Peridynamics with LAMMPS:  
A User Guide

Michael L. Parks *  Steven J. Plimpton †
Richard B. Lehoucq *  Stewart A. Silling ‡

*Computational Mathematics and Algorithms
†Scalable Algorithms
‡Multiscale Dynamic Material Modeling
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1320

Abstract

Peridynamics is a nonlocal formulation of continuum mechanics. The discrete peridynamic model has the same computational structure as a molecular dynamic model. This document details the implementation of a discrete peridynamic model within the LAMMPS molecular dynamic code.

This document provides a brief overview of the peridynamic model of a continuum, then discusses how the peridynamic model is discretized, and overviews the LAMMPS implementation. A nontrivial example problem is also included.
Contents

1 Introduction ................................................................. 9
  1.1 Typographical Conventions ............................................ 9

2 Getting Started .............................................................. 10
  2.1 Building the Peridynamic Module Within LAMMPS ..................... 10
  2.2 Input Script Basics ..................................................... 10
  2.3 Restrictions .......................................................... 12

3 Peridynamic Model of a Continuum ...................................... 13
  3.1 Basic Notation and Newton’s Second Law ............................... 13
  3.2 Proportional Microelastic Materials .................................... 13
  3.3 Damage ............................................................. 14

4 Discrete Peridynamic Model and LAMMPS Implementation .......... 15
  4.1 Newton’s Second Law and the Spatial Discretization .................. 15
  4.2 Short-Range Forces .................................................... 15
  4.3 Modification to the Particle Volume ................................... 16
  4.4 Discrete Equation of Motion .......................................... 16
  4.5 Breaking Bonds ....................................................... 17
  4.6 PseudoCode ......................................................... 18

5 A Numerical Example ...................................................... 19
  5.1 Problem Description and Setup ....................................... 19
  5.2 The Projectile ....................................................... 19
  5.3 Writing the LAMMPS Input File ....................................... 20
  5.4 Numerical Results and Discussion .................................... 20

References ................................................................. 23
Figures

1  Diagram showing horizon of a particular particle, demonstrating that the volume associated with particles near the boundary of the horizon are not completely contained within the horizon. ................................................................. 17
2  Target during and after impact. ................................................................. 22
Tables

1 Notational conventions. ................................................................. 9
1 Introduction

This document is organized as follows. In section 2 we discuss how to build the peridynamic module within LAMMPS, and discuss basic requirements for input scripts to use the peridynamic module. In section 3 we overview the relevant portions of the peridynamic model of a continuum material. In section 4 we discuss the discretization of the PD model and its LAMMPS implementation. Finally, in section 5, we discuss a LAMMPS simulation of a specific numerical experiment described in [6].

1.1 Typographical Conventions

Our typographical conventions are found in Table 1.

Table 1. Notational conventions.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbatim text</td>
<td>make</td>
<td>Text to be typed at your command prompt</td>
</tr>
<tr>
<td>&lt;text in angle brackets&gt;</td>
<td>&lt;your platform&gt;</td>
<td>User specified statement</td>
</tr>
</tbody>
</table>

Finally, note all norms \(\|\cdot\|\) are taken to be the 2-norm, \(\|\cdot\|_2\).
2 Getting Started

In this section, we assume that you already have a working LAMMPS installation. For more on downloading and building LAMMPS, see http://lammps.sandia.gov. This document only provides information related to the peridynamic module within LAMMPS. For questions regarding the usage of LAMMPS, please see the LAMMPS documentation.

2.1 Building the Peridynamic Module Within LAMMPS

In the LAMMPS distribution, the peridynamic model is distributed as an add-on module, which means that it is not by default compiled with the rest of LAMMPS. To instruct LAMMPS to build the peridynamic module, go to the LAMMPS source subdirectory (/src) and type

```
make yes-peri
```

followed by

```
make <your platform>
```

to compile LAMMPS on your particular platform.

2.2 Input Script Basics

Here we provide a listing of commands that must be included in a LAMMPS input script to utilize the peridynamic module. These commands assume knowledge of the peridynamic PMB model (section 3) and its discretization (section 4). This is not an inclusive list of LAMMPS commands. For a complete example script, see section 5.

