasticity model could realistically produce ductile fracture as well as brittle fracture. Our work is the first application of peridynamics in graphics that could create a wide range of material phenomena including elasticity, plasticity, and fracture. The complete framework provides an attractive alternative to existing methods for producing modern visual effects.

Keywords: peridynamics, fracture, elastoplasticity

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The simulation of deformable materials has been an important research topic in computer graphics for decades, since the early work by Terzopoulos and colleagues [TPBF87]. One of the strongest driving forces behind the active research is the persistently growing need for higher realism from the visual effects industry. Materials in the real world exhibit complex behaviours, such as coupled elastoplastic deformations, fracture, etc. The complicated material behaviours are difficult to replicate by any single method despite the numerous ones that have been developed thus far. Existing approaches generally excel at some phenomena but would stumble (if not fail) at others. For instance, mesh-based methods [MG04, ITF04, TSIF05, SB12] are a good choice to simulate elastic deformations whereas not preferred for phenomena that involve topological changes. Particle-based methods [MCG03, PKA*05, SSC*13] are considered suitable for modelling topological changes, however the inherent loss of connectivity information would cause undesirable numerical fracture [LZLW11, ZZL*16] while simulating large deformations.

We build on recent developments of peridynamics theory in the computational physics community [Sil00, SEW*07, Mit11, ELP13, MO14] and propose a novel framework for graphical animation of varied deformation behaviours and fracture. Our aim is to enrich available options of simulation techniques for easier and better animation production. Peridynamics was first adopted to animation applications by Levine et al. [LBC*14] where they described a simple spring-mass system to handle brittle fracture of solids. In contrast, we handle elastoplasticity, brittle fracture and ductile fracture in a single framework. To this end, we propose several novel contributions in this work. We first present an elastoplastic constitutive model in the peridynamics-based framework with simple extension to anisotropy, and the model is validated against results produced by finite element method (FEM). Furthermore, we show that both brittle and ductile fracture phenomena can be naturally represented with nearly no effort by integrating a simple fracture criterion into this material model. This is due to the integral-based formulation of peridynamics, in which forces at a material point are computed by gathering contributions from all material points in its...
interaction range through integration. On the other hand, methods
based on classical continuum mechanics formulate force computa-
tions with partial differential equations that fail to be applicable
directly on singularities such as a crack. This feature makes our
peridynamics-based framework more attractive over existing ap-
proaches for producing animations that involve fracture. Lastly, our
method is simple to implement and trivially parallelizable, providing
a useful alternative to previous methods for animation production.

2. Related Work

A large body of literature has been devoted to physical simulation
of natural phenomena as a result of active research. A complete
literature review is beyond the scope of this paper. In the following,
we comment only on the representative works most related to ours.

Elastoplasticity Animation. The modelling of deformable plas-
ticity in graphics dates back to the pioneering work by Terzopoulos
and Fleischer [TF88]. O’Brien and colleagues [OBH02] incorpo-
rated a similar additive plasticity model into a finite element simu-
lation to animate ductile fracture. The strain measure was decom-
posed into two components, where one is due to elastic deformation
and the other due to plastic deformation. Müller et al. [MKN*04]
employed this model in their point-based framework and simulated
plastic behaviours of objects. Irving et al. [ITFO4] presented a mul-
tiplicative formulation of plasticity and pointed out that their model
was better handling finite plastic deformation than the additive one.
In contrast to the additive model, they decomposed the deformation
gradient into two components through multiplication. The multi-
plicative model was extensively used by later works to animate
phenomena that involve plasticity. Bargteil et al. simulated large
viscoplastic flow [BWHT07], Gerszewski and his colleagues ani-
imated elastoplastic solids [GBB09] and Stomakhin et al. modelled
plasticity of snow [SSC*13], just to name a few. Unfortunately,
neither of the above plasticity models applies in the peridynamics
framework because there is no concept of strain nor deformation
gradient in the integral-based formulation. As a result, we present
a new constitutive model for peridynamics in this work to animate
elastoplastic solids.

