juliacon

Simulation of fracture and damage with Peridynamics.jl

Kai Partmann[1], Manuel Dienst[1], and Kerstin Weinberg[1]

[1]Chair of Solid Mechanics, University of Siegen, Siegen, Germany

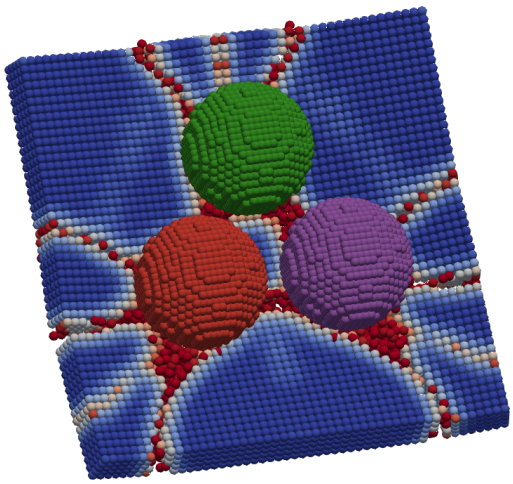

Fig. 1: High-velocity contact simulation of three spheres crashing into a rectangular panel; logo of Peridynamics.jl

1. Summary

Peridynamics is a nonlocal continuum mechanics formulation, which was introduced by Silling [16]. It has gained increased popularity as an approach for modeling fracture. The deformation of the solid is described by integro-differential equations that are also fulfilled for discontinuities, making it very capable of modeling crack propagation and fragmentation with large displacements. Much peridynamics research has been done in recent years, summarized in various review papers and books [3, 8, 12].
Typically, in peridynamics the continuum is discretized by material points. Points interact only with other points inside of their specified *neighborhood* or *point family* $\mathcal{H}$, which is defined as the set of points inside a sphere with the radius δ, also named the *horizon*. The interaction of the point $\boldsymbol{X}$ with its neighbor $\boldsymbol{X}'$ is called *bond* and defined as

$$\boldsymbol{\Delta X} = \boldsymbol{X}' - \boldsymbol{X} \ . \tag{1}$$

The equation of motion reads

$$\varrho\, \ddot{\boldsymbol{u}}(\boldsymbol{X}, t) = \boldsymbol{b}^{\mathrm{int}}(\boldsymbol{X}, t) + \boldsymbol{b}^{\mathrm{ext}}(\boldsymbol{X}, t) \ , \tag{2}$$

with the mass density ϱ, the point acceleration vector $\ddot{\boldsymbol{u}}$, and the point force density vectors $\boldsymbol{b}^{\mathrm{int}}$ and $\boldsymbol{b}^{\mathrm{ext}}$. Various material formulations of peridynamics exist for the calculation of the internal force density $\boldsymbol{b}^{\mathrm{int}}$, and all of them are based on the nonlocal interactions between material points.

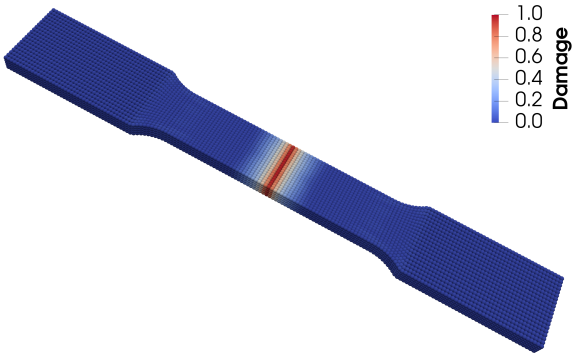

Fig. 2: Fracture simulation of tensile tension test with a crack propagating in the middle of the specimen

The general internal force density for state-based peridynamics is defined as

$$\boldsymbol{b}^{\mathrm{int}}(\boldsymbol{X}, t) = \int_{\mathcal{H}} \boldsymbol{t} - \boldsymbol{t}' \, \mathrm{d}V' \ , \tag{3}$$

with the *force vector states* $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{\Delta X}, t)$ and $\boldsymbol{t}' = \boldsymbol{t}(-\boldsymbol{\Delta X}, t)$. In the first original bond-based formulation of peridynamics, the force vector states $\boldsymbol{t}$ and $\boldsymbol{t}'$ have the same value and opposite direction. In bond-based peridynamics, this implies an intrinsic limitation to only one material parameter and, consequently, restrictions on the Poisson's ratio [17]. To overcome these restrictions, a state-based peridynamics was established. In the ordinary state-based peridynamics, the deformation states of neighboring points also influence the internal force density [17]. This leads to force vector states which are still collinear but not of same value anymore.
Further developments are summarized as non-ordinary state-based peridynamics. A recent development in this regard is continuum-kinematics-inspired peridynamics [7]. Another peridynamic formulation is the local continuum consistent correspondence formulation of non-ordinary state-based peridynamics, where an elastic model from the classical local material theory can be used to calculate the internal force density.
`Peridynamics.jl` is an open source Julia [1] implementation of peridynamics. It can be used to conduct simulations with applications such as crack propagation due to external loading conditions (see Fig. 2), or multi-body contact simulations (see Fig. 1). Users can specify arbitrary geometries as a point cloud to use as a spatial discretization for a simulation. It is also possible to import meshes generated with ABAQUS and convert them into point clouds. Multiple peridynamic material models are implemented and can be used in dynamic simulations using Velocity Verlet integration or in quasi-static simulations with an adaptive dynamic relaxation algorithm [10].

The primary purpose of the package is to provide a framework for peridynamic simulations for a broad user base. A user-friendly interface makes it possible to define geometries and boundary conditions with just a few lines of code. Due to Julia's multiple dispatch functionality, custom material models can be defined to extend the package. Therefore, researchers can use the package to develop new peridynamic material models, simplifying their workflow by utilizing the Julia ecosystem.

