SANDIA REPORT

2008-0135
Unlimited Release
Printed July 2008

Peridynamics with LAMMPS: A User Guide v0.1 Beta

Michael L. Parks, Pablo Seleson, Steven J. Plimpton, Richard B. Lehoucq, and Stewart A. Silling

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

2008-0135
Unlimited Release
Printed July 2008

Peridynamics with LAMMPS: A User Guide v0.1 Beta

Michael L. Parks [*] Pablo Seleson [§]
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

[§]Dept. of Scientific Computing
400 Dirac Science Library
Florida State University
Tallahassee, FL 32306-4120

Abstract

Peridynamics is a nonlocal formulation of continuum mechanics. The discrete peridynamic model has the same computational structure as a molecular dynamics model. 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. An example problem is also included.

Contents

Figures

Tables

1 Introduction

This document details the implementation of a discrete peridynamic model within the LAMMPS molecular dynamic code.

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.

Notation	Example	Description
`Verbatim text`	`make`	Text to be typed at your command prompt
<text in angle brackets>	<your platform>	User specified statement

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 installlation. 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. To use 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
```

The “skin” distance used when computing neighborlists should be defined appropriately for your choice of simulation parameters. Your input script should contain the command

```
neighbor <skin> bin
```

where the “skin” should be set to a value such that the peridynamic horizon plus the skin distance is larger than the maximum possible distance between two bonded particles (before their bond breaks). A peridynamic simulation also requires a peridynamic pair style be used. Your input script should contain the command

```
pair_style peri/pmb
```

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

pair_coeff <type 1> <type 2> <c> <delta> <s_{00}> <α>

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 density <ρ>
set group all volume <V_i>

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., $V_i = \Delta x^3$.

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
```

3 Peridynamic Model of a Continuum

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, we 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 $\mathbf{x}$. The *displacement* of the particle at $\mathbf{x}$ in the reference configuration at some time t is denoted $\mathbf{u}(\mathbf{x}, t)$. The *position* of the particle that was at $\mathbf{x}$ in the reference configuration at some later time t is denoted $\mathbf{y}(\mathbf{x}, t) = \mathbf{x} + \mathbf{u}(\mathbf{x}, t)$. Given a two particles with positions $\mathbf{x}$ and $\mathbf{x}'$ in the reference configuration, we denote the relative position vector in the reference configuration as $\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x}$. We denote the relative displacement vector at some time t as $\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t)$. We note here that $\boldsymbol{\eta}$ is time-dependent, and that $\boldsymbol{\xi}$ is not. The acceleration of any particle at position $\mathbf{x}$ in the reference configuration at time t is written as

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{H}_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t) \tag{3.1}$$

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

The pairwise force function $\mathbf{f}$ can be written as

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = f(\boldsymbol{\eta}, \boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|},$$

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

3.2 Prototype Microelastic Materials

In a *prototype microelastic material* [6] the bond force varies linearly with the bond stretch. We assume that the scalar bond force f depends on $\boldsymbol{\eta}$ only through the bond stretch, defined as

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

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

$$f(\boldsymbol{\eta},\boldsymbol{\xi}) = g\left(s(t,\boldsymbol{\eta},\boldsymbol{\xi})\right)\cdot \mu(t,\boldsymbol{\eta},\boldsymbol{\xi}) \tag{3.2}$$

where g is a linear scalar-valued function given by

$$g\left(s(t,\boldsymbol{\eta},\boldsymbol{\xi})\right) = \left\{ \begin{array}{cc} cs(t,\boldsymbol{\eta},\boldsymbol{\xi}) & \text{if } \|\boldsymbol{\xi}\| \le \delta \\ 0 & \text{otherwise} \end{array} \right\},$$

where c is a constant of the form[1]

$$c = \frac{18k}{\pi\delta^4}, \tag{3.3}$$

where δ is the horizon, and k is the bulk modulus of the material, and μ is the history-dependent scalar boolean function

$$\mu(t,\boldsymbol{\eta},\boldsymbol{\xi}) = \left\{ \begin{array}{ll} 1 & \text{if } s(t',\boldsymbol{\eta},\boldsymbol{\xi}) < \min\left(s_0(t',\boldsymbol{\eta},\boldsymbol{\xi}), s_0(t',\boldsymbol{\eta}',\boldsymbol{\xi}')\right) \text{ for all } 0 \le t' \le t \\ 0 & \text{otherwise} \end{array} \right\}. \tag{3.4}$$

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

$$s_0(t,\boldsymbol{\eta},\boldsymbol{\xi}) = s_{00} - \alpha s_{\min}(t,\boldsymbol{\eta},\boldsymbol{\xi}), \qquad s_{\min}(t) = \min_{\boldsymbol{\xi}} s(t,\boldsymbol{\eta},\boldsymbol{\xi}), \tag{3.5}$$

where s_{00} and α are material-dependant constants. The history function μ breaks bonds when the stretch s exceeds the critical stretch s_0.