Fracture Animation. Numerous methods have been proposed on
fracture animation [MBP14, WWD15] because the stunning phe-
nomenon of fracture and failure is an indispensable visual element
in animated movies and video games. Early approaches use sim-
ple schemes to model fracture, such as the finite difference method
[TF88], the mass-spring system [NTB*91] and the mass-point con-
straint system [SWB01]. O’Brien and Hodgins [OH99] adopted
techniques from continuum mechanics and presented a FEM-based
method to simulate brittle fracture of solids. They later extended
their method to ductile fracture by incorporating a plasticity model
[OBH02]. Müller et al. [MMDJ01] employed a quasi-static finite
element analysis to animate brittle fracture of stiff materials under-
going collisions. Parker and O’Brien [PO09] presented some useful
techniques for real-time simulation of fracture in game environ-
ment. One major issue in FEM-based methods is the generation
of fracture patterns on meshes, which could alter the underlying
mesh topology. Early methods typically made use of simple separa-
tion along mesh element boundaries [NTB*91, MMA99, SWB01,
MMDJ01] or even element deletion [FDA02]. Mesh subdivision
prior to splitting could somewhat increase the available geometric
details [MK00, BG00], whereas this tended to introduce elements
with poor aspect ratios. Allowing failure along more arbitrary paths
could generate more geometrically rich fracture patterns [NF99,
OH99, OHB02], albeit at the expense of complicated re-meshing.
Molino et al. [MBF04] proposed a virtual node algorithm to avoid
the complexity of remeshing, where elements were duplicated into
partially filled counterparts with virtual nodes. The virtual node
algorithm was frequently used by subsequent works on fracture
animation [BHTF07] and mesh cutting [SDF07, WJS14] due to
its simplicity compared to re-meshing methods. Kaufmann et al.
[KMB*09] adapted the extended finite element method (XFEM)
that enriches approximation by custom-designed basis functions, instead
of actual/virtual element cutting. Other representative mesh-based
methods resorted to modal analysis [GMD13] and pure geometric
mesh decompositions [MCK13, SO14] for real-time brittle fracture.
Most recently, several works explored the boundary element method
for rigid body fracture [ZBG15, HW15] where only surface meshes
were employed for both representation and computation.

In contrast to mesh-based approaches, meshless methods are
generally considered as a better solution for animating topological
changes. Based on the moving least square (MLS) meshless framew-
ork by Müller et al. [MKN*04], Pauly and colleagues [PAK*05]
developed a novel meshless method for fracture animation of elast-
plastic solids. Their method generates detailed crack surfaces and
allows arbitrary crack initiation/propagation. Steinemann et al.
[SOG09] employed surface mesh representation in meshless frame-
work and presented a novel surface tracking technique to efficiently
split the meshless deforming objects. Inspired by the rigid body
assumption for simulating brittle fracture, Liu et al. [LHLW11]
employed quasi-static analysis in a meshless local Petrov–Galerkin
framework. Stomakhin et al. modelled the fracture of snow us-
ing a meshless material point method [SSC*13]. Hegemann et al.
[HJST13] combined a level set–based mesh embedding technique with
the material point method to animate dynamic ductile fracture.

Peridynamics. The peridynamics theory was first proposed by
Silling [Si00] as a nonlocal reformulation of classical solid me-
chanics. It contrasts with classical (local) theory in that the state
of a material point is influenced by not necessarily the material
points located in its immediate vicinity, but also those over long
distances. The governing equations of the peridynamics theory are
spatial integral equations instead of partial differential equations.
The theory was further developed by subsequent works [SEW*07,
ELP13], and its applications to the engineering field such as multi-
scale material modelling [ABL*08, SC14] and fracture modelling
[AXS06, SWAB10, SA14] were studied. A comprehensive review
of the research literature in the computational physics community is
beyond our scope, we refer the readers to the book by Madenci and
Oterkus [MO14]. Levine et al. [LBC*14] first introduced peridy-
namics to graphics for fracture animation. Their method was limited
to brittle fracture of isotropic elastic materials with a single Poisson
ratio of 0.25. Our work, on the other hand, is a complete framework
that models elastoplasticity and anisotropy under various parameter
settings, representing brittle and ductile fracture with high realism.

3. Background

In the peridynamics theory, any material point \( x \) interacts with other
material points within a distance \( \delta \). The distance \( \delta \) is called the
Figure 1: A sphere shoots through walls made of different materials, causing varied fracture behaviours. From left to right: isotropic brittle fracture, anisotropic brittle fracture, isotropic ductile fracture and anisotropic ductile fracture.

The horizon of \( x \), and the material points within the horizon are referred as its family, \( H_x \). There are infinite number of family members for a material point before discretizing the continuum into discrete particles. Figure 2 is an illustration of the peridynamics discretization with particles. It seems analogous to other meshless methods based on classical theory [MCG03, MKN*04], and the difference lies in the scale of interaction radius \( \delta \). In the case of the classical (local) continuum model, the state of a particle is influenced by only particles in its immediate vicinity. For case of the peridynamics theory, however, the state of a particle is influenced by particles within a region of finite radius. The peridynamics theory is thus referred as a non-local theory. As the radius becomes infinitely large, the peridynamics theory becomes the continuous version of the molecular dynamics model. As the radius becomes smaller, it becomes the continuum mechanics model. Therefore, the peridynamics model establishes a connection between the continuum mechanics and molecular dynamics models.