LAMMPS has been modified to support SI units. Your LAMMPS input script should contain the command

```
units si
```

All quantities specified in the input script and data file, as well as quantities output to the screen, log file, and dump files will be in SI units.

Only a simple cubic lattice is currently supported. Your LAMMPS input script should contain the command

```
lattice sc <lattice constant>
```
A peridynamic simulation requires the “peri” atom style be used. Your input script should contain the command

    atom style peri

An associated required command tells LAMMPS to create a data structure used to index particles. Your input script should contain the command

    atom_modify map array

A peridynamic simulation also requires the “peri” pair style be used. Your input script should contain the command

    pair_style peri

to invoke the “peri” pair style, and the command

    pair_coeff <type 1> <type 2> <c> <delta> <α><β> <γ>

to define the arguments for the pairwise force of the PMB model. See section 3 for more on the PMB model.

The mass density and volume fraction for each particle must be defined. Your input script should contain the commands

    set group all rmass <ρ>
    set group all vfrac <V_i>

In LAMMPS, the density of a discrete perdynamic particle is stored in the variable normally reserved to store the mass of an atom. However, in the first line, you are setting the density \( \rho \) of all peridynamic particles, not the mass. In the second line, you are setting the volume of each peridynamic particle. For a simple cubic lattice, the volume should be equal to the cube of the lattice constant, i.e., \( V_i = \Delta^3 \).

If you wish to start a simulation with the velocity of the peridynamic particles set to zero, your input script should contain the command

    velocity all set 0.0 0.0 0.0 sum no units box

For a peridynamic simulation, we use a constant NVE integrator sampling from the microcanonical ensemble, since temperature is an ill-defined quantity for macroscopic PD particles and thus thermostatting (as in a constant NVT integration) is not needed. To use a constant NVE integrator, your input script should contain the command

    fix <fix id> all nve
2.3 Restrictions

LAMMPS operates in parallel in a spatial-decomposition mode [4], where each processor owns a spatial subdomain of the overall simulation domain and communicates with its neighboring processors via distributed-memory message passing (MPI) [7] to acquire ghost atom information to allow forces on the atoms it owns to be computed. LAMMPS also uses Verlet neighbor lists which are recomputed every few timesteps as particles move. On these timesteps, particles also migrate to new processors as needed.

When migrating particles to a new processor, LAMMPS was constructed under the assumption that particles always move slowly enough that they never cross the spatial subdomain owned by a processor in a single timestep. Should this occur, LAMMPS will hang. You should ensure your particles are moving slowly enough (or that your timestep small enough) so that this does not occur.
The following is not a complete overview of peridynamics, but a discussion of only those details specific to the model we have implemented within LAMMPS. To begin, it is useful to define the notation we will use.

3.1 Basic Notation and Newton’s Second Law

Within the peridynamic literature, the following notational conventions are generally used. The position of a given particle in the reference configuration is \( x \). The displacement of the particle at \( x \) in the reference configuration at some time \( t \) is denoted \( u(x, t) \). The position of the particle at \( x \) in the reference configuration at some time \( t \) is denoted \( y(x, t) = x + u(x, t) \). Given a two particles with positions \( x \) and \( x' \) in the reference configuration, we denote the interparticle distance in the reference configuration as \( \xi = x' - x \). We denote the relative displacement at some time \( t \) as \( \eta = u(x', t) - u(x, t) \). We note here that \( \eta \) is time-dependent, and that \( \xi \) is not. The acceleration of any particle at position \( x \) in the reference configuration at time \( t \) is written as

\[
\rho \ddot{u}(x, t) = \int_{\mathcal{H}_x} f(\eta, \xi) \, dV_{x'} + b(x, t) \tag{3.1}
\]

where \( \mathcal{H}_x \) is a neighborhood of \( x \), \( \rho \) is a mass density in the reference configuration, and \( b \) is a prescribed body force density field. \( f \) is a pairwise force function whose value is the force vector that particle \( x' \) exerts on \( x \), and has units force/volume\(^2\). We assume that each material has associated with it a positive scalar \( \delta \), called the horizon, such that if \( \|\xi\| > \delta \), then \( f(\eta, \xi) = 0 \), \( \forall \eta \).