Although $s_0(t,\boldsymbol{\eta},\boldsymbol{\xi})$ 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 $\mathbf{x}$ and $\mathbf{x}'$ 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 μ above, we have chosen to take the minimum of the two s_0 values for particles $\mathbf{x}$ and $\mathbf{x}'$ when determining if the $\mathbf{x}$-$\mathbf{x}'$ bond should be broken.

Following [6], we can define the damage at a point $\mathbf{x}$ as

$$\varphi(\mathbf{x},t) = 1 - \frac{\int_{\mathcal{H}_{\mathbf{x}}} \mu(t,\boldsymbol{\eta},\boldsymbol{\xi})dV_{\mathbf{x}'}}{\int_{\mathcal{H}_{\mathbf{x}}} dV_{\mathbf{x}'}}. \tag{3.6}$$

[1] This 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:

$$\rho\ddot{\mathbf{y}}(\mathbf{x},t) = \int_{\mathcal{H}_{\mathbf{X}}} \mathbf{f}(\boldsymbol{\eta},\boldsymbol{\xi})\, dV_{\mathbf{X}'} + \mathbf{b}(\mathbf{x},t),$$

where we explicitly track and store at each timestep the positions and not the displacements of the particles. We observe that $\ddot{\mathbf{y}}(\mathbf{x},t) = \ddot{\mathbf{x}} + \ddot{\mathbf{u}}(\mathbf{x},t) = \ddot{\mathbf{u}}(\mathbf{x},t)$, 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 Δx, where each particle i is associated with some volume fraction V_i. For any particle i, let $\mathcal{F}_i$ denote the family of particles for which particle i shares a bond in the reference configuration. That is,

$$\mathcal{F}_i = \{p \mid \|\mathbf{x}_p - \mathbf{x}_i\| \leq \delta\}.$$

The discretized equation of motion replaces (3.1) with

$$\rho\ddot{\mathbf{y}}_i^n = \sum_{p\in\mathcal{F}_i} \mathbf{f}(\mathbf{u}_p^n - \mathbf{u}_i^n, \mathbf{x}_p - \mathbf{x}_i)V_p + \mathbf{b}_i^n \tag{4.1}$$

where $\mathbf{f}$ is given in (3.2), n is the timestep number and subscripts denote the particle number, so that $\mathbf{u}_i^n = \mathbf{u}(\mathbf{x}_i, t_0 + n\Delta t)$.

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 $\mathbf{f}$ in (4.1) the following force

$$\mathbf{f}_S(\mathbf{y}_p, \mathbf{y}_i) = \min\left\{0, \frac{c_S}{\delta}(\|\mathbf{y}_p - \mathbf{y}_i\| - d_{pi})\right\} \frac{\mathbf{y}_p - \mathbf{y}_i}{\|\mathbf{y}_p - \mathbf{y}_i\|}, \tag{4.2}$$

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

$$c_S = 15c. \tag{4.3}$$

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

$$d_{pi} = \min\left\{0.9\,\|\mathbf{x}_p - \mathbf{x}_i\|, 1.35(r_p + r_i)\right\}, \tag{4.4}$$

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

$$\mathcal{F}_i^S = \{p \mid \ \|\mathbf{y}_p - \mathbf{y}_i\| \leq d_{pi}\}.$$

4.3 Modification to the Particle Volume

In a situation where two particles share a bond with $\|\mathbf{x}_p - \mathbf{x}_i\| = \delta$, 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 $\delta - r_i \leq \|\mathbf{x}_p - \mathbf{x}_i\| \leq \delta$ (c.f. Figure 1).

