The Journal of Open Source Software

NLMech: Implementation of finite difference/meshfree discretization of nonlocal fracture models

Prashant K. Jha[*1] **and Patrick Diehl**[†2]

1 Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, United States of America **2** Center for Computation & Technology, Louisiana State University, Baton Rouge, LA, United States of America

DOI: 10.21105/joss.03020

Software

- Review
- Repository
- Archive

Editor: Kevin M. Moerman

Reviewers:

- @vijaysm
- @chennachaos

Submitted: 08 December 2020
Published: 07 June 2021

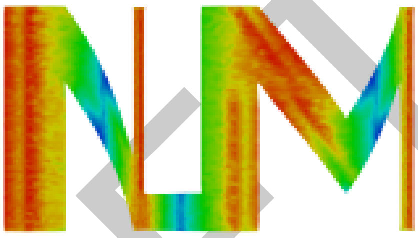

Figure 1: NLMech's logo which shows the obtained damage of a peridynamic simulation.

Summary

The open source code *NLMech* is an implementation of finite difference approximation of nonlocal models, *e.g.* peridynamic. Peridynamic (PD) (Silling et al., 2007; Silling & Askari, 2005) is a nonlocal formulation of classical continuum mechanics that is particularly robust in mechanical deformations involving crack (discontinuous displacement) and damage. The model seamlessly handles the two regime of deformation: elastic/smooth deformation and fracture. The constitutive laws describing the material behavior are simple to conceptualize and implement. Particularly, in numerical implementation, no special care for the modeling of cracks is required. Successful comparison of PD against a variety of experiments has been done (Diehl et al., 2019).

Unlike classical continuum mechanics, where the internal force is written in terms of the stress, in PD, the internal force at a given material point is due to the sum of the pairwise forces of the neighboring points. *i.e.* the force in PD is expressed as the integral of the pairwise force density between the given point and another point in the neighborhood. Neighborhood of point x is typically taken as the ball of radius δ, centered at x, where δ is the nonlocal length scale and is referred to as *horizon*. PD is often divided in two classes: bond-based and state-based models. In bond-based models, the two material points interact via a pairwise force law and the forces between the material points do not depend on the deformation state of surrounding points. In contrast, in the state-based models the volumetric deformation in the neighborhood of two points plays a role in the pariwise force. The governing equation of motion for the bond-based PD (Silling & Askari, 2005) reads as

$$\varrho(\mathbf{X})\ddot{\mathbf{u}}(t,\mathbf{X}) = \int\limits_{B_\delta(\mathbf{X})} \mathbf{f}(\mathbf{u}(t,\mathbf{X}') - \mathbf{u}(t,\mathbf{X}), \mathbf{X}' - \mathbf{X}) d\mathbf{X}' + \mathbf{b}(t,\mathbf{X}) \text{ in } D$$

[*] co-first author
[†] co-first author

and the governing equation for the state-based PD (Silling et al., 2007) reads as

$$\varrho(\mathbf{X})\ddot{\mathbf{u}}(t,\mathbf{X}) = \int_{B_\delta(\mathbf{X})} (T[\mathbf{X},t]\langle \mathbf{X}' - \mathbf{X}\rangle - T[\mathbf{X}',t]\langle \mathbf{X} - \mathbf{X}'\rangle)d\mathbf{X}' + \mathbf{b}(t,\mathbf{X}) \text{ in } D.$$

Here ϱ denotes density of the material, $\mathbf{u}$ displacement field in the material, $\ddot{\mathbf{u}}$ acceleration, and $\mathbf{b}$ external force density. The constitutive law, relating bond strain with bond force, is prescribed using either the pairwise force function $\mathbf{f}$ or the PD state T (Silling et al., 2007). In the library, the following material models are implemented:

- Elastic state-based PD model (Silling et al., 2007),
- Prototype micro-elastic brittle bond-based PD model (Silling & Askari, 2005),
- Nonlinear bond-based PD model (R. Lipton, 2014, 2016), and
- Nonlocal double-well state-based peridynamic model (Jha & Lipton, 2019; R. Lipton et al., 2018b).

For example, regarding input files for these models, we refer to the collection of examples in the documentation.

Currently, the library supports finite difference (or more generally meshfree) discretization. Using the triangulation of arbitrary domain, the library can create a meshfree discretization. The library is equipped with necessary modules, such as FE elements and quadrature integration rules, for finite element discretization of PD. Next, we briefly discuss the finite difference/meshfree discretization of PD. Figure 2 shows the domain D discretized with the nodes $X = \{X_i \in \mathbb{R}^3 | i = 1, \ldots, n\}$. Each node X_i represents a small area/volume denoted by V_i. In PD, as previously mentioned, each point X_i interacts with neighboring points in ball (discrete) $B_\delta(X_i) = \{X_j : |X_i - X_j| < \delta\}$.