Our motivation for choosing peridynamics is that it is more favourable to handle material discontinuities, such as cracks. This benefit inherently from an integral force formulation of its governing equations, which stands in contrast to the partial differential equations used in the classical formulations. As we know, spatial derivatives are not well defined at discontinuities. Therefore, special treatment is generally required for fracture modelling in existing methods that are based on classical continuum mechanics. For instance, the mesh-based methods [OH99, OBH02] employed remeshing operations and the meshless method by Pauly et al. [PKA*05] altered the particle weight functions. The peridynamics governing equations remain valid at discontinuities, and material damage is represented as part of the peridynamics constitutive model. These attributes permit fracture initiation and propagation to be modelled with arbitrary paths in the peridynamics framework.

In peridynamics, the governing equation at any point \( x \) is formulated in integral form as below:

\[
\rho \ddot{u}(x) = \int_{H_x} [T(x', x) - T(x, x')] dH + b(x),
\]

where \( \rho \) is the mass density, \( u \) denotes the displacement, \( b \) is the external loads due to gravity and impact forces and \( x' \) is one material point that belongs to the family \( H_x \) of \( x \). \( T(x', x) \) and \( T(x, x') \) are two essential terms in which the constitutive laws of materials are encoded. \( T(x', x) \) represents the internal force density exerted by \( x' \) on \( x \), and \( T(x, x') \) is the other way around. The two terms both appear in the governing equation to enforce the Newton’s third law, and

Figure 2: Illustration of peridynamics discretization. A continuum is represented as particles (pink dots), and any particle (green dot) interacts with the particles within its horizon (green circle).

Figure 3: An armadillo is initially anchored on its back and four limbs, and it deforms elastically when its back is released.
similar strategy was employed in smoothed particle hydrodynamics (SPH) methods [MCG03]. The angle brackets representation \( \langle \cdot \rangle \) was defined by Silling et al. [SEW*07] as a function inside the family \( H_e \), which they called as a state. Please note the integral form of the equation, which is the key difference between peridynamics and classical theory. The entire framework is built on displacements \( u \) instead of their spatial derivatives, thereby making discontinuity modelling trivial. With particle discretization, the integration within \( H_e \) is represented as summation over family particles:

\[
\rho \ddot{u}(x) = \sum_{x' \in H_e} \left[ T(x', x) - T(x, x') \right] V_e + b(x),
\]

with \( V_e \) as the volume of particle \( x' \).

4. Elastoplastic Model

In this section, we describe our constitutive model for peridynamics in detail. We start with the basic isotropic elastic model, then plasticity is incorporated, and finally we extend the model with anisotropy.

4.1. Isotropic elasticity

As is discussed in Section 3, the key to peridynamics-based constitutive modelling is the design of proper internal force density \( T(\cdot) \). Silling et al. [SEW*07] showed that peridynamic constitutive models can be designed to match many hyperelastic constitutive models under the classical elasticity theory. We derive our model based on the model described by Madenci and Oterkus [MO14], which matches the isotropic linear elasticity model in classical theory. The elastic internal force exerted by particle \( j \) on particle \( i \) is defined as below:

\[
T_i(x_j, x_i) = \frac{1}{2} \rho A \frac{y_j - y_i}{|y_j - y_i|},
\]

where \( x \) and \( y \) denote the positions of particles before and after deformation, respectively. The direction of the force density is along the deformed bond between the particles given by \( Y_j-Y_i \). \( A \) is a scalar that represents the force magnitude, and it is composed of two terms by addition \( A = A_{\text{dil}} + A_{\text{dev}} \), namely, the dilatation term \( A_{\text{dil}} \) and the deviatoric term \( A_{\text{dev}} \).