We choose to use a linear unitless nodal volume scaling function

$$\nu(\mathbf{x}_p - \mathbf{x}_i) = \left\{ \begin{array}{cl} -\frac{1}{2r_i}\|\mathbf{x}_p - \mathbf{x}_i\| + \left(\frac{\delta}{2r_i} + \frac{1}{2}\right) & \text{if } \delta - r_i \leq \|\mathbf{x}_p - \mathbf{x}_i\| \leq \delta \\ 1 & \text{if } \|\mathbf{x}_p - \mathbf{x}_i\| \leq \delta - r_i \\ 0 & \text{otherwise} \end{array} \right\}$$

If $\|\mathbf{x}_p - \mathbf{x}_i\| = \delta$, $\nu = 0.5$, and if $\|\mathbf{x}_p - \mathbf{x}_i\| = \delta - r_i$, $\nu = 1.0$, for example.

4.4 Discrete Equation of Motion

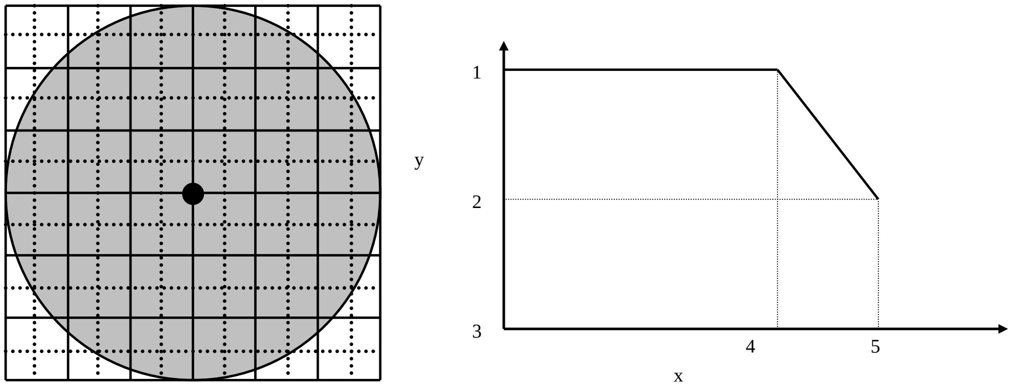
The semi-discrete equation of motion can be written as

$$\begin{aligned} \rho \ddot{\mathbf{y}}_i^n &= c \sum_{p \in \mathcal{F}_i} \left(\frac{\|\mathbf{y}_p - \mathbf{y}_i\| - \|\mathbf{x}_p - \mathbf{x}_i\|}{\|\mathbf{x}_p - \mathbf{x}_i\|} \right) \mu(t, \boldsymbol{\eta}, \boldsymbol{\xi}) \nu(\mathbf{x}_p - \mathbf{x}_i) V_p \frac{\mathbf{y}_p - \mathbf{y}_i}{\|\mathbf{y}_p - \mathbf{y}_i\|} \\ &+ \sum_{p \in \mathcal{F}_i^S} \min \left\{ 0, \frac{c_S}{\delta} (\|\mathbf{y}_p - \mathbf{y}_i\| - d_{ip}) \right\} V_p \frac{\mathbf{y}_p - \mathbf{y}_i}{\|\mathbf{y}_p - \mathbf{y}_i\|} + \mathbf{b}_i^n, \end{aligned}$$

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 (Algorithm 1) is generally expressed in three steps, as where m denotes the mass of a particle, and $\widetilde{\mathbf{f}}_i^n$ 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 δ as grey circular region. Dual mesh (dotted lines) shows boundaries of each particle.

(b) Plot of $\nu(\mathbf{x}_p - \mathbf{x}_i)$ vs. $\|\mathbf{x}_p - \mathbf{x}_i\|$.

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

Algorithm 1 Velocity Verlet

```
$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{\Delta t}{2m}\widetilde{\mathbf{f}}_i^n$
$\mathbf{y}_i^{n+1} = \mathbf{y}_i^n + \Delta t \mathbf{v}_i^{n+1/2}$
$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \frac{\Delta t}{2m}\widetilde{\mathbf{f}}_i^{n+1}$
```

