Implementing peridynamics within a molecular dynamics code

Michael L. Parks [a,*,1], Richard B. Lehoucq [a,1], Steven J. Plimpton [b,1], Stewart A. Silling [c,1]

[a] *Applied Mathematics and Applications, Sandia National Laboratories, P.O. Box 5800, MS 1320, Albuquerque, NM 87185, USA*
[b] *Scalable Algorithms, Sandia National Laboratories, P.O. Box 5800, MS 1316, Albuquerque, NM 87185, USA*
[c] *Multiscale Dynamic Materials Modeling, Sandia National Laboratories, P.O. Box 5800, MS 1322, Albuquerque, NM 87185, USA*

ARTICLE INFO

ABSTRACT

Peridynamics (PD) is a continuum theory that employs a nonlocal model to describe material properties. In this context, nonlocal means that continuum points separated by a finite distance may exert force upon each other. A meshless method results when PD is discretized with material behavior approximated as a collection of interacting particles. This paper describes how PD can be implemented within a molecular dynamics (MD) framework, and provides details of an efficient implementation. This adds a computational mechanics capability to an MD code, enabling simulations at mesoscopic or even macroscopic length and time scales.

1. Introduction

Molecular dynamics (MD) suffers from well-known computational limitations in the length and time scales it can address, even on large parallel supercomputers. Numerous recent efforts attempt to coarse-grain MD or to couple it to meso- and macro-scale models to enable "multiscale" modeling of phenomena such as crack growth, indentation, flow near surfaces, and heat transfer as described in [4,9–11].

The purpose of this paper is to describe how peridynamics [14, 17], a continuum theory, can be implemented within an MD framework so enabling meso-scale and macroscale modeling. As we shall see, the force interactions that result from discretizing peridynamics (PD) are similar to traditional MD forces, yet have additional characteristics that must be addressed for an efficient implementation. Thus, with minor modifications, an MD code can perform PD calculations. The recent paper [19] explains how PD converges to

the classical elastic material model assuming that the underlying deformation is sufficiently smooth. This suggests that a computational mechanics capability at length scales substantially beyond those typically associated with MD are realizable within an MD framework.

The PD theory of continuum mechanics belongs to the class of microcontinuum theories defined by generalizing the local force assumption to allow force at a distance (see [1,3] for general discussions and references) so introducing a length-scale. In the classical continuum context, "local force" means that only continuum points in direct contact can exert a force on each other. The force arises from a stress vector acting at a point on an oriented surface. In contrast, PD employs an integral operator to *sum* forces avoiding the use of stress/strain fields in its equation of motion. Instead, the material behavior in PD is specified by nonlocal force interactions, assumed to be a function of the positions of the continuum points. No assumption are made on the continuity or differentiability of the displacement field. Because the displacement field is not assumed even weakly differentiable, PD can be employed for deformation that does not satisfy the smoothness assumptions of classical continuum mechanics, e.g., fracture or fragmentation. The

---

reader is referred to [2,15,16] for information describing PD[2] modeling in several applications at length and timescales up to meters and seconds, respectively.

When PD is discretized, a meshless method [15] results, where the material is approximated as a collection of interacting continuum points. Meshless methods such as SPH (Smoothed Particle Hydrodynamics), EFG (Element Free Galerkin), RKPM (Reproducing Kernel Particle Method), and XFEM (Extended Finite Element Method), have received considerable attention from the continuum mechanics community [8], and are typically formulated by discretizing the classical equations using a set of nodes. Although a local force assumption is assumed in the continuum model, a notion of nonlocality is often introduced at the discrete level in an ad-hoc fashion. In contrast, discretized PD inherits nonlocality because the continuum PD model is nonlocal.

The remainder of the paper is organized as follows. We first review PD in Section 2 and its discretization in Section 3. We have implemented PD in LAMMPS, Sandia National Laboratories' molecular dynamics code [13]. Details of the implementation are discussed in Section 4. Finally, Section 5 illustrates the style and scope of macroscopic simulations such a modified MD code can then perform.

2. Peridynamics

We briefly review the salient details of peridynamics. The reader is referred to [14,15,17] for further details. Let a body in some reference configuration occupy a region $\mathcal{B}$. For any $\mathbf{x} \in \mathcal{B}$ the PD equation of motion is

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{B}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t), \quad t \geqslant 0, \tag{1}$$

where $\mathbf{u}(\mathbf{x}, t)$ is the displacement field with initial conditions $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$, $\dot{\mathbf{u}}(\mathbf{x}, 0) = \dot{\mathbf{u}}_0(\mathbf{x})$. The vector function $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$ denotes the force density per unit reference volume exerted on a point $\mathbf{y} = \mathbf{x} + \mathbf{u}(\mathbf{x}, t)$ by the point $\mathbf{y}' = \mathbf{x}' + \mathbf{u}'$, where $\mathbf{u}' = \mathbf{u}(\mathbf{x}', t)$. The vectors $\boldsymbol{\eta} = \mathbf{u}' - \mathbf{u}$, and $\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x}$ denote the relative displacement and relative position in the reference configuration, respectively. Hence $\mathbf{y}' - \mathbf{y} = \boldsymbol{\xi} + \boldsymbol{\eta}$, gives the current relative position between $\mathbf{x}$ and $\mathbf{x}'$ in the deformed configuration. The vector $\mathbf{b}(\mathbf{x}, t)$ is the loading force density, and the mass density is denoted by $\rho(\mathbf{x})$.