The dilatation term \( A_{\text{dil}} \) is due to the dilatation part of deformation, i.e., volume change without any shape distortion. It is defined as

\[
A_{\text{dil}} = 4\omega_{ij} b \frac{y_j - y_i}{|y_j - y_i|} \frac{x_j - x_i}{|x_j - x_i|} \theta_i,
\]

where \( \omega \) is a peridynamics material parameter and \( \omega_{ij} \) is the weight function between particle \( i \) and particle \( j \). For isotropic materials, \( \omega_{ij} \) is monotonically decreasing with respect to the distance between particles

\[
\omega_{ij} = \frac{\delta}{|x_j - x_i|}.
\]

Note that \( \omega_{ij} \) is defined in the material space, therefore can be pre-computed. The term \( \theta_i \) measures the dilatation at particle \( i \), which is defined with respect to the stretch of all bonds between particle \( i \) and its family:

\[
\theta_i = \frac{9}{4\pi \delta^2} \sum_{k=1}^{N} \omega_{ik} \frac{x_k - x_i}{|x_k - y_i|} \cdot (x_k - x_i) V_k,
\]

where \( N \) represents the number of family points \( k \) for point \( i \), and \( V_k \) are their volumes. The stretch \( s_{ik} \) of the bond between particles is defined as

\[
s_{ik} = \frac{|y_i - y_k|}{|x_i - x_k|} - 1.
\]

The deviatoric term \( A_{\text{dev}} \) is a result of distortion that does not cause change in volume. It is defined with respect to the deviatoric component of bond extension:

\[
A_{\text{dev}} = 4\pi \rho b \left( e_{ij} - \frac{\delta}{4 \rho |y_j - y_i|} \frac{x_j - x_i}{|x_j - x_i|} \theta_i \right),
\]

where \( b \) is a material parameter. We denote \( e_{ij} = |y_j - y_i| - |x_j - x_i| \) as the extension of the bond between particle \( i \) and particle \( j \). The term in brackets of Equation (8) is the deviatoric component of bond extension \( e_{ij}^d \):

\[
e_{ij}^d = e_{ij} - \frac{\delta}{4 \rho |y_j - y_i|} \frac{x_j - x_i}{|x_j - x_i|} \theta_i.
\]

Intuitively, \( e_{ij}^d \) is constructed by removing the dilatation component of bond extension from the total bond extension \( e_{ij} \).

In summary, the behaviour of our isotropic elastic model is controlled by two material parameters \( a \) and \( b \). The model is equivalent to the isotropic linear elasticity model in classical theory, please refer to the supplementary document for elaborated derivation. Here, we directly provide the conversion between the material parameters in this model and those from continuum mechanics:

\[
a = \frac{9 \kappa}{8\pi \delta^3}, \quad b = \frac{15 \mu}{2\pi \delta^3},
\]

where \( \kappa \) and \( \mu \) denote the bulk modulus and the shear modulus, respectively. In Figure 3, we demonstrate an example of the hyperelastic deformations animated with our model.

4.2. Plasticity

Plasticity model for peridynamics is less studied in literature due to its complexity. To our knowledge, Silling et al. [SEW*07] proposed the first plasticity model that is analogous to the von Mises flow model in classical theory. Mitchell presented a new framework for peridynamics-based plasticity modelling [Mit11] based on Silling et al.’s model, whereas the model has not been verified by experiments thus far. We adopt Mitchell’s model for practical applications, and propose novel modifications based on their work.
Figure 4: Comparison of elastic and plastic deformations by shooting a sphere at two walls made of different materials with identical initial configuration as in (a). The elastic wall deforms on impact (b), and recovers afterwards (c). The plastic wall undergoes permanent deformation (e) after the impact (d).

Our plasticity model is based on purely deviatoric plastic flow theory, therefore we start by decomposing the deviatoric bond extension $e_{ij}$ (see Equation (9)) into two components by addition:

$$ e_{ij} = e_{ij}^e + e_{ij}^p, $$

(11)

where $e_{ij}^e$ and $e_{ij}^p$ are the elastic and plastic part of the total deviatoric bond extension, respectively. To incorporate plasticity into the constitutive model, the deviatoric part of internal force density (see Equation (8)) is now redefined as below:

$$ A_{\text{dev}} = 4\omega_{ij}b(e_{ij}^e - e_{ij}^p) $$

(12)

with the contribution of plastic deviatoric bond extension removed from force computation. In case of elastic deformations, the term $e_{ij}^p$ vanishes and Equation (12) conforms to Equation (8).

A simple yield function $f(A_{\text{dev}})$ is used to determine whether deformation has entered the plasticity regime:

$$ f(A_{\text{dev}}) = \frac{(A_{\text{dev}})^2}{2} - \Psi_0, $$

(13)

where $\Psi_0$ is a critical parameter. The deformation is elastic if $f(A_{\text{dev}}) \leq 0$, and plasticity is present if $f(A_{\text{dev}}) > 0$.

