

Discretized peridynamics for brittle and ductile solids

Wenyang Liu and Jung-Wuk Hong^{*,†}

Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824, U.S.A.

SUMMARY

Peridynamics is a theory of continuum mechanics expressed in forms of integral equations rather than partial differential equations. In this paper, a peridynamics code is implemented using a graphics processing unit for highly parallel computation, and numerical studies are conducted to investigate the responses of brittle and ductile material models. Stress–strain behavior with different grid sizes and horizons is studied for a brittle material model. A comparison of stresses and strains between finite element analysis (FEA) and peridynamic solutions is performed for a ductile material. By applying the proposed procedure to bridge the material model defined for peridynamic bonds and the corresponding macroscale material model for FEA, peridynamics and FEA show good agreements as regards the stresses and strains. Copyright © 2011 John Wiley & Sons, Ltd.

Received 27 April 2011; Revised 30 June 2011; Accepted 7 July 2011

KEY WORDS: peridynamics; nonlocal; multiscale; GPU computing

1. INTRODUCTION

Predictions of crack initiation and growth have been considered as one of the most difficult problems because the conventional theory is formulated in partial differential equations, and a unique spatial derivative, however, does not exist on cracks. Therefore, a variety of methodologies have been proposed. Belytschko *et al.* [1] presented a minimal remeshing finite element method (FEM) for crack growth by adding discontinuity enriching functions. Moës *et al.* [2] presented a new finite element technique for modeling cracks using the partition of unity method and developed the extended FEM to model the growth of arbitrary cohesive cracks [3]. Foulk *et al.* [4] implemented a cohesive zone model in nonlinear finite element formulation. Kozicki *et al.* [5] showed a lattice-based discrete approach to model fracture in brittle materials, and Gils *et al.* [6] used a new energy-based failure index for delamination analysis. De *et al.* [7] presented development of the method of finite spheres. Hong *et al.* [8,9] used analytical transformation before the numerical integration to improve the method of finite sphere and proposed a technique to couple finite element and finite sphere discretizations.

Different from finite element analysis (FEA), in which elements are connected by a topological mesh, meshfree particle methods employ a finite number of discrete particles to describe the state of a system [10]. The difficulties for simulating many engineering problems such as fragmentation and fluid–structure interaction include remeshing and mapping the state variables from the old mesh to the new mesh [11]. In contrast, meshfree particle methods can easily handle large deformations and material failures. The meshfree particle methods can be classified into microscopic, mesoscopic and macroscopic meshfree particle methods according to the length scales. Holian *et al.* [12] simulated

^{*}Correspondence to: Jung-Wuk Hong, Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824, U.S.A.

[†]E-mail: jwh@egr.msu.edu

the opening-mode fracture under the tensile loading using molecular dynamics. Chen *et al.* [13] presented large deformation analysis of non-linear elastic and inelastic structures based on reproducing kernel particle method (RKPM). The application of RKPM includes, for example, elastic-plastic deformation and hyperelasticity [14].

Peridynamics is a new formulation of continuum mechanics [15]. The mathematical description of continuum mechanics is reformulated to integral equations rather than partial differential equations [16]. In peridynamics, the force is computed on a material particle, and peridynamics could be considered as a continuum version of molecular dynamics [17]. The consideration of interacting forces between particles over certain distance renders the methodology in the category of the nonlocal theory [18–20]. Peridynamics can simulate fractures and other discontinuities without any information to specify the direction of the crack growth. The definitions of stress and strain fields have not been established as in classical elasticity although some efforts have been made to establish a connection between two approaches [21]. The convergence of peridynamics towards the linear elasticity has been studied by several researchers [22, 23]. Researchers have applied peridynamics to study material responses such as deformations of a peridynamics bar and concrete structures under impact and explosive loadings [24–29]. Other advantages of peridynamics include that the peridynamics can be implemented within the framework of molecular dynamics [30]. In addition, the methodology can be coupled with conventional FEM [31]. Peridynamics has limitations in representing continuum material properties. It can only simulate materials that have Poisson's ratios of $1/4$. To extend the capability of peridynamics to model various materials that have different material properties, a generalized peridynamics was proposed, which directly incorporates a constitutive model from conventional solid mechanics [32–35]. Compared with macroscopic modeling such as FEA, three-dimensional (3D) peridynamic analysis of large simulation is computationally expensive. Macek *et al.* [31] compared computer wall-clock run times between the EMU and FEA implementations of peridynamics. The FEA implementation is much faster than the direct meshless method. Askari *et al.* [36] applied peridynamics to model material and structural failures using the Columbia Supercomputer at NASA Advanced Supercomputing Division.

In this paper, a peridynamics code for 3D simulations is implemented using an NVIDIA Tesla C1060 GPU (Santa Clara, CA, USA) for parallelization. A significant speedup over the serial calculation in a CPU is achieved, varying with the number of particles in peridynamic models. The effects of different horizons and grid sizes are investigated for a brittle material. The stresses and strains of FEA and peridynamic solutions, respectively, are compared for a ductile material. A multiscale procedure is proposed to bridge plastic simulations at different scales. The results after applying the proposed procedure show good agreements between FEA and peridynamics.