The pairwise force function \( f \) can be written as

\[
f(\eta, \xi) = f(\eta, \xi) \frac{\eta + \xi}{\|\eta + \xi\|},
\]

where \( f \) is a scalar-valued function. We observe here that the distance between two particles is always \( \|y' - y\| = \|\eta + \xi\| \). We also see that \( f \) always acts along a line connecting the two particles, as we expect.

3.2 Proportional Microelastic Materials

In a proportional microelastic material [6] the bond force varies linearly with the bond stretch. We assume that the scalar bond force \( f \) depends on \( \eta \) only through the bond stretch, defined as

\[
s(t, \eta, \xi) = \frac{\|\eta + \xi\| - \|\xi\|}{\|\xi\|}.
\]

Bond stretch is a unitless quantity, and identical to a one-dimensional definition of strain. As such, we see that a bond at its equilibrium length has stretch \( s = 0 \), and a bond at twice its equilibrium length has stretch \( s = 1 \).
3.3 Damage

Bonds are made to break when they are stretched beyond a given limit. Once a bond fails, it is failed forever [6]. Further, new bonds are never created during the course of a simulation. We will consider only \( f \) corresponding to a *prototype microelastic brittle* (PMB) material [6], so that \( f \) can be written as

\[
f(\eta, \xi) = g(s(t, \eta, \xi)) \cdot \mu(t, \eta, \xi)
\]

where \( g \) is a linear scalar-valued function given by

\[
g(s(t, \eta, \xi)) = \begin{cases} c \cdot s(t, \eta, \xi) & \text{if } ||\xi|| \leq \delta \\ 0 & \text{otherwise} \end{cases}
\]

where \( c \) is a constant of the form\

\[
c = \frac{18k}{\pi \delta^4},
\]

where \( \delta \) is the horizon, and \( k \) is the bulk modulus of the material, and \( \mu \) is the history-dependent scalar boolean function

\[
\mu(t, \eta, \xi) = \begin{cases} 1 & \text{if } s(t', \eta, \xi) < \min(s_0(t', \eta, \xi), s_0(t', \eta', \xi')) \text{ for all } 0 \leq t' \leq t \\ 0 & \text{otherwise} \end{cases}
\]

where \( \eta' = u(x'', t) - u(x', t) \) and \( \xi' = x'' - x' \). Here, \( s_0(t, \eta, \xi) \) is a *critical stretch* defined as

\[
s_0(t, \eta, \xi) = s_{00} - \alpha s_{\min}(t, \eta, \xi), \quad s_{\min}(t) = \min_{\xi} s(t, \eta, \xi),
\]

where \( s_{00} \) and \( \alpha \) are material-dependant constants. The history function \( \mu \) breaks bonds when the stretch \( s \) exceeds the critical stretch \( s_0 \).

Although \( s_0(t, \eta, \xi) \) is expressed as a property of a particle, bond breaking must be a symmetric operation for all particle pairs sharing a bond. That is, particles \( x \) and \( x' \) must utilize the same test when deciding to break their common bond. This can be done by any method that treats the particles symmetrically. In the definition of \( \mu \) above, we have chosen to take the minimum of the two \( s_0 \) values for particles \( x \) and \( x' \) when determining if the \( x-x' \) bond should be broken.

---

1This is for a three-dimensional model. \( c \) is different for two- and one-dimensional models. (c.f. [2]).
4 Discrete Peridynamic Model and LAMMPS Implementation

In LAMMPS, instead of (3.1), we model this equation of motion:

$$\rho \ddot{y}(x, t) = \int_{H(x)} f(\eta, \xi) \, dV_{\xi} + b(x, t),$$

where we explicitly track and store at each timestep the positions and not the displacements of the particles. We observe that \(\ddot{y}(x, t) = \ddot{x} + \ddot{u}(x, t) = \ddot{u}(x, t)\), so that this is equivalent to (3.1).