In case of plastic deformations, we project $A_{\text{dev}}$ onto the yield surface to obtain a critical value of deviatoric force density $A_{\text{dev}}^c$:

$$ A_{\text{dev}}^c = \sqrt{2\Psi_0}\text{sign}(A_{\text{dev}}), $$

(14)

where sign(·) is the sign function. $A_{\text{dev}}^c$ is used to compute the increment of plastic deviatoric bond extension:

$$ \Delta e_{ij}^p = \frac{1}{2b}(A_{\text{dev}} - A_{\text{dev}}^c), $$

(15)

The subscripts $n$ and $n+1$ denote the discretized point of time at which the bond extensions and corresponding increments are evaluated. The parameter $\gamma$ which does not appear in the original model of Mitchell’s [Mit11] is used to enforce a limit on the amount of plasticity. We found in experiments that with this parameter we obtain more control over the plastic behaviours (see Figure 5) and the stability of simulation is improved as well. Figure 4 shows a comparison of the simulation results using our elastoplastic constitutive model. Our elastic model produces correct elastic behaviours, and permanent deformation is captured when plasticity is involved.

4.3. Anisotropy

Our constitutive model is isotropic up to now, and we extend it to anisotropy in this section. We model anisotropy by manipulating the weight functions between particles (see Equation (5)) with direction information. The key idea is to associate an anisotropy matrix $G$ with each particle, so that applying the transformation to the bond between particles biases the influence weight towards preferred directions. The weight function $\omega_{ij}$ for anisotropic materials is computed as below:

$$ \omega_{ij} = \frac{\delta}{|G(x_j - x_i)|}, $$

(17)

© 2017 The Authors

Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.
Appealing anisotropic effects could be generated with our anisotropy model. See Figure 1 for a demonstration of the model applied to brittle and ductile fracture animation. Figure 6 compares the crack patterns generated by the brittle fracture examples in Figure 1.

5. Fracture

In this section, we present our fracture model and introduce the mesh embedding strategy we employ to generate crack surfaces.

5.1. Fracture criterion

Material damage can be modelled in peridynamics by permanently eliminating the bonds between particles. The dynamics with discontinuities is trivial due to the integral-based nature of peridynamics. For elastic brittle materials, a simple critical stretch is generally used as the fracture criterion. This criterion conforms to the physically plausible energy release rate, and it has been validated before [SA05]. Levine et al. [LBC*14] also utilized this criterion for brittle fracture modelling in computer graphics. In order to account for plasticity and model ductile fracture, we redefined the elastic critical stretch criterion as

\[ s_{ij}^* = \frac{e_{ij}^p}{|x_i - x_j|} \]  

(18)

Unfortunately, we found in experiments that this fracture criterion would cause unrealistic artefacts since bonds with smaller rest lengths are sometimes more prone to breaking. We alleviate this problem by incorporating the weight function \( \omega_{ij} \) into the criterion to increase the fracture criterion of closer family members. Shattering effects could arise for brittle materials, which lead to many tiny fragments. We could avoid the generation of too small fragments by continuously increasing the crack threshold as the material gets damaged. The final fracture criterion that we employ in our model is formulated as below:

\[ s_{ij} = (1 + \alpha \phi) \frac{s_{ij}^*}{\omega_{ij}} = \frac{(1 + \alpha \phi) e_{ij} - e_{ij}^p}{\delta}, \]  

(19)

where \( \phi = 1 - \frac{n_i}{N_i} \) measures the damage level of material point \( i \). \( n_i \) and \( N_i \) are the numbers of active bonds connecting \( i \) with its family members in the deformed and initial configurations, respectively. \( n_i \) gradually decreases as more bonds around \( i \) are broken, increasing the damage level of \( i \). The parameter \( \alpha \) is set to 0 by default, and could be used to mitigate the shattering effect while non-zero values are given. Figure 7 provides an example of controlling the dust with different \( \alpha \) values. It shows that our criterion is able to produce compelling results in practical use.

5.2. Embedded mesh

While particle-based discretization offers great simplicity, this simplicity does come at a cost that it is difficult to generate surface representation. This naturally motivates the use of mesh embedding approach, in which the boundary of volumetric meshes could represent the original object surface and the newly generated crack surfaces during simulation. We use tetrahedron meshes to represent object geometry, and the particles are initialized at the barycentres of each mesh element. The particle family members are initialized according to the mesh connectivity and a pre-specified horizon \( \delta \). To accommodate the topology changes resulted from fracture, we propose a simple strategy that dynamically split the embedded mesh along the elements.