We now discuss the properties that a function $\mathbf{f}$ should possess so that linear and angular momentum are conserved. Consider $\Omega_{\mathbf{x}} \subset \mathcal{B}$ and integrate (1) over $\Omega_{\mathbf{x}}$ to obtain

$$\int_{\Omega_{\mathbf{x}}} \rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) \, dV_{\mathbf{x}} = \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}/\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}} + \int_{\Omega_{\mathbf{x}}} \mathbf{b}(\mathbf{x}, t) \, dV_{\mathbf{x}}. \tag{2}$$

The first term on the right-hand side of (2) represents the internal force that the material $\mathcal{B}/\Omega_{\mathbf{x}}$ exerts on the material $\Omega_{\mathbf{x}}$. This internal force is nonlocal precisely because the interaction of material inside $\Omega_{\mathbf{x}}$ with material outside $\Omega_{\mathbf{x}}$ cannot be restricted to a contact force along the surface of $\Omega_{\mathbf{x}}$, in contrast to the classical theory of continuum mechanics. The derivation of (2) used the relation

$$\int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$= \int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}} + \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}/\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$= \mathbf{0} + \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}/\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}},$$

---

[2] See http://en.wikipedia.org/wiki/Peridynamics.

where the latter equality follows from conservation of linear momentum. Moreover, the conservation principle also implies that $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) + \mathbf{f}(-\boldsymbol{\eta}, -\boldsymbol{\xi}) = \mathbf{0}$.

To consider conservation of angular momentum, take the cross product of (1) with $\mathbf{y}$ and integrate over $\Omega_{\mathbf{x}}$ to obtain

$$\int_{\Omega_{\mathbf{x}}} \mathbf{y} \times \rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) \, dV_{\mathbf{x}} = \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}} \mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$+ \int_{\Omega_{\mathbf{x}}} \mathbf{y} \times \mathbf{b}(\mathbf{x}, t) \, dV_{\mathbf{x}}. \tag{3}$$

The left-hand side of (3) represents the torque about the origin caused by the material in the domain $\Omega_{\mathbf{x}}$. We may write the first term on the right-hand side as

$$\int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}} \mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}} = \int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{x}}} \mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$+ \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}/\Omega_{\mathbf{x}}} \mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}. \tag{4}$$

Under the assumption that $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$ is antisymmetric (from above), and also that $\mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$ is antisymmetric, we may conclude that $(\mathbf{y}' - \mathbf{y}) \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = 0$, and that

$$\int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{x}}} \mathbf{y} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}} = \int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{x}}} (\mathbf{y}' - \mathbf{y}) \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}} = 0.$$

This allows us to replace the first term on the right-hand side of (3) with the rightmost term in (4), and to conclude that $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$ is parallel to the current relative position vector $\mathbf{y}' - \mathbf{y}$.

The material behavior is specified by $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$, which maps the deformation given by $\boldsymbol{\eta}$ for a bond $\boldsymbol{\xi}$ to the force density (per unit volume). For a microelastic material, this map can be derived from a micropotential Φ. As an example, consider a prototype microelastic material [15] where the potential (per unit volume squared) is

$$\Phi(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{1}{2} \frac{c}{\|\boldsymbol{\xi}\|} \big(\|\boldsymbol{\eta} + \boldsymbol{\xi}\| - \|\boldsymbol{\xi}\| \big)^2, \tag{5}$$

where $c/\|\boldsymbol{\xi}\| > 0$ is the stiffness per unit volume squared and $\|\boldsymbol{\xi}\|$ is the equilibrium length of the spring. The gradient of (5) gives a pairwise force density per unit volume function of

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \nabla_{\boldsymbol{\eta}} \Phi(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{c}{\|\boldsymbol{\xi}\|} \big(\|\boldsymbol{\eta} + \boldsymbol{\xi}\| - \|\boldsymbol{\xi}\| \big) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|}. \tag{6}$$

A standard assumption is that for a given material, $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = 0$ for all $\boldsymbol{\eta}$ when $\|\boldsymbol{\xi}\| > \delta$ for some $\delta > 0$, the *horizon*. We denote the horizon of $\mathbf{x}$ by $\delta(\mathbf{x})$. We remark that $\delta(\mathbf{x})$ is a constitutive parameter defined in the reference configuration, and that $\mathbf{f}(\boldsymbol{\eta}, \delta)$ may not be a negligible force. We also denote by $\mathcal{H}(\mathbf{x})$ the spherical neighborhood of $\mathbf{x}$ in $\mathcal{B}$ with radius $\delta(\mathbf{x})$.

The spatial discretization of (2) divides the region $\mathcal{B}$ into subdomains $\Omega_{\mathbf{x}}$ so that $\mathcal{B} = \bigcup_{\mathbf{x} \in \mathcal{B}} \Omega_{\mathbf{x}}$ where the number of subdomains is finite. A discrete approximation to the internal force in (2) using (6) results in

$$\int_{\Omega_{\mathbf{x}}} \int_{\mathcal{B}/\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$= \int_{\Omega_{\mathbf{x}}} \int_{\mathcal{H}(x)/\Omega_{\mathbf{x}}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \, dV_{\mathbf{x}'} \, dV_{\mathbf{x}}$$

$$\approx \sum_{\mathbf{x}' \in \mathcal{H}(\mathbf{x})/\Omega_{\mathbf{x}}} \frac{c}{\|\boldsymbol{\xi}\|} \big(\|\boldsymbol{\eta} + \boldsymbol{\xi}\| - \|\boldsymbol{\xi}\| \big) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|} V_{\mathbf{x}'} V_{\mathbf{x}}, \tag{7}$$

where $V_{\mathbf{x}'}$ and $V_{\mathbf{x}}$ are the volumes associated with $\Omega_{\mathbf{x}'}$ and $\Omega_{\mathbf{x}}$, respectively. Hence, the discretization of PD gives rise to an interacting set of "particles" approximating the behavior of the material where a particle $\mathbf{x}$ is identified with a subdomain $\Omega_{\mathbf{x}}$. The resulting semidiscrete equation of motion can be combined with a velocity-Verlet time integration scheme for a fully discrete approximation to the PD equation of motion (1). This correspondence allows PD to be implemented within an MD framework because of the analogous computational structure.