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_{min} in (3.5), then loop over all bonds again and break bonds with a stretch $s > s0$ as in (3.4), and finally loop over all particles and compute forces for the next step of Algorithm 1. For reasons of computational efficiency, we will utilize the values of $\mathbf{s}_0$ from the *previous* timestep when deciding to break a bond. For the first timestep, $\mathbf{s}_0$ is initialized to $\boldsymbol{\infty}$ 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 projectile in the next section is placed such that it does not impact the brittle plate until several 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

```
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.
Initialize bonds between all particles where $\|\mathbf{x} - \mathbf{x}'\| \leq \delta$.
Initialize $\mathbf{s}_0 = \infty$ {Initialize each entry to MAX_DOUBLE.}
while not done do
  Perform step 1 of Algorithm 1, updating velocities of all particles.
  Perform step 2 of Algorithm 1, updating positions of all particles.
  $\tilde{\mathbf{s}}_0 = \infty$ {Initialize each entry to MAX_DOUBLE.}
  for $i = 1$ to $N$ do
    {Compute short-range forces}
    for all particles $j \in \mathcal{F}_i^S$ (the short-range family of nodes for particle $i$) do
      $r = \|\mathbf{y}_i - \mathbf{y}_j\|$.
      $dr = \min\{0, r - d\}$. {Short-range forces are only repulsive, never attractive}
      $k = \frac{c_S}{\delta} V_k dr$. {$c_S$ defined in (4.3)}
      $\mathbf{f} = \mathbf{f} - k \frac{\mathbf{y}_i - \mathbf{y}_j}{\|\mathbf{y}_i - \mathbf{y}_j\|}$.
    end for
  end for
  for $i = 1$ to $N$ do
    {Compute bond forces.}
    for all particles $j$ sharing a bond with particle $i$ do
      $r = \|\mathbf{y}_i - \mathbf{y}_j\|$.
      $dr = r - \|\mathbf{x}_i - \mathbf{x}_j\|$.
      $k = \frac{c}{\|\mathbf{x}_i - \mathbf{x}_j\|} V_j \nu(\mathbf{x}_i - \mathbf{x}_j) dr$. {$c$ defined in (3.3)}
      $\mathbf{f} = \mathbf{f} - k \frac{\mathbf{y}_i - \mathbf{y}_j}{\|\mathbf{y}_i - \mathbf{y}_j\|}$.
      if $\frac{dr}{\|\mathbf{x}_i - \mathbf{x}_j\|} > \min(\mathbf{s}_0(i), \mathbf{s}_0(j))$ then
        Break $i$'s bond with $j$. {$j$'s bond with $i$ will be broken when this loop iterates on $j$}
      end if
      $\tilde{\mathbf{s}}_0(i) = \min(\tilde{\mathbf{s}}_0(i), s_{00} - \alpha \frac{dr}{\|\mathbf{x}_i - \mathbf{x}_j\|})$.
    end for
  end for
  $\mathbf{s}_0 = \tilde{\mathbf{s}}_0$. {Store for use in next timestep.}
  Perform step 3 of Algorithm 1, updating velocities of all particles.
end while
```

4.7 Damage

The damage associated with every particle (see (3.6)) can optionally be computed and output with a LAMMPS data dump. To do this, your input script must contain the command

```
compute <ComputeID> all damage/atom
```

This enables a LAMMPS per-atom compute to calculate the damage associated with each particle every time a LAMMPS data dump is called. To output the results of this compute in your dump file, you must use the LAMMPS dump command, as

```
dump <DumpID> all custom <N> <output filename> tag type x y z c_<ComputeID>
```

where N is the number of timesteps between dumps.

4.8 Pitfalls

Parallel Scalability. 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. LAMMPS decomposes the overall simulation domain so that spatial subdomains of nearly equal volume are assigned to each processor. When each subdomain contains nearly the same number of particles, this results in a reasonable load balance among all processors. As is more typical with some peridynamic simulations, some subdomains may contain many particles while other subdomains contain few particles, resulting in a load imbalance that impacts parallel scalability.

Setting the "skin" distance. The neighbor command with LAMMPS is used to set the so-called "skin" distance used when building neighbor lists. All atom pairs within a cutoff distance equal to the horizon δ plus the skin distance are stored in the list. Unexpected crashes in LAMMPS may be due to too small a skin distance. The skin should be set to a value such that δ plus the skin distance is larger than the maximum possible distance between two bonded particles. For example, if s_{00} is increased, the skin distance may also need to be increased.

"Lost" particles. All particles are contained within the "simulation box" of LAMMPS. The boundaries of this box may change with time, or not, depending on how the LAMMPS boundary command has been set. If a particle drifts outside the simulation box during the course of a simulation, it is called *lost*.

As an option of the themo_modify command of LAMMPS, the lost keyword determines whether LAMMPS checks for lost atoms each time it computes thermodynamics and what it does if atoms are lost. If the value is ignore, LAMMPS does not check for lost atoms. If the value is error or warn, LAMMPS checks and either issues an error or warning. The code will exit with an error and continue with a warning. This can be a useful debugging option. The default behavior of LAMMPS is to exit with an error if a particle is lost.