2. THEORY

2.1. Peridynamic formulation

The equation of motion in the conventional continuum mechanics is derived from the principle of linear momentum that the rate of change of linear momentum equals the force applied on the body as [37]

$$\rho \ddot{\mathbf{u}} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{b}, \quad (1)$$

where ρ is the density, $\ddot{\mathbf{u}}$ is the acceleration vector, $\boldsymbol{\sigma}$ is the stress matrix, and \mathbf{b} is the body force vector. The formulation requires a unique spatial derivative, which, however, does not exist along discontinuities. In contrast, peridynamics uses integration to compute the force on a material particle [16]. The equation of motion of the particle I in the reference configuration as shown in Figure 1 is written as [38]

$$\rho \ddot{\mathbf{u}}(\mathbf{x}_I, t) = \int_{\mathcal{H}_I} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_J + \mathbf{b}(\mathbf{x}_I, t), \quad (2)$$

where \mathcal{H}_I is the neighborhood of the particle I , ρ is the mass density in the reference configuration, \mathbf{f} is a pairwise force vector that the particle J exerts on the particle I , \mathbf{x}_I denotes the coordinate

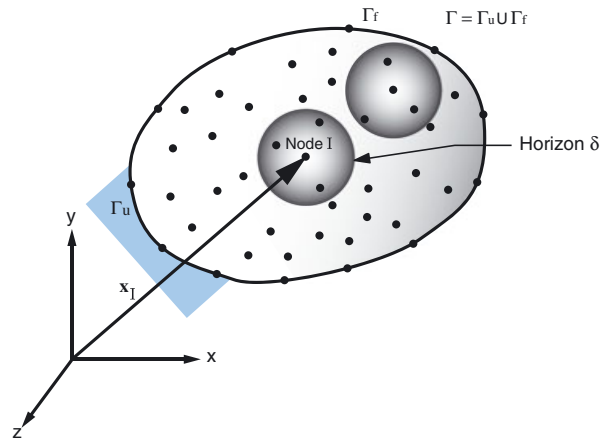


Figure 1. Schematic of peridynamics.

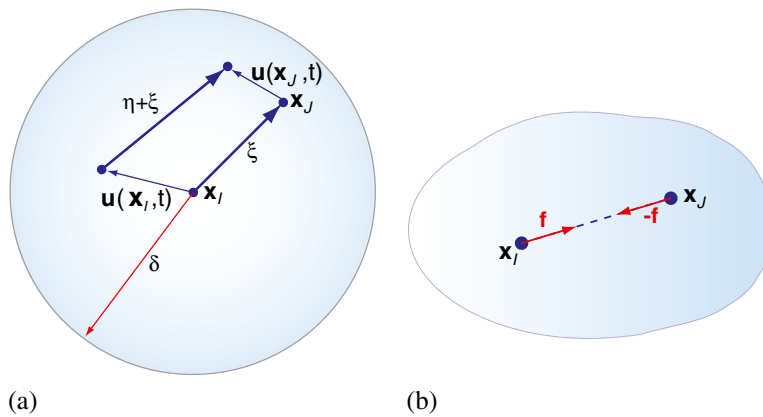


Figure 2. (a) Relationships among relative position vector and the relative displacement vector within a peridynamic horizon; (b) pairwise force vector.

of the particle I , and \mathbf{x}_J is the coordinate of the particle J that interacts with the particle I . The displacement of the particle I at time t is expressed as $\mathbf{u}(\mathbf{x}_I, t)$.

The relative position vector of two particles in the reference configuration shown in Figure 2 is expressed as [17]

$$\boldsymbol{\xi} = \mathbf{x}_J - \mathbf{x}_I, \quad (3)$$

and the relative displacement vector at time t is written as

$$\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}_J, t) - \mathbf{u}(\mathbf{x}_I, t). \quad (4)$$

For each material, a scalar δ , called the horizon, is assumed to exist to determine the interacting spatial range between the particles I and J such that

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \mathbf{0} \quad \forall \boldsymbol{\eta}, \quad \text{if } \|\boldsymbol{\xi}\| > \delta. \quad (5)$$

The pairwise force vector \mathbf{f} has the direction of the connection between the particle I and the particle J as

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = f(\boldsymbol{\eta}, \boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{\|\boldsymbol{\eta} + \boldsymbol{\xi}\|}, \quad (6)$$

where f is a scalar-valued pairwise force function, and $\|\cdot\|$ is the Euclidean norm. For the particle J within the horizon of the particle I , the scalar-valued pairwise force function for elastic materials is expressed as

$$f(\eta, \xi) = c \cdot s(t, \eta, \xi), \quad (7)$$

where c is the micromodulus, s is the bond stretch between particles possessing a similar concept of the strain in elasticity. The bond stretch s is defined as

$$s(t, \eta, \xi) = \frac{\|\eta + \xi\| - \|\xi\|}{\|\xi\|}, \quad (8)$$

where $\|\xi\|$ is the original bond length in the reference coordinate, and $\|\eta + \xi\|$ is the current bond length between two particles. If the stretch $s = 0$, then there is no pairwise force f between the particles.