4.1 Newton’s Second Law and the Spatial Discretization

The region defining a peridynamic material is discretized into particles forming a simple cubic lattice with lattice constant \(\Delta x\), where each particle \(i\) is associated with some volume fraction \(V_i\). For any particle \(i\), let \(F_i\) denote the family of particles for which particle \(i\) shares a bond in the reference configuration. That is,

$$F_i = \{ p \mid \|x_p - x_i\| \leq \delta \}.$$

The discretized equation of motion replaces (3.1) with

$$\rho \ddot{y}_i^n = \sum_{p \in F_i} f(u^n_p - u^n_i, x_p - x_i)V_p + b^n_i \quad (4.1)$$

with where \(f\) is given in (3.2), \(n\) is the timestep number and subscripts denote the particle number, so that \(u^n_i = u(x_i, t_0 + n\Delta t)\).

4.2 Short-Range Forces

In the model discussed so far, particles interact only through their bond forces. A particle with no bonds becomes a free non-interacting particle. To account for contact forces, short-range forces are introduced [5]. We add to the force \(f\) in (4.1) the following force

$$f_S(y_p, y_i) = \min\{0, \frac{c_S}{\delta}(\|y_p - y_i\| - d_{pi})\} \frac{y_p - y_i}{\|y_p - y_i\|}, \quad (4.2)$$

where \(d_{pi}\) is the short-range interaction distance between particles \(p\) and \(i\), and \(c_S\) is a multiple of the constant \(c\) from (3.3). Note that the short-range force is always repulsive, never attractive. In practice, we choose

$$c_S = 15c. \quad (4.3)$$

For the short-range interaction distance, we choose [5]

$$d_{pi} = \min \{0.9 \|x_p - x_i\|, 1.35(r_p + r_i)\}, \quad (4.4)$$
where \( r_i \) is called the *node radius* of particle \( i \). Given a discrete lattice, we choose \( r_i \) to be half the lattice constant.\(^2\) Given this definition of \( d_{pi} \), contact forces appear only when particles are under compression.

When accounting for short-range forces, it is convenient to define the short-range family of particles
\[
\mathcal{F}_i^S = \{ p \mid \| y_p - y_i \| \leq d_{pi} \}.
\]

### 4.3 Modification to the Particle Volume

In a situation where two particles share a bond with \( \| x_p - x_i \| = \delta \), for example, we suppose that only approximately half the volume of each particle is “seen” by the other [5]. When computing the force of each particle on the other we use \( V_p/2 \) rather than \( V_p \) in (4.1). As such, we introduce a nodal volume scaling function for all bonded particles where \( \delta - r_S \leq \| x_p - x_i \| \leq \delta \).

We choose to use a linear unitless nodal volume scaling function
\[
\nu(x_p - x_i) = \begin{cases} 
-\frac{1}{2r_S} \| x_p - x_i \| + \left( \frac{\delta}{2r_S} + \frac{1}{2} \right) & \text{if } \delta - r_S \leq \| x_p - x_i \| \leq \delta \\
1 & \text{if } \| x_p - x_i \| \leq \delta - r_S \\
0 & \text{otherwise}
\end{cases}
\]

If \( \| x_p - x_i \| = \delta \), \( \nu = 0.5 \), and if \( \| x_p - x_i \| = \delta - r_S \), \( \nu = 1.0 \), for example.

### 4.4 Discrete Equation of Motion

The semi-discrete equation of motion can be written as
\[
\rho \ddot{y}_i^n = c \sum_{p \in \mathcal{F}_i} \left( \frac{\| y_p - y_i \| - \| x_p - x_i \|}{\| x_p - x_i \|} \right) \mu(t, \eta, \xi) \nu(x_p - x_i) V_p \left( \frac{y_p - y_i}{\| y_p - y_i \|} \right) + \sum_{p \in \mathcal{F}_i^S} \min\{0, \frac{cs}{\delta} (\| y_p - y_i \| - d_{ip}) \} V_p \left( \frac{y_p - y_i}{\| y_p - y_i \|} \right) + b_i^n,
\]
accounting for short-range forces and nodal volume scaling.