The force functions introduced in [14] (and discussed above) assumes a central force interaction. The recent paper [17] generalizes central force interactions to those depending upon the force state, or collective behavior, at $\mathbf{y}'$ and $\mathbf{y}$. This extension of PD allows a continuum generalization of multibody force interactions. We also remark that an MD notion of a cutoff can be introduced so that if $\|\boldsymbol{\eta} + \boldsymbol{\xi}\|$ exceeds a prescribed value, the force interaction is set to zero. The cutoff is defined in the deformed configuration in contrast to the material parameter $\delta(\mathbf{x})$ defined in the reference configuration.

3. Inter-particle forces used in peridynamics

We now discuss the form of inter-particle forces that result from a discretization of PD. As an example, we derive forces and a particle equation of motion for the *prototype microelastic brittle* (PMB) material model introduced in [15]. A PMB material specializes the force interaction of (7) to allow for a bond breaking mechanism. We also discuss short-range repulsive forces and the calculation of particle volumes.

3.1. Prototype Microelastic Brittle (PMB) materials

For a PMB material

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = g(\boldsymbol{\eta}, \boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|}, \tag{8a}$$

$$g(\boldsymbol{\eta}, \boldsymbol{\xi}) = \begin{cases} cs(t, \boldsymbol{\eta}, \boldsymbol{\xi})\mu(\boldsymbol{\eta}, \boldsymbol{\xi}), & \|\boldsymbol{\xi}\| \leqslant \delta, \\ 0, & \|\boldsymbol{\xi}\| > \delta, \end{cases} \tag{8b}$$

depends upon the *bond strain*

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

and a history-dependent scalar boolean function

$$\mu(t, \boldsymbol{\eta}, \boldsymbol{\xi}) = \begin{cases} 1, & s(t', \boldsymbol{\eta}, \boldsymbol{\xi}) < \min\{s_0(t', \boldsymbol{\eta}, \boldsymbol{\xi}), s_0(t', \boldsymbol{\eta}', \boldsymbol{\xi}')\}, \\ & 0 \leqslant t' \leqslant t, \\ & s_0(t, \boldsymbol{\eta}, \boldsymbol{\xi}) = s_{00} - \alpha s_{\min}(t, \boldsymbol{\eta}, \boldsymbol{\xi}), \\ & s_{\min}(t) = \min_{\boldsymbol{\xi}} s(t, \boldsymbol{\eta}, \boldsymbol{\xi}), \\ & \boldsymbol{\eta}' = \mathbf{u}(\mathbf{x}'', t) - \mathbf{u}(\mathbf{x}', t), \ \boldsymbol{\xi}' = \mathbf{x}'' - \mathbf{x}', \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $s_0(t, \boldsymbol{\eta}, \boldsymbol{\xi})$ is a *critical strain* and s_{00} and α are material-dependent constants. μ is 1 for an unbroken bond and 0 otherwise. 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 particles 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 bond between $\mathbf{x}$ and $\mathbf{x}'$ should be broken.

We remark that (8) and (9) imply that a PMB material does not allow "healing", e.g., once a bond between two particles is broken, the bond remains broken. The assumption is that once the underlying material fractures, then the material remains fractured.

This is in contrast to MD where force interactions between atoms may be zero or non-zero over time. However, material healing (or lack thereof) depends upon the constitutive relationship $\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})$ and does not represent any intrinsic limitation of PD.

3.2. Short-range forces

In the preceding section, particles interact only through bond forces. A particle with no bonds becomes a free non-interacting particle. To prevent subsequent particle overlap, short-range repulsive forces are introduced. We add to the force $\mathbf{f}$ in (1) the following force

$$\mathbf{f}_S(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|} \min\left\{0, \frac{c_S}{\delta}\left(\|\boldsymbol{\eta} + \boldsymbol{\xi}\| - d_S\right)\right\}, \tag{10a}$$

$$d_S = \min\{0.9\|\mathbf{x} - \mathbf{x}'\|, 1.35(r_S + r_S')\}, \tag{10b}$$

where r_S is defined as the *node radius*. Given a discrete lattice of particles, we choose r_S to be half the lattice constant. Note that short-range forces are only repulsive, never attractive. The repulsive force above may also be replaced by a "hard" potential, e.g., $\|\boldsymbol{\eta} + \boldsymbol{\xi}\|^{-12}$, the repulsive part of the Lennard-Jones potential.