Figure 3(a) shows a brittle material model defined for peridynamics. In the elastic regime, the bond force is a scalar function of the bond stretch s , and the critical stretch for bond failure is denoted as s_0 . Once a bond fails, it cannot sustain force any more, which makes the peridynamic model hysteretic. Figure 3(b) defines a ductile material model for peridynamics. A bond yields at the yield limit s_y , and shows perfect plasticity up to s_0 .

To form the peridynamic equation of motion, the material domain is discretized with a number of nodes as shown in Figure 4. The distances of two adjacent nodes are identical over the domain and denoted as Δx . Therefore, the volume representation for each node is $(\Delta x)^3$. The peridynamic

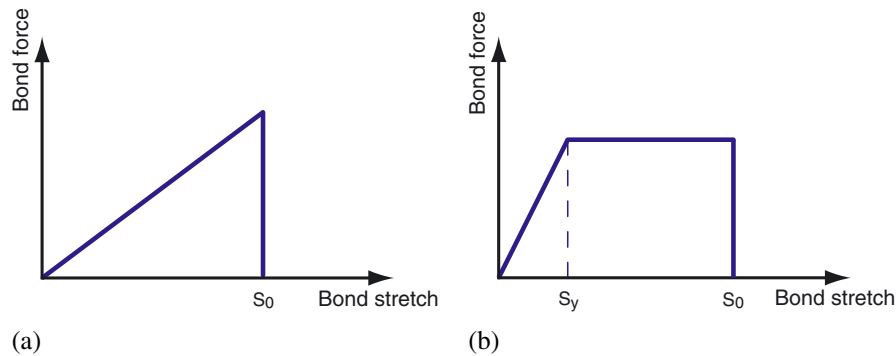


Figure 3. (a) Bond force as a function of bond stretch in a brittle material model; (b) bond force as a function of bond stretch in a ductile material model.

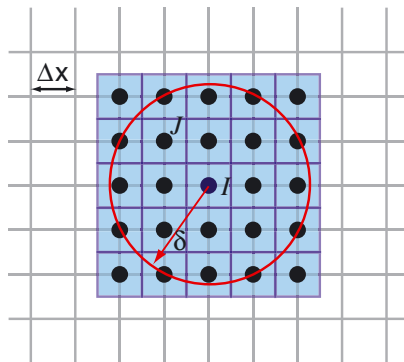


Figure 4. Discretized domain for computation.

equation of motion after discretization is written as

$$\rho \ddot{\mathbf{u}}_I^t = \sum_{J=1}^{N_{\mathcal{H}_I}} \mathbf{f}(\eta^t, \xi) V_J + \mathbf{b}_I^t, \quad (9)$$

where $\ddot{\mathbf{u}}_I^t$ is the acceleration of the particle I at time t , $\mathbf{f}(\eta^t, \xi)$ is the pairwise force, $N_{\mathcal{H}_I}$ is the total number of particles within the horizon of the particle I , and \mathbf{b}_I^t is the body force at time t .

To consider the volume reduction of a particle that has an intersection with the horizon boundary as illustrated in Figure 5(a), a volume reduction scheme is introduced [38] as follows:

$$V_J(\|\xi\|) = \begin{cases} \left(\frac{\delta - \|\xi\|}{2r_j} + \frac{1}{2} \right) V_J & \text{if } (\delta - r_j) \leq \|\xi\| \leq \delta \\ V_J & \text{if } \|\xi\| \leq (\delta - r_j) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $(\delta - r_j)$ is the distance from which the volume is reduced, and r_j is chosen to be half of the grid spacing Δx in the numerical implementation. Figure 5(b) illustrates the volume change of the particle J as the distance between the particle I and the particle J increases.

2.2. Micromodulus for three-dimensional discretized peridynamics

In the 3D domain, an elastic body is subjected to the external force such that the bond stretch is a constant value s in the domain. The bonds connected to each volume are determined for a specific horizon, (e.g., $\delta = 2\Delta x$). The stress σ_x on the cross section of the particle I at the coordinate \mathbf{x}_I can be obtained by adding all the bond forces passing through or ending at the cross section A_I as shown in Figure 6. Figure 7 represents the number of bonds passing through or ending at the cross section for different horizons. The number of the bonds for the horizon $\delta = 2\Delta x$ is 11, and it increases to 631 for the horizon $\delta = 5\Delta x$.