The peridynamic module within LAMMPS does not check for lost atoms. If a particle with unbroken bonds is lost, those bonds are marked as broken by the remaining particles.

Defining the peridynamic horizon δ. In the `pair_coeff` command, the user must specify the horizon δ. This argument determines which particles are bonded when the simulation is initialized. It is recommended that δ be set to a small fraction of a lattice constant larger than desired.

For example, if the lattice constant is 0.0005 and you wish to set the horizon to three times the lattice constant, then set δ to be 0.0015001, a value slightly larger than three times the lattice constant. This guarantees that particles three lattice constants away from each other are still bonded. If δ is set to 0.0015, for example, floating point error may result in some pairs of particles three lattice constants apart not being bonded.

4.9 Bugs

The user is cautioned that this code is a beta release. If you are confident that you have found a bug in the peridynamic module, please send an email to the developers. First, check the "New features and bug fixes" section of the LAMMPS website site to see if the bug has already been reported or fixed. If not, the most useful thing you can do for us is to isolate the problem. Run it on the smallest number of atoms and fewest number of processors and with the simplest input script that reproduces the bug. In your email, describe the problem and any ideas you have as to what is causing it or where in the code the problem might be. We'll request your input script and data files if necessary.

4.10 Modifying and Extending the Peridynamic Module

The peridynamic module of LAMMPS contains only a simple implementation of the PMB peridynamic model. To add new features or peridynamic potentials to the peridynamic module, the user is referred to section 8 of the LAMMPS user manual, *Modifying & extending LAMMPS*. It is suggested that the user start with the PMB model as a template.

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 disk 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 $s_{00} = 0.0005$ 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

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

The CFL analysis from [6] shows that a timestep of 1.0×10^{-7} is safe.

We observe here that in IEEE double-precision floating point arithmetic when computing the bond stretch $s(t, \boldsymbol{\eta}, \boldsymbol{\xi})$ at each iteration where $\|\boldsymbol{\eta} + \boldsymbol{\xi}\|$ is computed during the iteration and $\|\boldsymbol{\xi}\|$ 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 $csV_i \approx 4.68 \times 10^{-4}$, 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:

```
**if** $|dr| < \varepsilon_{machine}$ **then**
    $dr = 0$.
**end if**
```

Qualitatively, this says that displacements from equilibrium 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 2 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 3 and 4. In line 5 we specify that peridynamic particles are to be used for this simulation. In line 7, we set the "skin" distance used in building the LAMMPS neighborlist. In line 8 we set the lattice constant (in meters) and in line 10 we define the spatial region where the target will be placed. In line 12 we specify a rectangular box enclosing the target region that defines the simulation domain. Line 14 fills the target region with atoms. Lines 15 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 23 sets the initial velocity of all particles to zero. 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 29 declares a compute style for the damage (percentage of broken bonds) associated with each particle. Line 30 sets the timestep, line 31 instructs LAMMPS to provide a screen dump of thermodynamic quantities every 200 timesteps, and line 32 instructs LAMMPS to create a data file (`dump.output`) with a complete snapshot of the system every 100 timesteps. This file can be used to create still images or movies. Finally, line 33 instructs LAMMPS to run for 2,000 timesteps.

