Peridynamics-Based Fracture Animation for Elastoplastic Solids

W. Chen, F. Zhu, J. Zhao, S. Li, G. Wang

1. Revision Summary

We have carefully revised our submission according to the suggestions of the reviewers. The major changes to the manuscript include (but not limited to):

1. Known typos are removed with thorough editing.
2. More information about the discretization of equations of motion is included.
3. XFEM related citation is included.
4. Revised description of fracture criteria (section 5.1).

In addition, we respond to the comments of each reviewer in this revision report. We try to answer the questions with clear clarification and detailed discussion.

2. Answers to Reviewers' Comments

Reviewer #1:

Q: *More explicit discussion on damping is preferred. Particularly, it is very discouraging to know the proposed method needs 'air damping' to stabilize the simulation.*

We apologize for the imprecise description in prior revision note that causes the misunderstanding of requiring damping forces to stabilize the simulation, which is not as a matter of fact. We used air damping to slow the motion of objects down so that the length of the video would not be too long, and self-collision handling would be easier (tricky idea). However, we only used it in very few of our examples, namely Figure 1(c)(d) and Figure 3. We submitted a video of the example in Figure 3 to support our point, where the settings were identical as before but the damping forces were removed. The simulation is still stable, only that our simple self-collision handling strategy fails to fully resolve the self-penetrations and it takes more time for the armadillo to reach rest. We also applied Laplacian smoothing of velocity to a few examples, where the smoothed velocity of a particle is computed as $\mathbf{v}_{new} = \mathbf{v}_{old} + \lambda \mathbf{L}(\mathbf{v}_{old})$. The smoothing is almost negligible since we set λ to a very small value of 0.001. We admit that the use of damping forces in our experiment is quite casual, but we argue it is not a necessity for our method. For instance, the issue of self-penetration could be addressed by superior detection and resolution strategies.

Reviewer #2:

Q: *The authors need to be aware of the limitations of mesh embedding strategy.*

Using pre-generated tetrahedral mesh for crack surface representation is not an optimal solution, since the quality of the crack surfaces highly depends on that of the tetrahedral mesh. For instance, there may be lack of smoothness and fine details. We could address these issues in the future by applying smoothing and dynamical remeshing to the tetrahedral mesh. We could also explore other crack surface representation strategy. It is an interesting future work which we believe is beneficial to other meshless methods as well.

Q: *Explicit comparisons to other methods(i.e., tet-based fracture following O'Brien's work and MLS fracture following Müller's or Pauly's work) is missing.*

O'Brien's seminal work [OH99, OBH02] requires element cutting and dynamical remeshing, which are somewhat complicated to implement. Methods based on the virtual node algorithm [MBF04] are simpler, but still not trivial. Other tet-based fracture work [SWB01, MMDJ01] cracks objects along element boundaries, which couples dynamics computation with surface representation. Crack propagation in these methods is not determined by stress analysis alone but by meshing as well, therefore accuracy is compromised. We also use embedded tet mesh for crack representation, but the physics is applied on the particles. The MLS fracture work of Pauly's [PKA*05] is a great method, which generates detailed crack surfaces and allows arbitrary crack initiation/propagation. However, the MLS approximation fails to handle degenerated particle configuration where neighbors of a particle are colinear. Besides, they used a series of techniques to handle the complex topology situations in crack propagation. In contrast, our method does not need any explicit handling.

In summary, our approach is framework for fracture simulation parallel to existing methods. The benefits of

our method include simplicity of implementation, trivial handling of discontinuities, and etc.

Reviewer #3:

Q: *How much does the interaction radius parameter δ affect the simulation quality? Do the comparisons with FEM hold up for any value of δ? Or does it need to be tuned? How sensitive is the range of values for which peridynamics gives good result, and are there rules of thumb for how δ should be selected?*

We provide a supplementary video that compares the simulation of a bending beam using four different δ values. The results are all physically plausible, which shows that the value of δ does not affect the simulation quality. We also find that the results with δ values of 1.0λ and 1.5λ are hardly distinguishable. Therefore the simulation results are not very sensitive to the parameter δ.

Peridynamics converges to the classical continuum mechanics while δ approaches zero [WA05]. Therefore we used a δ value of 1.0λ for most FEM comparisons, which amounts to the nodal force computation of FEM using the 1-ring neighbors of mesh nodes. Although not always necessary, fine-tuning the δ value is a viable way of getting well-synchronized results. For instance, in Figure 13, we adjust it to 1.38λ to achieve our best result.

In practice, we recommend a δ value in the range between 1.0λ and 2.0λ. Larger values could result in inadequate family members for particles close to object boundary, therefore causing the material to behave softer than it should be. Please refer to the bending beam with a δ value of 3.0λ in the right-bottom of the supplementary video. In this case, a correction for boundary particles needs to be taken into account.