The projection of the pairwise force \mathbf{f} in x -direction is expressed as

$$f_x = f \frac{|x_J - x_I|}{\|\xi\|}, \quad (11)$$

where $f = c_3 s$ is the magnitude of the pairwise force between the particle I and the particle J , c_3 is the micromodulus for 3D model, and $|x_J - x_I|$ is the distance between two particles in the x -coordinate. Considering all particles have the same volume $(\Delta x)^3$, the stress σ_x on the cross section of the particle I is given by

$$\sigma_x = \frac{1}{A_I} \sum_{J=1}^{N_c} (f_x V_J) V_I, \quad (12)$$

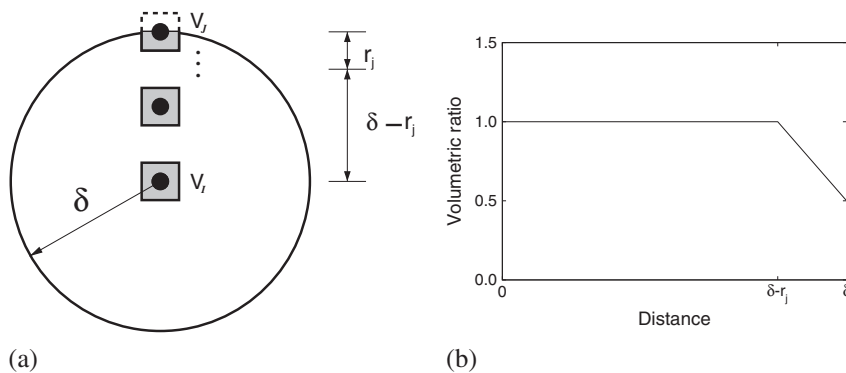


Figure 5. (a) Volume calculation scheme to consider reduced volume on the boundary of a horizon; (b) volumetric ratio in a horizon [38].

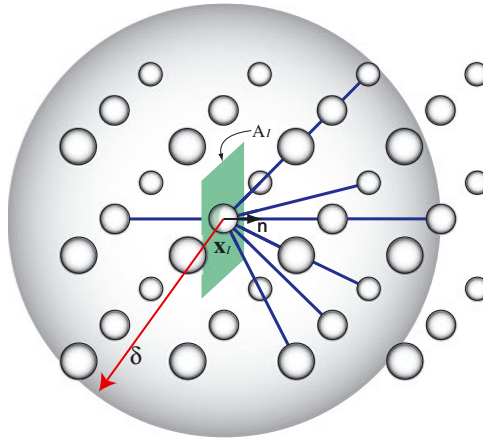


Figure 6. Pairwise forces acting on a cross section of a particle in 3D domain. Each node, represented by a sphere, has a volume of $(\Delta x)^3$, and $\delta = 2\Delta x$.

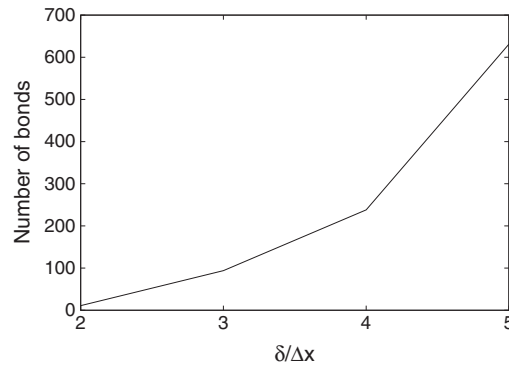


Figure 7. Number of bonds acting on a cross section of a particle in the three-dimensional domain.

where N_L is the number of bonds passing through or ending at the cross section of a particle, and the cross-sectional area $A_I = (\Delta x)^2$ as shown in Figure 6. Because the elastic body is subjected to the isotropic expansion, the stresses in y and z directions are identical to the x -directional stress σ_x , expressed as $\sigma_y = \sigma_z = \sigma_x$. By applying Hooke's law in the linear elasticity, the peridynamic Young's modulus E_{pd} can be obtained as

$$E_{pd} = \frac{\sigma_x}{s} - \frac{\nu(\sigma_y + \sigma_z)}{s}, \quad (13)$$

where Poisson's ratio $\nu = 1/4$. Setting the peridynamic Young's modulus E_{pd} in Equation (13) identical to the Young's modulus of the material E , the micromodulus c_3 , by substituting Equations (11) and (12) to Equation (13), can be calculated. For example, the micromodulus for the horizon $\delta = 2\Delta x$ is

$$c_3 = 0.302942 \frac{E}{(\Delta x)^4}. \quad (14)$$

The micromoduli c_3 for the horizons from $2\Delta x$ to $5\Delta x$ are summarized in Table I. Figure 8 shows the comparison of micromoduli c_3 and $\tilde{c}_3 = \frac{18k}{\pi\delta^4}$ [17, 19, 22] as the horizon increases for an elastic material that has Young's modulus $E = 70$ GPa and Poisson's ratio $\nu = 1/4$. Compared with the micromodulus \tilde{c}_3 , which is obtained by the analytical calculation, the numerically determined micromodulus c_3 is 1.27 times larger than \tilde{c}_3 for the horizon $\delta = 2\Delta x$. As the horizon δ increases, c_3 converges to the analytically obtained micromodulus \tilde{c}_3 .

Table I. Micromodulus c_3 in three-dimensional domain.