2. Statement of need

Various peridynamics software projects exist, including `PeriPy` [2], `NLMech` [9], `Peridigm` [11], and `PeriHub` [18]. `Peridigm` stands out as a mature package capable of large-scale peridynamics simulations that also includes several material models and features. However, the installation process and integrating custom models is notably complex.
Installing and extending `Peridynamics.jl` is straightforward, as it benefits from the Julia ecosystem. Due to its extendability via multiple-dispatch, the package is attractive to researchers developing new peridynamic models. `Peridynamics.jl` was already used in numerous publications [4, 6, 15, 13, 14, 5], which shows the practical usability of the package.
Furthermore, practitioners interested in high-performance computing can also use the default methods of the package for exciting simulations. The package is used for teaching and was already used by several students for their final thesis.

3. Perspectives

In version `v0.2.0`, a multithreading approach is employed. An extension to a hybrid approach with MPI or multithreading is made in version `v0.3.0`. The internals have been thoroughly reworked to incorporate a data structure that can be used for parallel simulations with either MPI or multithreading. These changes also include new material models and a simplified API. For future versions it is planned to add improved material models and to utilize `DifferentialEquations.jl` for the time integration.

4. Acknowledgments

The authors gratefully acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) in the project WE 2525/15-1 and WE 2525/15-2.
The authors gratefully acknowledge the support and the supercomputing resources of the Paderborn Center for Parallel Computing (https://pc2.uni-paderborn.de).

5. References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. doi:10.1137/141000671.

[2] B. Boys, T.J. Dodwell, M. Hobbs, and M. Girolami. Peripy - a high performance opencl peridynamics package. *Computer Methods in Applied Mechanics and Engineering*, 386:114085, 2021. doi:10.1016/j.cma.2021.114085.

[3] Patrick Diehl, Serge Prudhomme, and Martin Lévesque. A Review of Benchmark Experiments for the Validation of Peridynamics Models. *Journal of Peridynamics and Nonlocal Modeling*, 1(1):14–35, 2019. doi:10.1007/s42102-018-0004-x.

[4] Kai Friebertshäuser (Partmann), Marita Thomas, Sven Tornquist, Kerstin Weinberg, and Christian Wieners. Dynamic fracture with a continuum-kinematics-based peridynamic and a phase-field approach. *PAMM*, 22(1):e202200217, 2023. doi:10.1002/pamm.202200217.

[5] Kai Friebertshäuser (Partmann), Marita Thomas, Sven Tornquist, Kerstin Weinberg, and Christian Wieners. Dynamic phase-field fracture in viscoelastic materials using a first-order formulation. *PAMM*, 22(1):e202200249, 2023. doi:10.1002/pamm.202200249.

[6] Kai Friebertshäuser (Partmann), Christian Wieners, and Kerstin Weinberg. Dynamic fracture with continuum-kinematics-based peridynamics. *AIMS Materials Science*, 9(6):791–807, 2022. doi:10.3934/matersci.2022049.

[7] Ali Javili, A. McBride, and Paul Steinmann. Continuum-kinematics-inspired peridynamics. mechanical problems. *Journal of the Mechanics and Physics of Solids*, 131, 07 2019. doi:10.1016/j.jmps.2019.06.016.

[8] Ali Javili, Rico Morasata, Erkan Oterkus, and Selda Oterkus. Peridynamics review. *Mathematics and Mechanics of Solids*, 24(11):3714–3739, 2019. doi:10.1177/1081286518803411.

[9] Prashant K. Jha and Patrick Diehl. Nlmech: Implementation of finite difference/meshfree discretization of nonlocal fracture models. *Journal of Open Source Software*, 6(65):3020, 2021. doi:10.21105/joss.03020.

[10] Bahattin Kilic and Erdogan Madenci. An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics*, 53(3):194–204, 2010. doi:10.1016/j.tafmec.2010.08.001.

[11] David J. Littlewood, Michael L. Parks, John T. Foster, John A. Mitchell, and Patrick Diehl. The peridigm meshfree peridynamics code. *Journal of Peridynamics and Nonlocal Modeling*, 6(1):118–148, 2024. doi:10.1007/s42102-023-00100-0.

[12] Erdogan Madenci and Erkan Oterkus. *Peridynamic Theory and Its Applications*. Springer New York, New York, 2014. doi:10.1007/978-1-4614-8465-3.

[13] Kai Partmann, Manuel Dienst, and Kerstin Weinberg. Peridynamic computations of wave propagation and reflection at material interfaces. *Archive of Applied Mechanics*, July 2024. doi:10.1007/s00419-024-02646-x.

[14] Kai Partmann, Manuel Dienst, Christian Wieners, and Kerstin Weinberg. Peridynamic computations for thin elastic rods. *PAMM*, 2024. doi:10.1002/pamm.202400103.

[15] Kai Partmann, Christian Wieners, and Kerstin Weinberg. Continuum-kinematics-based peridynamics and phase-field approximation of non-local dynamic fracture. *International Journal of Fracture*, 2023. doi:10.1007/s10704-023-00726-7.

[16] S. A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175 – 209, 2000. doi:10.1016/S0022-5096(99)00029-0.

[17] S. A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. Peridynamic States and Constitutive Modeling. *Journal of Elasticity*, 88(2):151–184, 2007. doi:10.1007/s10659-007-9125-1.

[18] Christian Willberg, Jan-Timo Hesse, Marc Garbade, Martin Rädel, Falk Heinecke, Andreas Schuster, and Anna Pernatii. A user material interface for the peridynamic peridigm framework. *SoftwareX*, 21:101322, 2023. doi:10.1016/j.softx.2023.101322.