Q: *There are missing steps for discretization. How do we go from equation 1 to discrete equations of motion, where $\mathbf{x}$ and $\mathbf{x}'$ are particles rather than arbitrary points in the material?*

We have revised the descriptions in section 3 to better clarify the issue of discretization, and the discrete form of equation 1 is added to the manuscript (equation 2). Please refer to the manuscript.

Q: *Can't the use of the weights defined in equation 4 (equation 5 in revised manuscript) be interpreted as just a discretization of divergence, except over the entire horizon rather than over an element neighborhood?*

We do not think the weight function in Peridynamics could be regarded as a discretization of divergence in FEM. In fact, by comparing the governing equations in both theories, we could easily derive that the divergence of stress in FEM is equivalent to the integration of force density over the horizon, which is

$$\nabla \cdot \sigma = \int_{H_\mathbf{x}} [\mathbf{T}\langle \mathbf{x}', \mathbf{x} \rangle - \mathbf{T}\langle \mathbf{x}, \mathbf{x}' \rangle] dV_{\mathbf{x}'}$$

From discretization, the integration on the right hand side is replaced by summation

$$\nabla \cdot \sigma = \sum_{H_\mathbf{x}} [\mathbf{T}\langle \mathbf{x}', \mathbf{x} \rangle - \mathbf{T}\langle \mathbf{x}, \mathbf{x}' \rangle] V_{\mathbf{x}'}$$

We note that it is the force summation over the horizon that should be interpreted as a discretization of divergence. Although weight functions are encoded in force density, ω is used only to compute the force summation in a weighted way, which is inherently different from other methods like FEM and SPH where weight functions are simutaneously used to calculate quantities' first-order or second-order derivatives (e.g. in SPH, $\nabla \cdot \mathbf{A_i} = -\frac{1}{\rho_i} \sum_j (\mathbf{A_i} - \mathbf{A_j}) \cdot \nabla \omega_{ij}$).

That is, in our formulation, it is only the displacements of material points that are involved, instead of their spatial derivatives, which consequently lead to the most significant difference between Peridynamic and other mesh (FEM) or meshless methods.

Q: *Correction for typos.*

We have fixed as many typos as we could with several passes of thorough editing. Please refer to the revised manuscript.

Q: *Lack discussion for XFEM-based methods in the related work.*

We have cited relevant XFEM-based fracture methods in the revised manuscript. Nevertheless, the discussion is limited to work in the graphics community. Discussion on the vast XFEM literature in engineer is beyond the scope of this paper.

Q: *How is equation 17 (equation 18 in revised manuscript) used, exactly? It is compared against some fixed user-specified threshold? What are "rest_bond_num" and "initial_bond_num" in the definition of ϕ?*

Equation 18 is the critical value of bond stretch for fracture. At run-time, the stretch of bonds between particles are computed via equation 7 and then compared against the critical value to determine whether fracture occurs. In fact, equation 18 amounts to the cauchy stretch with plastic deformation reasonably excluded. We further improve it to equation 19 by incorporating the weight function and increasing the critical value through ϕ as fracture proceeds. "rest_bond_num" is the rest active bonds during fracture and "initial_bond_num" is the initial total bonds before fracture. We have revised the description of this part in the manuscript.

Q: *Imprecise statements about the relationship between rotation artifacts, geometric linearity, and stress-strain linearity.*

We have removed the inappropriate statements from the revised manuscript.

Q: *For the FEM comparison examples: How many elements were used for each example? How did the performance compare to the performance of peridynamics?*

The number of elements used in FEM simulations are identical to the number of particles used in our peridynamics simulations. We initialized our particles in the centers of FEM elements. We did not measure the precise performance of the FEM simulations. In our implementation, the performance of FEM simulations is almost on a par with the peridynamics simulations.

Q: *What conclusions should be drawn from the supplemental video(random cube) in prior submission?*

The random cube video clip was meant to clarify that we did not take into account element inversions and thus our method failed to recover the cube from a randomly inverted configuration.

References

[MBF04] MOLINO N., BAO Z., FEDKIW R.: A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 385–392. 1

[MMDJ01] MÜLLER M., MCMILLAN L., DORSEY J., JAGNOW R.: Real-time simulation of deformation and fracture of stiff materials. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation* (New York, NY, USA, 2001), Springer-Verlag New York, Inc., pp. 113–124. 1

[OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Trans. Graph. 21*, 3 (July 2002), 291–294. 1

[OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 137–146. 1

[PKA*05] PAULY M., KEISER R., ADAMS B., DUTRÉ P., GROSS M., GUIBAS L. J.: Meshless animation of fracturing solids. *ACM Trans. Graph. 24*, 3 (July 2005), 957–964. 1

[SWB01] SMITH J., WITKIN A., BARAFF D.: Fast and controllable simulation of the shattering of brittle objects. *Comput. Graph. Forum 20*, 2 (2001), 81–90. 1

[WA05] WECKNER O., ABEYARATNE R.: The effect of long-range forces on the dynamics of a bar. *Journal of the Mechanics and Physics of Solids 53*, 3 (2005), 705 – 728. 2