3.3. The discrete equation of motion

The region defining a peridynamic material is discretized into particles forming a cubic lattice with lattice constant a, where each particle i is associated with some volume $V_i = a^3$. Recall that $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the reference configuration (initial position) and position at time t, respectively, of particle i. Further, for any particle i, let

$$\mathcal{F}_i = \left\{ j \mid \|\mathbf{x}_j - \mathbf{x}_i\| \leqslant \delta, \ j \neq i \right\}, \tag{11a}$$

$$\mathcal{F}_i^S = \left\{ j \mid \|\mathbf{y}_j - \mathbf{y}_i\| \leqslant d_S, \ j \neq i \right\}, \tag{11b}$$

where d_S is defined in (10b). The former denotes the family of particles within a distance δ of particle i in the reference configuration, and the latter denotes the family of particles within a distance d_S of particle i in the current configuration.

We explicitly track and store the positions and not the displacements of the particles associated with a discretization of (2) because $\ddot{\mathbf{y}}(\mathbf{x}, t) = \ddot{\mathbf{x}} + \ddot{\mathbf{u}}(\mathbf{x}, t) = \ddot{\mathbf{u}}(\mathbf{x}, t)$. Using (8), (10)–(11) the semi-discrete peridynamic equation of motion can then be written as

$$\begin{aligned} \left(\rho(\mathbf{x}_i)V_i\right)\ddot{\mathbf{y}}_i^n = & \sum_{j \in \mathcal{F}_i} \mathbf{f}\left(\mathbf{u}_j^n - \mathbf{u}_i^n, \mathbf{x}_j - \mathbf{x}_i\right)\tilde{V}_j V_i \\ & + \sum_{j \in \mathcal{F}_i^S} \mathbf{f}_S\left(\mathbf{u}_j^n - \mathbf{u}_i^n, \mathbf{x}_j - \mathbf{x}_i\right)V_j V_i + \mathbf{b}_i^n V_i, \end{aligned} \tag{12}$$

where $\tilde{V}_j = \nu(\mathbf{x}_j - \mathbf{x}_i)V_j$. We introduce the function $\nu(\mathbf{x}_j - \mathbf{x}_i)$ as a scale factor on V_j for the following reason. Some of the particles j to which particle i is bonded will be near the boundary of $\mathcal{H}(\mathbf{x}_i)$ (the sphere of radius δ surrounding particle i). For these interactions only a portion of V_j is inside the sphere and the bond strength should be diminished as a result. The following linear dimensionless nodal volume scaling function accounts for this effect:

$$\nu(\mathbf{x} - \mathbf{x}') = \begin{cases} -\frac{1}{2r_S}\|\mathbf{x}_j - \mathbf{x}_i\| + (\frac{\delta}{2r_S} + \frac{1}{2}), \\ \qquad \delta - r_S \leqslant \|\mathbf{x}_j - \mathbf{x}_i\| \leqslant \delta, \\ 1, & \|\mathbf{x}_j - \mathbf{x}_i\| \leqslant \delta - r_S, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

Note that if $\|\mathbf{x}_j - \mathbf{x}_i\| = \delta$, $\nu = 0.5$, and if $\|\mathbf{x}_j - \mathbf{x}_i\| \leqslant \delta - r_S$, $\nu = 1.0$.

4. Implementation of PD in LAMMPS

We now explain how Eq. (12) was implemented within the molecular dynamics code LAMMPS [13], an open-source, general-purpose, massively parallel MD simulator. LAMMPS provides a variety of interatomic potentials for biological and polymer systems, solid-state materials, and other coarse-grained models, but PD was the first continuum-level model added to the code. LAMMPS is well-suited for implementing PD because it is designed to allow new potentials, boundary conditions, and particle attributes to be easily added without affecting the code's operation when non-PD models are simulated.

From an MD perspective, Eq. (12) can be rewritten succinctly as a *potential* for the energy of particle i in the following form

$$U_i = \sum_{j \in \mathcal{F}_i^S} \Phi_{\text{short-range}}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i)$$
$$+ \sum_{j \in \mathcal{F}_i} \Phi_{\text{bond}}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i), \qquad (14a)$$

where

$$\Phi_{\text{short-range}}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i)$$
$$= \begin{cases} \frac{c_s}{2\delta}(\|\mathbf{y}_j - \mathbf{y}_i\| - d_S)^2, & \|\mathbf{y}_j - \mathbf{y}_i\| \leqslant d_S, \\ 0, & \text{otherwise,} \end{cases} \qquad (14b)$$

$$\Phi_{\text{bond}}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i)$$
$$= \begin{cases} \Phi(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i), & \text{if bond unbroken,} \\ 0, & \text{otherwise,} \end{cases} \qquad (14c)$$

where $\Phi(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i)$ is defined in Eq. (5).

The first term $\Phi_{\text{short-range}}$ is the short-range potential (derived from Eq. (10)) that prevents particles from overlapping. When particles are separated by a distance greater than d_S the interaction is zero. This term is effectively the repulsive portion of a harmonic spring with equilibrium length d_S. Note that this is a much softer short-range repulsive potential than the $\|\mathbf{x}_j - \mathbf{x}_i + \mathbf{u}_j - \mathbf{u}_i\|^{-12}$ repulsion provided, for example, by a Lennard-Jones interaction [7]. (See [2] for results combining a Lennard-Jones interaction and harmonic potential.)