When discretizing time in LAMMPS, we instead use a velocity-Verlet scheme, where both the position and velocity of the particle are stored explicitly. The velocity-Verlet scheme is generally expressed in three steps, as where \( m \) denotes the mass of a particle, and \( \vec{f}_i^n \) is the net force on particle \( i \) at timestep \( n \).

\(^2\)For a simple cubic lattice, \( \Delta x = \Delta y = \Delta z \).
(a) Two-dimensional diagram showing particle on mesh (solid lines) with horizon $\delta$ as grey circular region. Dual mesh (dotted lines) shows boundaries of each particle.

(b) Plot of $\nu(x_p - x_i)$ vs. $\|x_p - x_i\|$. 

Figure 1. Diagram showing horizon of a particular particle, demonstrating that the volume associated with particles near the boundary of the horizon are not completely contained within the horizon.

Algorithm 1 Velocity Verlet

1: $v_{i}^{n+1/2} = v_{i}^{n} + \frac{\Delta t f_{i}^{n}}{2m}$
2: $y_{i}^{n+1} = y_{i}^{n} + \Delta t v_{i}^{n+1/2}$
3: $v_{i}^{n+1} = v_{i}^{n+1/2} + \frac{\Delta t f_{i}^{n+1}}{2m}$

4.5 Breaking Bonds

During the course of simulation, it may be necessary to break bonds, as described in section 3.3. A naïve implementation would have us first loop over all bonds and compute $s_{\text{min}}$ in (3.5), then loop over all bonds again and break bonds with a stretch $s > s_{0}$ as in (3.4), and finally loop over all particles compute forces for the next step of Algorithm 1. For reasons of computational efficiency, we will utilize the values of $s_{0}$ from the previous timestep when deciding to break a bond. For the first timestep, $s_{0}$ is initialized to $\infty$ for all nodes. This means that no bonds may be broken until the second timestep. As such, it is recommended that the first few timesteps of the peridynamic simulation not involve any actions that might result in the breaking of bonds. As a practical example, the hard sphere in the next section is placed such that it does not impact the brittle plate until 1000 timesteps into the simulation.
4.6 PseudoCode

A sketch of the peridynamic implementation in LAMMPS appears in Algorithm 2.

Algorithm 2 PMB Peridynamic Model in LAMMPS

1: Fix \( s_{00}, \alpha, \) horizon \( \delta, \) spring constant \( c, \) timestep \( \Delta t, \) and generate initial lattice of particles with lattice constant \( \Delta x. \) Let there be \( N \) particles.
2: Initialize bonds between all particles where \( \| \mathbf{x} - \mathbf{x}' \| \leq \delta. \)
3: Initialize \( s_0 = \infty \) \{Initialize each entry to MAX\_DOUBLE.\}
4: while not done do
5: Perform step 1 of Algorithm 1, updating velocities of all particles.
6: Perform step 2 of Algorithm 1, updating positions of all particles.
7: \( s_0 = \infty \) \{Initialize each entry to MAX\_DOUBLE.\}
8: for \( i = 1 \) to \( N \) do
9: \{Compute short-range forces\}
10: for all particles \( k \in \mathcal{F}_S^i \) (the short-range family of nodes for particle \( i \)) do
11: \( r = \| \mathbf{y}_i - \mathbf{y}_k \|. \)
12: \( dr = \min\{0, r - d\}. \) \{Short-range forces are only repulsive, never attractive\}
13: \( k = \frac{c}{r^2} \cdot V_k \cdot dr. \) \{\( c \) defined in (4.3)\}
14: \( f = f - k \frac{\mathbf{y}_i - \mathbf{y}_k}{\| \mathbf{y}_i - \mathbf{y}_k \|}. \)
15: end for
16: end for
17: for \( i = 1 \) to \( N \) do
18: \{Compute bond forces.\}
19: for all particles \( k \) sharing a bond with particle \( i \) do
20: \( r = \| \mathbf{y}_i - \mathbf{y}_k \|. \)
21: \( dr = r - \| \mathbf{x}_i - \mathbf{x}_k \|. \)
22: \( k = \frac{c}{\| \mathbf{x}_i - \mathbf{x}_k \|} \cdot V_k \cdot \nu(\mathbf{x}_i - \mathbf{x}_k)dr. \) \{\( c \) defined in (3.3)\}
23: \( f = f - k \frac{\mathbf{y}_i - \mathbf{y}_k}{\| \mathbf{y}_i - \mathbf{y}_k \|}. \)
24: if \( \frac{dr}{\| \mathbf{x}_i - \mathbf{x}_k \|} > \min(s_0(i), s_0(k)) \) then
25: Break \( i \)'s bond with \( k. \) \{\( k \)'s bond with \( i \) will be broken when this loop iterates on \( k \}\)
26: end if
27: \( s_0(i) = \min(s_0(i), s_0 - \alpha \frac{dr}{\| \mathbf{x}_i - \mathbf{x}_k \|}). \)
28: end for
29: end for
30: \( s_0 = \tilde{s}_0. \) \{Store for use in next timestep.\}
31: Perform step 3 of Algorithm 1, updating velocities of all particles.
32: end while
5 A Numerical Example