We achieve this by maintaining a crack face set and continuously adding the shared triangles to it for fracture happened between immediate elements. At each time step, we investigate merely those vertices that involved in crack face set and determine whether its connected tetrahedra have been separated by crack faces. If this is the case, we split the vertex, assign corresponding tetrahedra to each of them, and make a topological change on the embedded mesh. After splitting, we could safely remove only those crack faces that directly results in our vertex splitting. If the connected tetrahedra are separated into more than two groups, we accordingly split vertex into multiple copies. Simultaneous splitting of multiple vertices at one crack face are also compatible in our method. Figure 8 shows an example of using our strategy to produce complex crack surfaces.

After handling mesh topology, we update the vertex positions using the velocities of corresponding particles. A simple weighted-average approach is employed to update the vertex positions:

\[ \omega_v = \frac{1}{p} \sum_{p} \frac{1}{4} m_p, \]  

(20)

\[ v_v = \frac{1}{\omega_v} \sum_{p} \frac{1}{4} m_p v_p, \]  

(21)

\[ x_v^{t+1} = x_v^t + \Delta t v_v, \]  

(22)

where subscripts \( p \) and \( v \) represent the particle and the mesh vertex, respectively. \( m_v \) and \( v_v \) are the mass and velocity of the particle, \( v_v \) and \( x_v \) are the velocity and position of the mesh vertex.

6. Results

We present the results produced with our method in this section. All our examples are run on a 3.5 GHz, Intel Core i7-5930K CPU with 32 G RAM. The embedded tetrahedron meshes are generated...
Constitutive model validation. We validate our constitutive model by simulating deformations with varied material properties and performing comparisons with the results of FEM. Figure 3 shows an example of isotropic elastic deformations, where the back and four limbs of an armadillo are initially anchored and then the anchor on the back is removed. The deformations are plausible and no undesirable numerical fracture occurs when the arms of the armadillo are overstretched. In Figure 4, we compare the results of elasticity and plasticity, and our model produces correct deformation behaviours. Figure 5 demonstrates the varied effects produced by tuning the amount of maximum plasticity. Figure 9 compares the elastic deformations of stretching beams with different Poisson ratio values. Unlike Levine et al.’s model [LBC*14], our constitutive model is not limited to a single Poisson ratio. Finally, we conduct comparisons with FEM through Figures 10–13. The deformations of a bending beam in Figure 10 produced with our method are almost identical to those generated by FEM, under both stiff and soft material settings. We further demonstrate the accuracy of our method in constitutive modelling using comparisons with quantitative error analysis. The swing (see Figure 11) and twist (see Figure 12) deformations of the bar produced by our method are as accurate as those by corotated linear FEM with position deviations less than 10%. The position deviations are measured over the diagonal of the object’s bounding box: Error = \|x - x'\|/d where d is the length of the diagonal. Note that our constitutive model alleviates artefacts of the classical linear model albeit derived from it. It is because peridynamics does not employ the geometric linear approximation as the Cauchy strain in continuum mechanics does. Thus peridynamics does not suffer from ghost forces while undergoing rigid rotations. We also obtain nice accuracy for the non-cyclic vibrations of the armadillo presented in Figure 13. Therefore, we
Fracture animation. Our method could simulate brittle and ductile fracture with compelling visual realism. In Figure 1, a wide range of fracture behaviours are generated, including isotropic brittle fracture, anisotropic brittle fracture, isotropic ductile fracture and anisotropic ductile fracture. This demonstrates the capability of our method in simulating fracture. We believe our approach is the first peridynamics-based framework in graphics with such flexibility. Figure 8 shows an example of shooting a bullet into a jello-like object. Our method handles well the generation of the complex crack surfaces. The armadillo in Figure 14 is stretched until its limbs tear off. The behaviour of ductile fracture is correctly demonstrated, including the progressive generation of multiple cracks (see Figure 18). The glass wall in Figure 15 is pressed by a heavy metal ball. Cracks develop and propagate without shattering the glass into fragments. This phenomenon cannot be reproduced by the level set approach [HJST13] as mentioned in their paper, and it is challenging for the remeshing-based FEM methods [OH99, OBH02] considering the mesh operations. In contrast, our method handles the complicated propagation process well, including the branching and merging of cracks. The approach by Pauly et al. [PKA*05] could produce results comparable to ours, employing explicit handling of the topology events. Our method, however, requires none. In Figure 16, a bunny made of elastic material falls to the ground and shatters. Our method is able to realistically capture the secondary fracture of the fragments. Figure 17 is another demonstration of our method in handling crack tips. A thin sheet with initial cracks is torn on two sides. The cracks gradually proceed and finally shatter the sheet. Please note the filaments generated as a result of crack branching and merging.