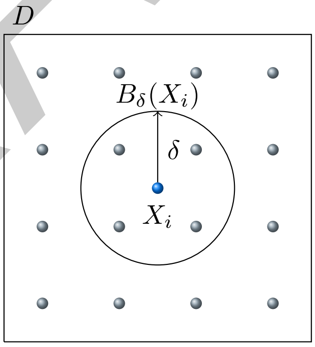

Figure 2: Adapted from (Diehl, 2020).

The discrete equation of motion is written as, for the bond-based PD,

$$\varrho(X_i)\ddot{\mathbf{u}}(t,X_i) = \sum_{j \in B_\delta(X_i)} \mathbf{f}(\mathbf{u}(t,X_j) - \mathbf{u}(t,X_i), X_j - X_i)V_j + \mathbf{b}(t,X_i) \text{ in } D,$$

and, the state-based PD,

$$\varrho(X_i)\ddot{\mathbf{u}}(t,X_i) = \sum_{j \in B_\delta(X_i)} (T[X_i,t]\langle X_j - X_i\rangle - T[X_j,t]\langle X_i - X_j\rangle)V_j + \mathbf{b}(t,X_i) \text{ in } D.$$

Here $\mathbf{u}(t,X_i)$ denotes the displacement of node X_i at time $0 \leq t \leq T$. For the time discretization, we can consider: *1)* implicit time integration and *2)* explicit time integration using either central difference or velocity verlet scheme.

Software Implementation and Applications

NLMech relies on the following open source softwares: HPX (Kaiser et al., 2020), Blaze (Iglberger et al., 2012), Blaze_Iterative, Gmsh (Geuzaine & Remacle, 2009), VTK (Schroeder et al., 2004), and yaml-cpp. For details about the specific version, we refer to NLMech's documentation.

NLMech was used for the following applications/publications:

- Numerical convergence of finite difference approximations for state based perdidynamic fracture models (Jha & Lipton, 2019)
- Complex fracture nucleation and evolution with nonlocal elastodynamics (R. P. Lipton et al., 2019)
- Free damage propagation with memory (R. Lipton et al., 2018a)
- Kinetic relations and local energy balance for linear elastic fracture mechanics from a nonlocal peridynamic model (Jha & Lipton, 2020)

For an updated list of applications/publications, we refer to corresponding NLMech documentation.

Statement of need

Nonlocal models, like peridynamic, are computationally expensive. Several publications on GPU-based implementations (Diehl, 2012; Diehl & Schweitzer, 2015; Mossaiby et al., 2017) and one commercial implementation in LS-DYNA (Ren et al., 2017) can be found in literature. However, from an open source perspective, only two other peridynamic implementations are available: Peridigm (Littlewood, 2015) and PDLammps (Parks et al., 2008). Both of these codes rely on the Message Passing Interface (MPI). On modern supercomputers' many core architectures where the threads per computational node increase, it is more and more important to focus on the fine-grain parallelism with increasing cores per computational nodes. NLMech is based on the C++ standard library for parallelism and concurrency (HPX) (Kaiser et al., 2020). For more details on utilization of asynchronous many-task systems, we refer to Diehl et al. (2020). The library implements the experimental nonlinear bond-based and state-based models, and the process of adding new material models is simple following the existing templates.

Future directions

We are interested in extending/improving the library with

- implementation of new material models,
- higher order time discretization schemes,
- local-nonlocal coupling methods, and
- further optimization of nonlocal computation.

If you are interested in contributing, please read our guidelines and our code of conduct before doing so.

Acknowledgments

NLMech has been funded by:

- Army Research Office Grant # W911NF-16-1-0456 to PI Dr. Robert Lipton (Professor at Louisiana State University). This grant supported Prashant K. Jha on a postdoctoral position from October 2016 - July 2019.
- Canada Research Chairs Program under the Canada Research Chair in Multiscale Modelling of Advanced Aerospace Materials held by M. Lévesque; Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants Program under Discovery Grant RGPIN-2016-06412.
- We are grateful for the support of the Google Summer of Code program funding internships.

For a updated list of previous and current funding, we refer to the corresponding NLMech website.