To introduce the peridynamic implementation within LAMMPS, we replicate a numerical experiment taken from section 6 of [6].

5.1 Problem Description and Setup

We consider the impact of a rigid sphere on a homogeneous block of brittle material. The sphere has diameter 0.01 m and velocity 100 m/s directed normal to the surface of the target. The target material has density $\rho = 2200$ kg/m$^3$. A PMB material model is used with $k = 14.9$ GPa and critical bond stretch parameters given by $s_0 = 0.0005$ and $\alpha = 0.25$. A three-dimensional simple cubic lattice is constructed with lattice constant 0.0005 m and horizon 0.0015 m. (The horizon is three times the lattice constant.) The target is a cylinder of diameter 0.074 m and thickness 0.0025 m, and the associated lattice contains 103,110 particles. Each particle $i$ has volume fraction $V_i = 1.25 \times 10^{-10}$ m$^3$.

The spring constant in the PMB material model is

$$c = \frac{18k}{\pi \delta^4} = \frac{18(14.9 \times 10^9)}{\pi (1.5 \times 10^{-3})^4} \approx 1.6863 \times 10^{22}. \quad (5.1)$$

The CFL analysis from [6] shows that a timestep of $1.0 \times 10^{-9}$ is safe. Note that this calculation ignores the short-range interactions; the true timestep limitation should be smaller than this due to the larger spring constant present in short-range interactions.

We observe here that in IEEE double-precision floating point arithmetic when computing the bond stretch $s(t, \eta, \xi)$ at each iteration where $\|\eta + \xi\|$ is computed during the iteration and $\|\xi\|$ was computed and stored for the initial lattice, it may be that $fl(s) = \varepsilon$ with $|\varepsilon| \leq \varepsilon_{machine}$ for an unstretched bond. Taking $\varepsilon = 2.220446049250313 \times 10^{-16}$, we see that the value $c \cdot s \cdot V_i \approx 4.68 \times 10^{-4}$, computed when determining $f$, is perhaps larger than we would like, especially when the true force should be zero. One simple way to avoid this issue is to insert the following instructions in Algorithm 2 after instruction 21:

1: if $|dr| < \varepsilon_{machine}$ then
2: $dr = 0$.
3: end if

Qualitatively, this says that displacements on the order of $10^{-6}$Å are taken to be exactly zero, a seemingly reasonable assumption.

5.2 The Projectile

The projectile used in the following experiments is not the one used in [6]. The projectile used here exerts a force

$$F(r) = -k_s(r - R)^2$$
on each atom where $k_s$ is a specified force constant, $r$ is the distance from the atom to the center of the indenter, and $R$ is the radius of the projectile. The force is repulsive and $F(r) = 0$ for $r > R$. For our problem, the projectile radius $R = 0.05$ m, and we have chosen $k_s = 1.0 \times 10^{17}$ (compare with (5.1) above).