The second term Φ_{bond} is the cohesive potential for the material, summed over all $j \in \mathcal{F}_i$ particles that are initially within a distance δ of particle i. This is effectively a list of harmonic "bond" partners of particle i. Note that the effective bond strength falls off as the inverse of the initial bond length $\mathbf{x}_j - \mathbf{x}_i$, which can be seen from Eq. (5), and is also a function of the scaled volume factor $\tilde{V}_j$ defined in Eq. (12). The bond potential is set to zero once it stretches beyond a critical length, as discussed for the PMB material model in Eqs. (8) and (9). An individual bond is active for all time t until this occurs. If the bond breaks, it is never again active, even if the two particles later come close together. However, as explained at the end of Section 3.1, this is a constitutive assumption on the material, and bond healing may easily be incorporated.

The critical strain s_0 of Eq. (9) is defined on a per-particle basis and is computed each timestep for testing that particle's bonds. To ensure symmetry in bond breaking, if the strain of the bond between particles i, j exceeds the smaller of s_0 for particle i or s_0 for particle j, then the bond "breaks". The bond is deleted from the lists $\mathcal{F}_i$ and $\mathcal{F}_j$ of bond partners for both particles and contributes no energy or force to the system for all subsequent timesteps.

As Eq. (14a) implies, the following parameters are input by the user to define the PD potential for a specific material: ρ, c, s_{00}, α, and δ. If different particle types represent multiple materials in a more complex model, each of these parameters can be defined for each pair of interacting particle types. The functional forms of $\Phi_{\text{short-range}}$ and Φ_{bond} are specific to the PMB material model of

Section 3. However, PD models for other materials result in potential functions with similar characteristics: a short-range repulsive term and a history-dependent cohesive term that can turn off as large deformations occur.

We now detail how Eq. (14a) was implemented in LAMMPS. First, a set of consistent units suitable for macroscopic simulations was needed. LAMMPS allows the user to choose units convenient for their particular simulations, each of which is implemented as a handful of conversion factors. These are used when, for example, kinetic energy $(1/2mv^2)$ is computed with velocities in Angstroms/fs and the result should be in Kcal/mole (for an atomistic simulation). For peridynamic simulations, an "SI" option was added where energy = Joules, distance = meters, time = seconds, etc.

LAMMPS operates in parallel in a spatial-decomposition mode [13], where each processor owns a sub-domain of the overall simulation box and the particles within the sub-domain. To compute pairwise or bond forces, a processor communicates with its neighboring processors via distributed-memory message passing (MPI) to acquire information about nearby "ghost" particles owned by those processors. To improve the efficiency of the computation of pairwise forces, LAMMPS uses Verlet neighbor lists [20] that are recomputed every few timesteps via binning particles on a regular grid and searching nearby bins to find neighbors [5]. These are also the timesteps on which particles migrate to new processors as needed. For peridynamic simulations, Verlet neighbor lists are used only when computing the family of short-range interacting particles, $\mathcal{F}_i^S$.

Each PD particle stores 4 quantities in addition to the usual coordinates, velocities, forces, etc. These are the critical strain s_0 from Eq. (9), the particle volume V, the particle density ρ, and the initial position of the particle $\mathbf{x}_0$. V is precomputed for each particle based on the initial problem geometry.

These 4 quantities migrate with particles as they move from processor to processor. Additionally, the s_0 value for each ghost particle is communicated every timestep, since it is a dynamic quantity and the breakage criterion for bond i, j depends on the s_0 value of both particles. Similarly, the V value for ghost particles is needed to scale the bond force between particles i, j. Since V is a static quantity, it is only communicated on timesteps when reneighboring is done.

The short-range term in Eq. (14a) is conceptually identical to standard short-range pair potential computations within an MD code. In LAMMPS a "half" neighbor list is used to efficiently find neighboring particles within a distance d_S on a given timestep. By "half" we mean that any interacting pair i, j is only stored once in the list, either by particle i or by particle j. The forces on particles i, j are computed for each pair in the list. At the end of the force computation, forces on ghost particles are communicated back to the owning processor.

Computation of the bond term in Eq. (14a) requires knowledge of which bonds are already broken. Thus each particle stores a list of its bond partners, denoted by $\mathcal{F}_i$ in (11a), and flags them as they break. For each partner, the initial bond distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ is also stored, since it is used to calculate bond strength. The bond family of particles $\mathcal{F}_i$ for each particle is computed only once, on the first timestep of the simulation, based on the initial undeformed state of the material. The union of $\mathcal{F}_i$ over all particles is effectively a "full" neighbor list where the geometric neighbors of particle i within a cutoff distance δ are stored. By "full" we mean the interacting pair i, j is stored twice, once by particle i and once by particle j.

With this information, the bond term in Eq. (14a) can be computed by looping over the particles in $\mathcal{F}_i$ for each particle i. The bond is skipped if it previously broke. If both particles i, j are owned by the processor, the bond is also skipped if $i > j$, tak-
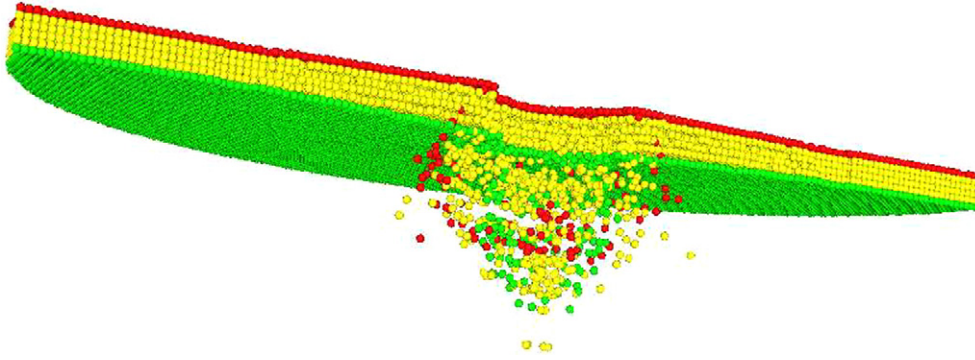
Fig. 1. Cut view of target after impact by a projectile.