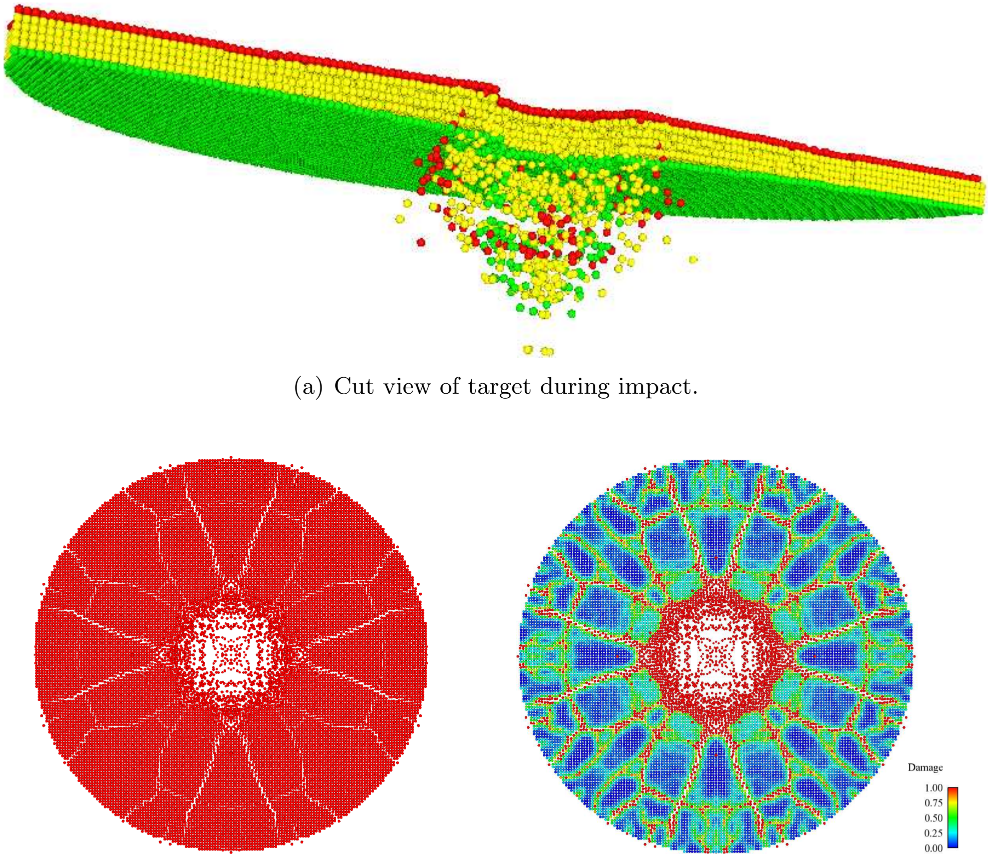
5.4 Numerical Results and Discussion

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 plot of damage on the top monolayer was created by coloring each particle according to its damage (see (3.6)).

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. To do this, a perturbed lattice of points can be prepared in a data file and read into LAMMPS using the `read_data` command.

Algorithm 3 Example LAMMPS Input Script

```
# 3D Peridynamic simulation with projectile
units           si
dimension       3
boundary        s s s
atom_style      peri
atom_modify     map array
neighbor        0.0010 bin
lattice         sc 0.0005
# Create desired target
region          target cylinder y 0.0 0.0 0.037 -0.0025 0.0 units box
# Make 1 atom type
create_box      1 target
# Create the atoms in the simulation region
create_atoms    1 region target
pair_style      peri/pmb
#               <type1> <type2>    <c>    <horizon>  <s00> <alpha>
pair_coeff         *       *    1.6863e22 0.0015001 0.0005  0.25
# Set mass density
set             group all density 2200
# volume = lattice constant^3
set             group all volume 1.25e-10
# Zero out velocities of particles
velocity        all set 0.0 0.0 0.0 sum no units box
# Use velocity-Verlet time integrator
fix             F1 all nve
# Construct spherical indenter to shatter target
fix             F2 all indent 1e17 sphere 0.0 0.0051 0.0 0.005 vel 0.0 -100.0 0.0 units box
# Compute damage for each particle
compute         C1 all damage/atom
timestep        1.0e-7
thermo          200
dump            D1 all custom 100 dump.output tag type x y z c_C1
run             2000
```



(a) Cut view of target during impact.

(b) Top monolayer showing fragmentation.

(c) Top monolayer showing damage. (blue = 0% broken bonds; red = 100% broken bonds)

Figure 2. Target during (a) and after (b,c) impact.

References

[1] CEI, *EnSight web page.* http://www.ensight.com/.

[2] E. Emmrich and O. Weckner, *On the well-posedness of the linear peridynamic model and its convergence towards the Navier equation of linear elasticity*, Commun. Math. Sci., 5 (2007), pp. 851–864.

[3] S. J. Plimpton, *Pizza.py webpage.* http://www.cs.sandia.gov/~sjplimp/pizza.html.

[4] S. J. Plimtpon, *Fast parallel algorithms for short-range molecular dynamics*, J. Comp. Phys., 117 (1995), pp. 1–19. Available at http://lammps.sandia.gov.

[5] S. A. Silling. Personal communication, 2007.

[6] S. A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics*, Computer and Structures, 83 (2005), pp. 1526–1535.

[7] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*, vol. 1, The MIT Press, Cambridge, MA., 2 ed., 1998.

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

Sandia National Laboratories