### 5.3 Writing the LAMMPS Input File

We discuss the example input script from Algorithm 3. In line 3 we specify that SI units are to be used. We specify the dimension (3) and boundary conditions (“shrink-wrapped”) for the computational domain in lines 4 and 5. In line 6 we specify that peridynamic particles are to be used for this simulation. In line 8, we set the “skin” distance used in building the LAMMPS neighborlist, used when computing short-range forces. In line 9 we set the lattice constant (in meters) and in line 11 we define the spatial region where the target will be placed. In line 13 we specify a rectangular box enclosing the target region that defines the simulation domain. Line 15 fills the target region with atoms. Lines 16 and 17 define the peridynamic pairwise force function, and lines 19 and 21 set the particle density and particle volume, respectively. The particle volume should be set to the cube of the lattice constant for a simple cubic lattice. Line 25 instructs LAMMPS to integrate time with velocity-Verlet, and line 27 creates the spherical projectile, sending it with a velocity of 100 m/s towards the target. Line 28 sets the timestep, line 29 instructs LAMMPS to provide a screen dump of thermodynamic quantities every 200 timesteps, and line 30 instructs LAMMPS to create a data file (dump.output) with a complete snapshot of the system every 1000 timesteps. This file can be used to create still images or movies. Finally, line 31 instructs LAMMPS to run for 200,000 timesteps.

### 5.4 Numerical Results and Discussion

We ran the input script from Algorithm 3. Images of the disk (projectile not shown) appear in Figure 2. Visualization was done with the EnSight visualization package [1]. The LAMMPS dump file was converted to an EnSight format with the pizza.py toolkit [3].

The symmetry in the computed solution arises because a perfect lattice was used, and a because a perfectly spherical projectile impacted the lattice at its geometric center. To break the symmetry in the solution, the nodes in the peridynamic body must be perturbed slightly from the lattice sites.
Algorithm 3 Example LAMMPS Input Script

1: # 3D Peridynamic simulation with indenter
2: # Use SI units
3: units       si
4: dimension   3
5: boundary    s s s
6: atom_style  peri
7: atom_modify map array
8: neighbor    0.0006 bin
9: lattice     sc 0.0005
10: # Create desired target
11: region     target cylinder y 0.0 0.0 0.037 -0.0025 0.0 units box
12: # Make 1 atom types
13: create_box 1 region target
14: # Create the atoms in the simulation region
15: create_atoms 1 target
16: pair_style  peri
17: pair_coeff  * * 1.6863e22 0.0015 0.0005 0.25
18: # Set mass density
19: set         group all rmass 2200
20: # vfrac = lattice constant^2
21: set         group all vfrac 1.25e-10
22: # Zero out velocities of particles
23: velocity    all set 0.0 0.0 0.0 sum no units box
24: # Use velocity-Verlet time integrator
25: fix         F1 all nve
26: # Construct spherical nanointenter to shatter target
27: fix         F2 all indent 1e17 sphere 0.0 0.0051 0.0 0.005 vel 0.0 -100.0 0.0 units box
28: timestep    1.0e-9
29: thermo      200
30: dump        D1 all atom 1000 dump.output
31: run         200000
Figure 2. Target during and after impact.

(a) Cut view of target during impact.

(b) Top monolayer showing fragmentation (center debris removed).
References


DISTRIBUTION:

1  MS 1322  John Aidun, 1435
1  MS 1318  Ken Alvin, 1416
1  MS 1320  Scott Collis, 1414
1  MS 1320  Richard Lehoucq, 1414
1  MS 1320  Michael Parks, 1414
1  MS 1316  Steven Plimpton, 1416
1  MS 1322  Stewart Silling, 1435
1  MS 0899  Technical Library, 9536 (electronic copy)
1  MS 0123  D. Chavez, LDRD Office, 1011