Horizon δ	Micromodulus c_3 (N/m ⁶)
$2\Delta x$	$0.302942 \frac{E}{\Delta x^4}$
$3\Delta x$	$0.052385 \frac{E}{\Delta x^4}$
$4\Delta x$	$0.017290 \frac{E}{\Delta x^4}$
$5\Delta x$	$0.006819 \frac{E}{\Delta x^4}$

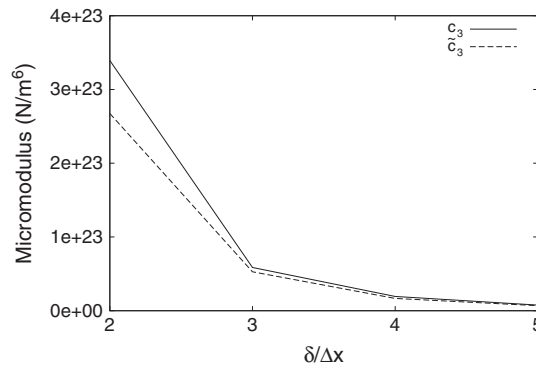


Figure 8. Comparison of micromoduli for different horizons ($E = 70$ GPa, and $\nu = 1/4$). c_3 is given in Table I and $\tilde{c}_3 = \frac{18k}{\pi \delta^4}$.

3. COMPUTATION

3.1. Numerical implementation

To calculate the bond forces on peridynamic particles after discretization in Equation (9), a neighbor list is created for each particle in the reference configuration. In the subsequent calculations, the neighbor list is referred to determine the total bond force acting on a particle. The Velocity Verlet scheme [39] is used for the time integration to update positions, velocities, and accelerations as

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \quad (15)$$

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}(t)\Delta t, \quad (16)$$

$$\mathbf{a}(t) = \frac{1}{\rho} \mathbf{f}_V(t), \quad (17)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \Delta t/2) + \frac{1}{2}\mathbf{a}(t + \Delta t)\Delta t, \quad (18)$$

where \mathbf{x} is the position vector, \mathbf{v} is the velocity vector, \mathbf{a} is the acceleration vector, and \mathbf{f}_V is the scaled force vector, which is defined as

$$\mathbf{f}_V = \sum_{J=1}^{N_{\mathcal{H}_I}} \mathbf{f}(t) V_J. \quad (19)$$

The pairwise bond force \mathbf{f} is decided by all neighbors within the horizon of a particle. The acceleration is calculated by dividing the scaled pairwise force \mathbf{f}_V , which has the unit of force per volume, by the density ρ .

3.2. Implementation of peridynamics in GPU

Graphics processing units, originally developed for the acceleration of graphics rendering, have a large number of transistors devoted to data processing rather than data caching and flow control [40]. This advantage has been taken for numerical computations. GPUs have evolved for many data-parallel and computing-intensive programs, and a variety of compilers have been developed. Researchers have used GPUs for highly parallel computation. Harris [41] described a method to simulate stable fluid using a GPU. Kruger *et al.* [42] introduced a framework for solving sets of algebraic equations in GPUs. Manavski *et al.* [43] used GPUs as an accelerator for Smith–Waterman sequence alignment. Phillips *et al.* [44] implemented a multi-block turbulent flow solver in GPU processors.

We use a GPU, NVIDIA Tesla C1060, that has 4 GB memory and 30 multiprocessors, and each multiprocessor has eight cores. Therefore, 240 cores are available in total. Each core can execute a sequential thread, and the clock rate is 1296 MHz. Figure 9 shows the block diagram of a GPU accelerator [45]. The GPU has its own memory, which is called the device memory, and the accelerator communicates with CPU using I/O commands and the direct memory access transfer [46]. The data bandwidth is 512-bits wide. To port the peridynamics code to a GPU platform, PGI Fortran accelerator programming model [47, 48], a high-level implicit programming model for the general purpose computation on GPUs, is utilized. A set of compiler directives designed for GPUs is available to specify regions of the code to be offloaded from the CPU to the GPU. The code implemented with accelerator directives is portable even without a GPU unit because the directives are considered as comments by other Fortran compilers.

A benchmark problem for multiplication of a matrix $\mathbf{A}_{n \times n}$ and a vector $\mathbf{b}_{n \times 1}$ is written with directives as:

```
!$acc region
do k=1,1000,1
  do i =1,n
    r(i)=0
    do j = 1,n
```

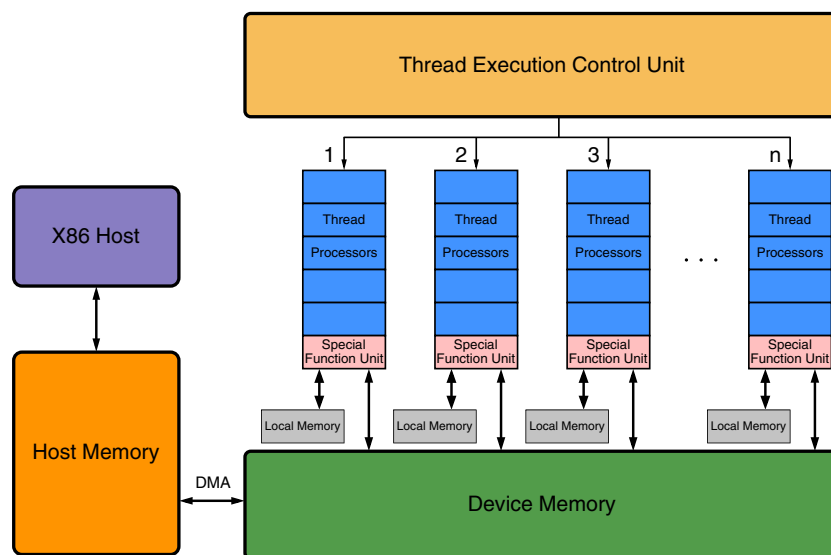


Figure 9. NVIDIA Tesla block diagram [46].


```

      r(i)=r(i)+A(i,j)*b(j)
    enddo
  enddo
enddo
!$acc end region