ing advantage of Newton's 3rd law. For active bonds, the bond is flagged as "broken" if the bond strain $s(t, \boldsymbol{\eta}, \boldsymbol{\xi})$ exceeds the current strain criterion s_0 for either particle i or j as defined in (9). While bond forces are computed, a new strain criterion s_0 is also calculated, which will be used to break bonds on the next timestep. Note that the *min* operation in (9) implies a loop over all bonds of particle i. As each bond is calculated, the contribution to the new s_0 of both the i and j particles is accumulated. Thus at the end of the bond loop, each particle has a new s_0, valid for the next timestep. Since the bond partner list stores all bonds for each particle owned by a processor, no extra communication is necessary to generate s_0, e.g., due to bonds with ghost particles.

The bond partner list is another particle property that must migrate with particles as they move to new processors. This is done by packing and unpacking the $\mathcal{F}_i$ set of bond neighbors and distances into a message-passing buffer each time a particle migrates. During this operation, broken bonds are pruned from the list, so that a minimal amount of information is communicated.

Once the short-range and bond terms of (14a) have been computed, a final scaling by the volume V_i of each particle is applied. This results in an effective force on each particle that can be used by a standard MD time integrator (velocity Verlet in the case of LAMMPS) in the usual way to update particle velocities and coordinates. For PD models we use a constant NVE integrator. Thermostatting is not used, since temperature is an ill-defined quantity for macroscopic PD particles. The "pressure" due to PD interactions can be computed via the virial in the usual MD manner, except that the kinetic energy term contributing to the pressure is ignored. Alternatively, a precise notion of PD stress has been formulated in [6] and may be used.

5. Numerical experiments

To validate the new additions to LAMMPS against an existing PD code, EMU [18], the experiment in Section 6 of [15] was performed. Consider the impact of a rigid sphere on a homogeneous block of brittle material. The sphere has diameter 0.01 m and velocity of 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 strain parameters given by $s_{00} = 0.0005$ and $\alpha = 0.25$. The target was created as a 3D cubic lattice of particles with lattice constant $a = 0.0005$ m and horizon distance $\delta = 0.0015$ m $= 3a$. The target is a cylinder of diameter 0.074 m and thickness 0.0025 m, and contains 103,110 particles. Each particle i has volume $V_i = a^3 = 1.25 \times 10^{-10}$ m^3.

The stiffness constant c in the PMB material model was set to

$$c = \frac{18k}{\pi \delta^4} = \frac{18(14.9 \times 10^9)}{\pi (1.5 \times 10^{-3})^4} \approx 1.6863 \times 10^{22}. \qquad (15)$$

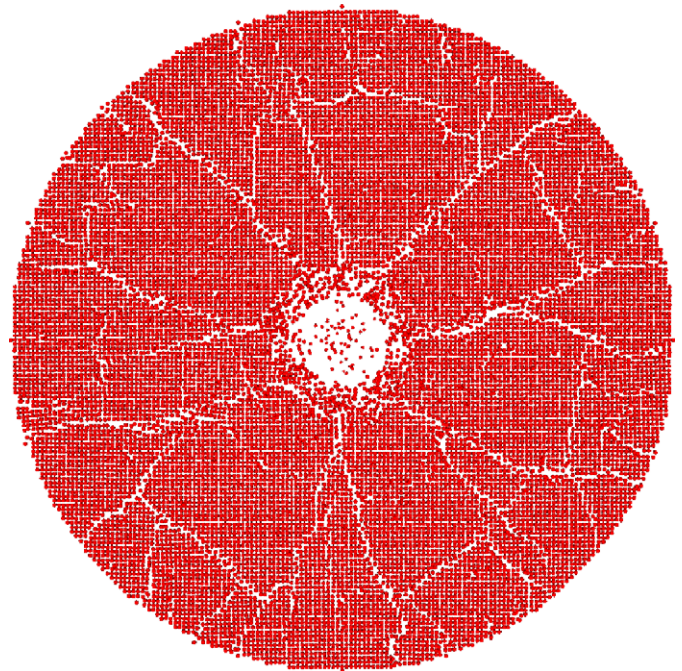The timestep was set to 1.0×10^{-9} seconds.

Fig. 2. Top monolayer of brittle target showing fragmentation.

The projectile used in the LAMMPS simulation was similar, but not identical to the one used in [15]. The projectile was modeled as an indenter, exerting a force

$$F(r) = -k_s (r - R)^2$$

on each particle, where k_s is a specified force constant, r is the distance from the particle to the center of the indenter, and R is the radius of the indenter. The force is repulsive and $F(r) = 0$ for $r > R$. For our problem, the projectile radius was $R = 0.05$ m and $k_s = 1.0 \times 10^{17}$ (compare with (15) above).

A 200,000 timestep simulation was performed. A sample cut view of the disk (projectile not shown) appears in Fig. 1, showing the debris cloud that results from the impact. An image of the top monolayer of particles at the end of the simulation is illustrated in Fig. 2, showing fracture of the brittle target. These results agree qualitatively with EMU, both in the size and shape distributions of the resulting fragments.