References

Diehl, P. (2020). *EMU-nodal discretization*. https://doi.org/10.6084/m9.figshare.12320726.v1

Diehl, P. (2012). *Implementierung eines peridynamik-verfahrens auf GPU* [Master's thesis]. https://doi.org/10.18419/opus-2879

Diehl, P., Jha, P. K., Kaiser, H., Lipton, R., & Lévesque, M. (2020). An asynchronous and task-based implementation of peridynamics utilizing HPX—the c++ standard library for parallelism and concurrency. *SN Applied Sciences*, *2*(12), 1–21. https://doi.org/10.1007/s42452-020-03784-x

Diehl, P., Prudhomme, S., & Lévesque, M. (2019). A review of benchmark experiments for the validation of peridynamics models. *Journal of Peridynamics and Nonlocal Modeling*, *1*(1), 14–35. https://doi.org/10.1007/s42102-018-0004-x

Diehl, P., & Schweitzer, M. A. (2015). Efficient neighbor search for particle methods on GPUs. In *Meshfree methods for partial differential equations VII* (pp. 81–95). Springer. https://doi.org/10.1007/978-3-319-06898-5_5

Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, *79*(11), 1309–1331. https://doi.org/10.1002/nme.2579

Iglberger, K., Hager, G., Treibig, J., & Rüde, U. (2012). High performance smart expression template math libraries. *2012 International Conference on High Performance Computing & Simulation (HPCS)*, 367–373. https://doi.org/10.1109/hpcsim.2012.6266939

Jha, P. K., & Lipton, R. (2019). Numerical convergence of finite difference approximations for state based peridynamic fracture models. *Computer Methods in Applied Mechanics and Engineering*, *351*, 184–225. https://doi.org/10.1016/j.cma.2019.03.024

Jha, P. K., & Lipton, R. P. (2020). Kinetic relations and local energy balance for LEFM from a nonlocal peridynamic model. *International Journal of Fracture*, 1–15. https://doi.org/10.1007/s10704-020-00480-0

Kaiser, H., Diehl, P., Lemoine, A. S., Lelbach, B. A., Amini, P., Berge, A., Biddiscombe, J., Brandt, S. R., Gupta, N., Heller, T., Huck, K., Khatami, Z., Kheirkhahan, A., Reverdell, A., Shirzad, S., Simberg, M., Wagle, B., Wei, W., & Zhang, T. (2020). HPX - the c++

standard library for parallelism and concurrency. *Journal of Open Source Software*, *5*(53), 2352. https://doi.org/10.21105/joss.02352

Lipton, R. (2014). Dynamic brittle fracture as a small horizon limit of peridynamics. *Journal of Elasticity*, *117*(1), 21–50. https://doi.org/10.1007/s10659-013-9463-0

Lipton, R. (2016). Cohesive dynamics and brittle fracture. *Journal of Elasticity*, *124*(2), 143–191. https://doi.org/10.1007/s10659-015-9564-z

Lipton, R. P., Lehoucq, R. B., & Jha, P. K. (2019). Complex fracture nucleation and evolution with nonlocal elastodynamics. *Journal of Peridynamics and Nonlocal Modeling*, *1*(2), 122–130. https://doi.org/10.1007/s42102-019-00010-0

Lipton, R., Said, E., & Jha, P. (2018a). Free damage propagation with memory. *Journal of Elasticity*, *133*(2), 129–153. https://doi.org/10.1007/s10659-018-9672-7

Lipton, R., Said, E., & Jha, P. K. (2018b). Dynamic brittle fracture from nonlocal double-well potentials: A state-based model. In G. Z. Voyiadjis (Ed.), *Handbook of nonlocal continuum mechanics for materials and structures* (pp. 1–27). Springer International Publishing. https://doi.org/10.1007/978-3-319-22977-5_33-1

Littlewood, D. J. (2015). Roadmap for peridynamic software implementation. *SAND Report, Sandia National Laboratories, Albuquerque, NM and Livermore, CA*. https://doi.org/10.2172/1226115

Mossaiby, F., Shojaei, A., Zaccariotto, M., & Galvanetto, U. (2017). OpenCL implementation of a high performance 3D peridynamic model on graphics accelerators. *Computers & Mathematics with Applications*, *74*(8), 1856–1870. https://doi.org/10.1016/j.camwa.2017.06.045

Parks, M. L., Lehoucq, R. B., Plimpton, S. J., & Silling, S. A. (2008). Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications*, *179*(11), 777–783. https://doi.org/10.1016/j.cpc.2008.06.011

Ren, B., Wu, C., & Askari, E. (2017). A 3D discontinuous galerkin finite element method with the bond-based peridynamics model for dynamic brittle failure analysis. *International Journal of Impact Engineering*, *99*, 14–25. https://doi.org/10.1016/j.ijimpeng.2016.09.003

Schroeder, W. J., Lorensen, B., & Martin, K. (2004). *The visualization toolkit: An object-oriented approach to 3D graphics*. Kitware.

Silling, S. A., & Askari, E. (2005). A meshfree method based on the peridynamic model of solid mechanics. *Computers & Structures*, *83*(17-18), 1526–1535. https://doi.org/10.1016/j.compstruc.2004.11.026

Silling, S. A., Epton, M., Weckner, O., Xu, J., & Askari. (2007). Peridynamic states and constitutive modeling. *Journal of Elasticity*, *88*(2), 151–184. https://doi.org/10.1007/s10659-007-9125-1