```

In the fixed format, PGI Accelerator directives are specified with the sentinel `!$acc`. The directive `!$acc region` defines the region to be compiled for execution in the GPU [48]. For benchmarking, the multiplication is repeated one thousand times, and the summation is stored in a vector **r**. Figure 10 compares the wall-clock times by serial calculation in the CPU and by the parallelized calculation in the GPU. For the matrix of the size n smaller than 5000, the GPU calculation is merely a few times faster than the serial calculation. However, as the size of the matrix increases, the speedup of the GPU calculation over the serial calculation increases significantly.

For the effective parallelization in GPUs, all the necessary information in the main memory is required to be transferred to the GPU memory for the calculation. We use the data clauses to control the transfer of selected data between the CPU and the GPU, which reduces the system overhead in copying data. The peridynamic model is initialized in the CPU, and then variables and arrays are copied to the GPU for the calculation within the data region using directives as

```

!$acc data region copyin(list), local(list)
!$acc updateout(list)
!$acc end data region

```

The clause `copyin(list)` is used to copy the listed variables and arrays from the CPU to the GPU memory, and the clause `local(list)` declares local variables and arrays to be allocated only in the GPU memory [48]. Calculation results in the GPU are copied to the main memory once per multiple calculation steps using the directive `!$acc updateout(list)`. It should be noted that copying the whole array is faster than copying the noncontiguous subarrays [46].

In the subroutine to calculate bond forces on each particle in a discretized peridynamic model, the conventional algorithm includes two dependent loops as shown in Algorithm 1. However, the loop dependency disables optimal parallelization on GPUs. Thus, the nested loops are separated into two independent loops as shown in Algorithm 2. Peridynamics codes using Algorithms 1 and 2, respectively, are compiled with PGFORTRAN v10.3 for the serial calculation in the CPU. We compared the wall-clock times to calculate bond forces in a 3D peridynamic model, which has 4,725 particles with the horizon $\delta = 2\Delta x$. As summarized in Table II, it takes 13 113 milliseconds to complete 1 time-step using Algorithm 1 for bond force calculations, and Algorithm 2 spends 21 080 milliseconds. By explicitly setting the memory considering the maximum number of neighbors in the peridynamic model, Algorithm 2 consumes more memory and time than Algorithm 1 if no GPU is utilized. Overall, Algorithm 1 is 1.61 times faster than Algorithm 2 in the CPU.

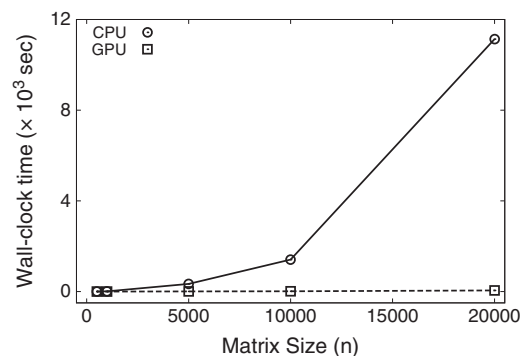


Figure 10. Benchmark result of matrix multiplication.

Algorithm 1 Dependent loops.

```

do i = 1, all particles

    M = number of neighbors of particle i

    {Calculate bond forces on particle i}

    do j = 1, M

        f(i) = f(i) + c*s

    end do

end do

```

Algorithm 2 Independent loops.

```

N = all particles

M = maximum number of neighbors of all particles

{Calculate each bond force}

do i = 1, N*M

    f_bond(i)=c*s

end do

{Sort bond forces to each particle}

do j = 1, all particles

    f(j) = summation of f_bond sorted on particle j

end do

```

Table II. Wall-clock time to calculate bond forces for one time-step using a node in the CPU. The three-dimensional peridynamic model has 4725 particles, the horizon $\delta = 2\Delta x$ and $\Delta x = 0.5$ mm.

	Algorithm 1	Algorithm 2
Wall-clock time (millisecond)	13 113	21 080

Table III. Wall-clock time to run peridynamic models for 500 000 time-steps on GPU and CPU, respectively. Peridynamic models have the horizon $\delta = 2\Delta x$ and $\Delta x = 0.5$ mm.

Number of particles	Time (minute) by the GPU	Time (minute) by the CPU
$11 \times 8 \times 8$	3.5	19.5
$21 \times 15 \times 15$	10.5	144.0
$41 \times 29 \times 29$	65.7	1294.1

For the comparison of the performance by the GPU-enabled calculation using Algorithm 2 and the serial calculation executed in the CPU using Algorithm 1, 3D peridynamic simulations are conducted, and the wall-clock times are summarized in Table III. With the grid size $\Delta x = 0.5$ mm, the first peridynamic model consists of $11 \times 8 \times 8$ particles, the second model consists of $21 \times 15 \times 15$ particles, and the third model consists of $41 \times 29 \times 29$ particles. Young's modulus E is 70 GPa, Poisson's ratio is 0.25, and the critical bond stretch is $s_0 = 0.004$. The time-step is set to $dt = 1 \times 10^{-8}$ s.