To break symmetries, each particle in the initial lattice was randomly perturbed by a distance no more than 10% of the lattice constant. These perturbed positions were used as the reference configuration for each particle. Such random perturbations are routinely included as a way of incorporating the inherent randomness in the distribution of defects in real materials. However, these per-
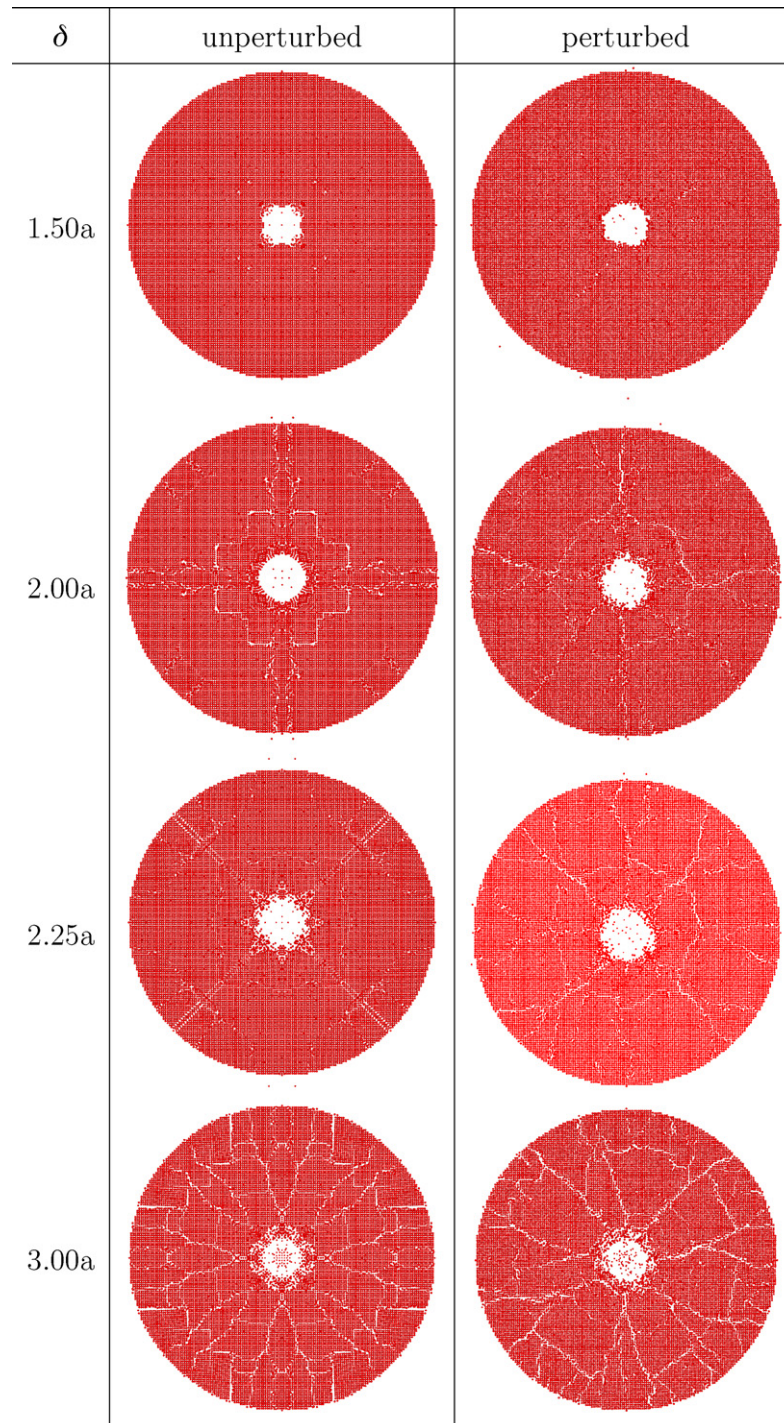
Fig. 3. Top monolayer of target after impact for different values of δ, and for initially unperturbed and perturbed meshes. For sufficiently large δ, crack growth is arbitrary. Perturbation of the initial mesh acts only to break symmetry of the solution.

turbations have only a minor effect on crack trajectory, and are not needed to reproduce crack growth directions as dictated by loading conditions.

To demonstrate further the effects of symmetry-breaking and the peridynamic horizon δ upon the solution, we show in Fig. 3 this numerical experiment repeated for several values of δ, where the initial mesh is either perturbed or unperturbed. Qualitatively, the results do not depend on regularity of the lattice for large enough δ. However, a symmetric initial mesh acted upon by a symmetric projectile produces a symmetric solution, as is required.

Finally, we compare the serial and parallel performance of PD within LAMMPS to that of a standard Lennard-Jones (LJ) model. For the PD calculation, a non-periodic cube of size 2.3 cm on a side was simulated, using roughly $100,000$ particles on a simple cubic lattice. The same PD material parameters described in the previous section were used. For the LJ calculation, a 3D periodic cube with the same number of particles was used to represent a solid. The LJ cutoff was set to 3σ so that the number of neighbors per particle roughly matched the number of bond partners per particle in the PD system. Benchmark runs were performed on a large Linux cluster built consisting of 3.6 GHz Intel EM64T processors

Table 1
CPU timings (in seconds) for 10,000 timesteps of 100,000-particle peridynamics (PD) and Lennard-Jones (LJ) systems

Number of processors	LJ (sec)	LJ speedup	PD (sec)	PD speedup
1	4053.0	1.0	17302.6	1.0
2	2157.9	1.9	9028.4	1.9
4	1133.3	3.6	4673.6	3.7
8	587.6	6.9	2413.4	7.2
16	317.7	12.8	1327.9	13.0
32	172.5	23.5	714.6	24.2
64	94.0	43.1	380.2	45.5

and an Infiniband communication network with 230 Mb/sec and 9 μsec bandwidth and latency performance for point-to-point MPI message passing.