Choice of $\delta$. We use a $\delta$ value of $1.0\lambda$ for most of the FEM comparisons, where $\lambda$ is the average edge length of the embedded tetrahedron mesh. As peridynamics converges to classical continuum mechanics while $\delta$ approaches zero [WA05], this specific choice of $\delta$ amounts to the nodal force computation of FEM using the 1-ring neighbours. Although not always necessary, fine-tuning the $\delta$ value is a viable way of getting well-synchronized results. For instance, in Figure 13, we adjust it to $1.38\lambda$ to achieve our best result. We studied the effect of different $\delta$ values with experiments, and the results reveal that simulation plausibility is not very sensitive to
Figure 15: A glass wall on ground is pressed by a heavy metal ball and cracks without separating. Our method can model the complicated crack propagation process, including branching and merging.

Figure 16: An elastic bunny falls to the ground and shatters into pieces. Note the secondary fracture of the fragments.

In Table 1, we list the detailed parameter settings and the performance data for all the examples presented in the paper. The results in this paper are produced with practical computation time. Note that we employ damping forces in a few of the examples, mainly to ease the burden of self-collision handling, which is not the major concern of this work. The experimented damping models include a simple air damping $v_{\text{new}} = (1 - \lambda_a) v_{\text{old}}$, and a Laplacian smoothing $v_{\text{new}} = v_{\text{old}} + \lambda_l L(v_{\text{old}})$. Please refer to Table 1 for the specific values of the damping coefficients $\lambda_a$ and $\lambda_l$ in each example. Although effective in practice, these damping forces are not a necessity for our method since the issue of self-penetration could be (and should be) addressed by superior detection and resolution strategies.

7. Discussions

We have introduced a novel meshless framework for graphical modelling and animation of elastoplastic solids. Our work is the first peridynamics-based framework in computer graphics that can simulate a wide range of material behaviours, including elasticity, plasticity and fracture.

Our work is not without limitations. Currently, we cannot afford large time steps because we used explicit time integration in our implementation. In the future, we plan to incorporate implicit time integration into our framework to achieve less restricted time step size. Another limitation of our work stems from the mesh embedding strategy for crack surface representation. The level of crack detail is highly dependent on the embedded mesh resolution. In addition, we generate crack surfaces by separating the mesh elements, which could cause the zig-zag artefact (see Figure 18). This artefact might be alleviated by smoothing the crack surfaces somewhat, or employing the virtual node algorithm. While we used multi-threading in our implementation, the performance of our method could be further improved with general purpose graphics processing units (GPGPU) techniques.

Other interesting avenues for future work include combining peridynamics theory with existing methods from classical theory such as FEM and SPH. These methods complement each other, and their combination could produce a new method that is more powerful for animation production.
Table 1: Model information, simulation parameters and performance data for all our examples. \( \lambda \) is the average edge length of the embedded tetrahedron mesh used for particle initiation.