Each simulation is carried out for 500 000 time-steps. Figure 11(a) shows the wall-clock times to run the peridynamic models. For the first peridynamic model, which has 704 particles, the GPU calculation takes 3.5 min, whereas the serial calculation spends 19.5 min in the CPU. The GPU calculation is 5.6 times faster than the serial calculation of the peridynamics code. For the second peridynamic model, which has 4725 particles, the GPU calculation spends 10.5 min, whereas the serial calculation spends 144.0 min. For the third peridynamic model, which has 34,481 particles, the GPU calculation spends 65.7 min, whereas the serial calculation uses 1294.1 min. The GPU calculation is 19.7 times faster than the serial calculation. In general, as the number of particles in a peridynamic model increases, the speedup of the GPU enabled version of the peridynamics code over the serial version in the CPU increases as shown in Figure 11(b).

4. CASE STUDIES

4.1. Brittle material

In the peridynamic framework, a 3D rectangular bar is modeled as shown in Figure 12(a). The dimensions of the peridynamic bar are 10 mm \times 7 mm \times 7 mm. Boundary conditions are set by imposing a constant velocity to the particles on both ends of the bar such that $v_x = -10$ mm/s at the left-end boundary region and $v_x = 10$ mm/s at the right-end boundary region. For the material, Young's modulus E is 70 GPa, and Poisson's ratio ν is 0.25. Time-step dt is set to $dt = 5 \times 10^{-8}$ s. To simulate brittle failure of materials, the critical stretch s_0 for bond failure is set as 0.01.

To study the response of the peridynamic bar for different horizons, the rectangular bar is uniformly discretized with particles distributed in a $21 \times 15 \times 15$ grid. The grid size Δx is 0.5 mm, and the horizons δ of $2\Delta x$, $3\Delta x$ and $4\Delta x$ are used in the model. To calculate the cross-sectional stress,

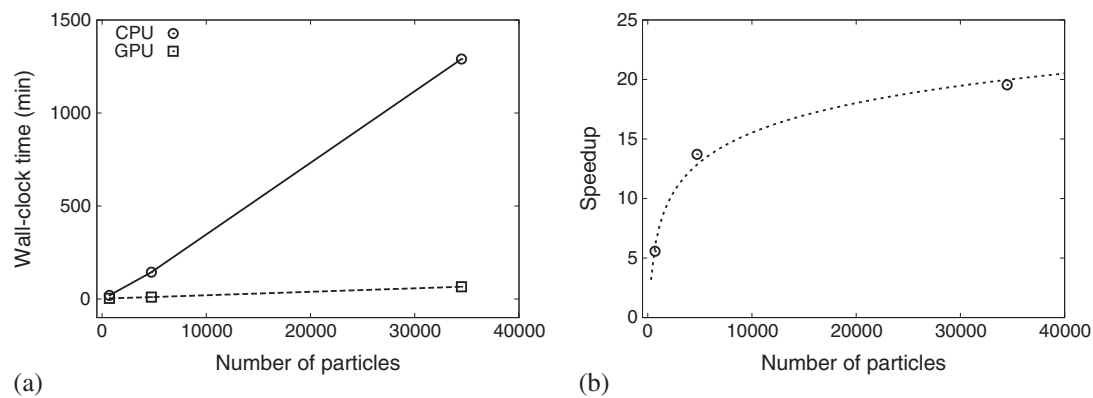


Figure 11. Comparison between the GPU enabled version of peridynamics code and the serial version. (a) Wall-clock time; (b) the speedup of the GPU version over the serial version.

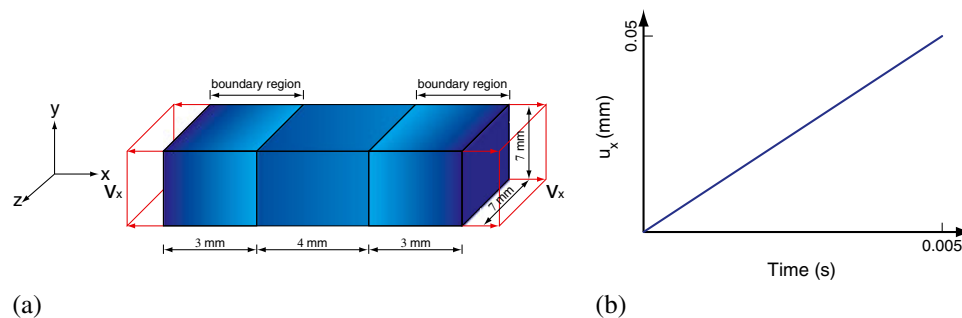


Figure 12. (a) Geometry of a three-dimensional bar and the boundary conditions; (b) displacement of the boundary region versus time.

the summation of all interparticle forces passing through the cross section of the bar is projected on the x -axis. Then the cross-sectional stress is obtained by dividing the resultant force by the sectional area. The macroscopic engineering strain is measured by dividing the variation of the length by the original length. Figure 13(a) shows the stress σ_x versus the strain ε_x on the cross section at $x = 5.0$ mm for different horizons. By calculating the tangent of the stress-strain curves for the horizon $\delta = 2\Delta x$ in Figure 13(a), the back-calculated Young's modulus E is 70.46 GPa. Compared with the exact value $E = 70.00$ GPa of the material, the error by the back-calculated Young's modulus is 0.66%. For the horizons $\delta = 3\Delta x$ and $\delta = 4\Delta x$, the back-calculated Young's moduli are smaller than the exact value as shown in Figure 13(a).