The timing results are presented in Table 1 for runs of 10,000 timesteps each, on processor counts from 1 to 64. The results indicate the PD potential is about four times more expensive to compute than a LJ potential. Similar parallel scalability for both models was observed.

6. Conclusions

Peridynamics (PD) is a continuum theory based on a nonlocal force model. We have shown that the inter-particle forces that result from discretizing PD have a functional form analogous to interatomic potentials commonly used in molecular dynamics (MD). We have demonstrated that PD can be implemented within an MD framework. Enhancing an MD code in such a way allows users familiar with MD to effectively simulate continuum material. The PD extensions made to the LAMMPS MD package are available for download from the LAMMPS WWW site http://lammps.sandia.gov. For more details on using the code, see the user guide [12].

Future work involves computing thermodynamic quantities such as temperature, computing the PD stress given by [6], implementing the more general PD state theory [17] to go beyond central force interactions, and the simulation of classical elasticity given by the results of [19]. The latter capability enables multiscale simulation because both molecular dynamics and classical elasticity can be performed within LAMMPS.

Acknowledgements

References

[1] Z.P. Bažant, M. Jirásek, Nonlocal integral formulations of plasticity and damage: Survey of progress, J. Eng. Mech. 128 (2002) 1119–1149.
[2] F. Bobaru, Influence of van der Waals forces on increasing the strength and toughness in dynamic fracture of nanofibre networks: A peridynamic approach, Modelling and Simulation in Materials Science and Engineering 15 (2007) 397–417.
[3] Y. Chen, J.D. Lee, A. Eskandarian, Atomistic viewpoint of the applicability of microcontinuum theories, Int. J. Solids Structures 41 (2004) 2085–2097.
[4] L.M. Dupuy, E.B. Tadmor, R.E. Miller, R. Phillips, Finite-temperature quasicontinuum: Molecular dynamics without all the atoms, Phys. Rev. Lett. 95 (6) (2005) 060202.
[5] R.W. Hockney, S.P. Goel, J.W. Eastwood, Quiet high-resolution computer models of a plasma, J. Comp. Phys. 14 (1974) 148–158.
[6] R.B. Lehoucq, S.A. Silling, Force flux and the peridynamic stress tensor, J. Mech. Phys. Solids 56 (2008) 1566–1577. Available at doi:10.1016/j.jmps.2007.08.004.
[7] J.E. Lennard-Jones, Cohesion, Proc. Phys. Soc. 43 (5) (1931) 461–482.
[8] S. Li, W.K. Liu, Meshfree Particle Methods, Springer-Verlag, 2004.
[9] B.Q. Luan, S. Hyun, J.F. Molinari, N. Bernstein, M.O. Robbins, Multiscale modeling of two-dimensional contacts, Phys. Rev. E (Statistical, Nonlinear, and Soft Matter Physics) 74 (4) (2006) 046710.
[10] R.E. Miller, E.B. Tadmor, Hybrid continuum mechanics and atomistic methods for simulating materials deformation and failure, in: J.J. de Pablo, W.A. Curtin (Eds.), Multiscale Modeling in Advanced Materials Research, in: MRS Bulletin, vol. 32, November 2007, pp. 920–926. Available at http://www.mrs.org/s_mrs/sec_subscribe.asp?CID=11404&DID=202874&action=detail.
[11] J.T. Padding, A.A. Louis, Hydrodynamic interactions and Brownian forces in colloidal suspensions: Coarse-graining over time and length scales, Phys. Rev. E (Statistical, Nonlinear, and Soft Matter Physics) 74 (3) (2006) 031402.
[12] M.L. Parks, S.J. Plimpton, R.B. Lehoucq, S.A. Silling, Peridynamics with LAMMPS: A user guide, Technical Report SAND 2008-1035, Sandia National Laboratories, January 2008. Available at http://www.sandia.gov/~mlparks.
[13] S.J. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comp. Phys. 117 (1995) 1–19. Available at http://lammps.sandia.gov.
[14] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, J. Mech. Phys. Solids 48 (2000) 175–209. Available at doi:10.1016/S0022-5096(99)00029-0.
[15] S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics, Comp. Struct. 83 (2005) 1526–1535. Available at doi:10.1016/j.compstruc.2004.11.026.
[16] S.A. Silling, F. Bobaru, Peridynamic modeling of membranes and fibers, Int. J. Non-Linear Mech. 40 (2005) 395–409. Available at doi:10.1016/j.ijnonlinmec.2004.08.004.
[17] S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, J. Elasticity 88 (2007) 151–184. Available at doi:10.1007/s10659-007-9125-1.
[18] S.A. Silling, EMU webpage, http://www.sandia.gov/emu/emu.htm.
[19] S.A. Silling, R.B. Lehoucq, Convergence of peridynamics to classical elasticity theory, J. Elasticity (2008), in press. Available at doi:10.1007/s10659-008-9163-3.
[20] L. Verlet, Computer experiments on classical fluids: I. Thermodynamical properties of Lennard–Jones molecules, Phys. Rev. 159 (1967) 98–103.