<table>
<thead>
<tr>
<th>Examples</th>
<th>Particle num</th>
<th>( \delta )</th>
<th>Bond num</th>
<th>( \kappa ) (KPa)</th>
<th>( \mu ) (KPa)</th>
<th>( \rho ) (Kg/m(^3))</th>
<th>( \Psi_0 )</th>
<th>( \lambda_{ij} )</th>
<th>( \lambda_t )</th>
<th>Performance (s/step)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass Wall</td>
<td>3.0 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>5.0 \times 10^7</td>
<td>3.3 \times 10^7</td>
<td>2.0 \times 10^7</td>
<td>2200</td>
<td>( \infty )</td>
<td>0.0</td>
<td>0.0005</td>
<td>1.0 \times 10^{-6}</td>
</tr>
<tr>
<td>Plastic Wall</td>
<td>3.0 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>5.0 \times 10^7</td>
<td>3.3 \times 10^3</td>
<td>2.0 \times 10^3</td>
<td>2200</td>
<td>10^{23}</td>
<td>0.2</td>
<td>0.05</td>
<td>1.0 \times 10^{-4}</td>
</tr>
<tr>
<td>Armadillo [Figure 3]</td>
<td>4.2 \times 10^5</td>
<td>1.50( \lambda )</td>
<td>7.5 \times 10^7</td>
<td>6.3 \times 10^4</td>
<td>3.8 \times 10^4</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>1.0 \times 10^{-4}</td>
</tr>
<tr>
<td>Elastic Wall</td>
<td>3.0 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>5.0 \times 10^7</td>
<td>1.0 \times 10^4</td>
<td>6.0 \times 10^3</td>
<td>1200</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>5.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Plastic Wall</td>
<td>3.0 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>5.0 \times 10^7</td>
<td>1.0 \times 10^4</td>
<td>6.0 \times 10^3</td>
<td>1200</td>
<td>10^{26}</td>
<td>0.2</td>
<td>( \infty )</td>
<td>5.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Wall [Figure 5]</td>
<td>3.0 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>5.0 \times 10^7</td>
<td>1.0 \times 10^4</td>
<td>6.0 \times 10^3</td>
<td>1200</td>
<td>10^{26}</td>
<td>(0.1, 0.15, 0.2)</td>
<td>( \infty )</td>
<td>5.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Jello [Figure 8]</td>
<td>4.6 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>8.6 \times 10^7</td>
<td>1.0 \times 10^3</td>
<td>4.6 \times 10^2</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>0.4</td>
<td>5.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Beam Stretch [Figure 9]</td>
<td>2.2 \times 10^4</td>
<td>1.0 ( \lambda )</td>
<td>1.3 \times 10^6</td>
<td>2.0 \times 10^3</td>
<td>(9.2, 1.5, 2.2) \times 10^2</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>1.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Soft Beam Bend</td>
<td>4.5 \times 10^4</td>
<td>1.0 ( \lambda )</td>
<td>6.9 \times 10^6</td>
<td>6.3 \times 10^3</td>
<td>3.8 \times 10^3</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>5.0 \times 10^{-5}</td>
</tr>
<tr>
<td>Still Beam Bend</td>
<td>4.5 \times 10^4</td>
<td>1.0 ( \lambda )</td>
<td>6.9 \times 10^6</td>
<td>3.3 \times 10^4</td>
<td>2.0 \times 10^4</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>2.5 \times 10^{-5}</td>
</tr>
<tr>
<td>Bar Swing [Figure 11]</td>
<td>9.2 \times 10^3</td>
<td>1.11 ( \lambda )</td>
<td>1.5 \times 10^5</td>
<td>5.0 \times 10^3</td>
<td>5.0 \times 10^3</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>1.0 \times 10^{-4}</td>
</tr>
<tr>
<td>Bar Twist [Figure 12]</td>
<td>9.2 \times 10^3</td>
<td>1.34 ( \lambda )</td>
<td>1.5 \times 10^5</td>
<td>1.0 \times 10^3</td>
<td>6.0 \times 10^2</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>1.0 \times 10^{-4}</td>
</tr>
<tr>
<td>Armadillo Vibrate</td>
<td>2.0 \times 10^4</td>
<td>1.38 ( \lambda )</td>
<td>1.8 \times 10^6</td>
<td>5.0 \times 10^3</td>
<td>3.0 \times 10^3</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>( \infty )</td>
<td>5.0 \times 10^{-4}</td>
</tr>
<tr>
<td>[Figure 13]</td>
<td>Armadillo [Figure 14]</td>
<td>4.2 \times 10^5</td>
<td>1.50( \lambda )</td>
<td>7.5 \times 10^7</td>
<td>6.3 \times 10^4</td>
<td>3.8 \times 10^4</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>0.61</td>
</tr>
<tr>
<td>Bunny [Figure 16]</td>
<td>5.2 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>8.8 \times 10^7</td>
<td>2.5 \times 10^2</td>
<td>1.2 \times 10^2</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>0.13</td>
<td>5.0 \times 10^{-4}</td>
</tr>
<tr>
<td>Thin Sheet [Figure 17]</td>
<td>1.6 \times 10^5</td>
<td>1.45( \lambda )</td>
<td>2.0 \times 10^7</td>
<td>5.0 \times 10^3</td>
<td>3.0 \times 10^3</td>
<td>1000</td>
<td>( \infty )</td>
<td>0.0</td>
<td>0.1</td>
<td>5.0 \times 10^{-5}</td>
</tr>
</tbody>
</table>

© 2017 The Eurographics Association and John Wiley & Sons Ltd.
Figure 17: A thin sheet with initial cracks is torn apart. The cracks proceed with branching and merging, separating the sheet into pieces.

Acknowledgements

We acknowledge the anonymous reviewers for their comments and suggestions that greatly improve our work. The authors would also like to thank Xiaowei He, Sheng Yang and Liyou Xu for discussions during our research. The work is supported by Grant No. 2015BAK01B06 from The National Key Technology Research and Development Program of China, and Grant No. 61232014, 61421062, 61170205, 61472010 from National Natural Science Foundation of China.

References


Supporting Information
Additional Supporting Information may be found in the online version of this article at the publisher’s web site:

Supplementary Technical Document: It presents the derivation of the constitutive model.

Video S1