To investigate the effect of the grid size, the peridynamic bar is discretized in a $41 \times 29 \times 29$ grid with a grid size $\Delta x = 0.25$ mm and in a $21 \times 15 \times 15$ grid with a grid size $\Delta x = 0.50$ mm, respectively. Figure 13(b) compares the cross-sectional stress σ_x at $x = 5.0$ mm for the grid size $\Delta x = 0.25$ mm and $\Delta x = 0.50$ mm, where the horizon δ is set as $2\Delta x$. As evident in Figure 13(b), the back-calculated Young's modulus E for the grid size $\Delta x = 0.25$ mm is very close to the back-calculated Young's modulus E for the grid size $\Delta x = 0.50$ mm. Because stresses and strains are not intrinsic variables in the peridynamic theory, the bond stretches within the grid width Δx are averaged for each particle. Strain distributions at times $t = 1.0 \times 10^{-3}$ s, $t = 1.6 \times 10^{-3}$ s, and $t = 1.8 \times 10^{-3}$ s denoted by I (no bond failure), II (a few bond failure) and III (massive bond failure) in Figure 13(b), respectively, are plotted for the rough and fine grids as shown in Figures 14 to 16. As the strain contours demonstrate, strain ε_x is about 0.005 at time $t = 1.0 \times 10^{-3}$ s as shown in Figure 14. At time $t = 1.6 \times 10^{-3}$ s, the cross-sectional stress σ_x reaches its maximum value, and the strain ε_x near the boundary regions is larger than 0.01 as shown in Figure 15(b). On the cross section at $z = 3.5$ mm, those particles having the nodal strain ε_x in the range of 0.007–0.010 form diagonal patterns as shown in Figure 15(d). At time $t = 1.8 \times 10^{-3}$ s, when the cross-sectional stress σ_x significantly decreases because of massive bond failure, particles having the nodal strain ε_x over 0.010 form an elliptical distribution on the cross section at $z = 3.5$ mm as shown in Figures 16(c) and (e). On the cross section at $x = 5.0$ mm, interior particles have the nodal strain ε_x over 0.01, and a failure core is formed as shown in Figures 16(d) and (f).

4.2. Ductile material

We consider a rectangular bar subjected to a stretch by imposing a constant velocity on particles located at the boundary regions such that $v_x = -10$ mm/s at the left-end boundary region and $v_x = 10$ mm/s at the right-end boundary region as shown in Figure 12(a). The dimensions of the bar are 10 mm in length, 7 mm in width, and 7 mm in thickness. The bar is uniformly discretized with

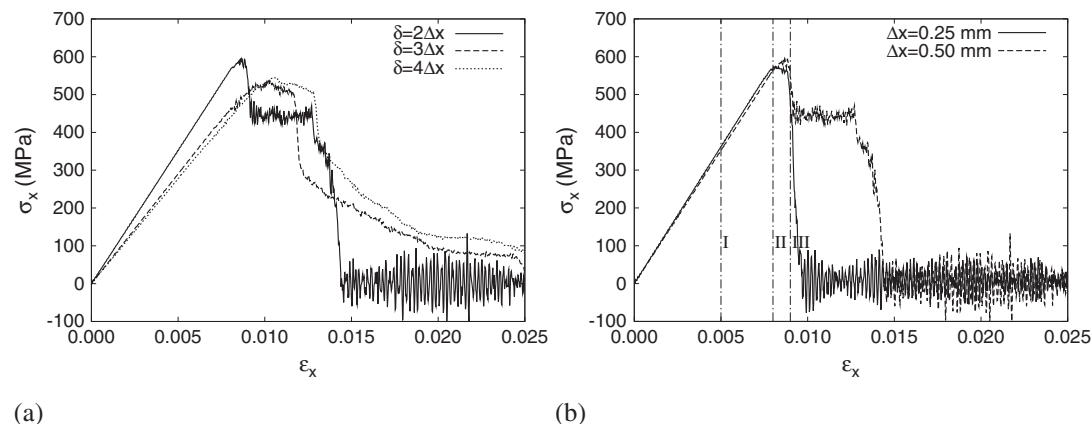


Figure 13. Stress on the cross section at $x = 5.0$ mm for (a) different horizons δ with the grid size $\Delta x = 0.5$ mm and (b) different grid sizes Δx with the horizon $\delta = 2\Delta x$. ε_x is the engineering strain imposed by the displacement at both ends of the bar.

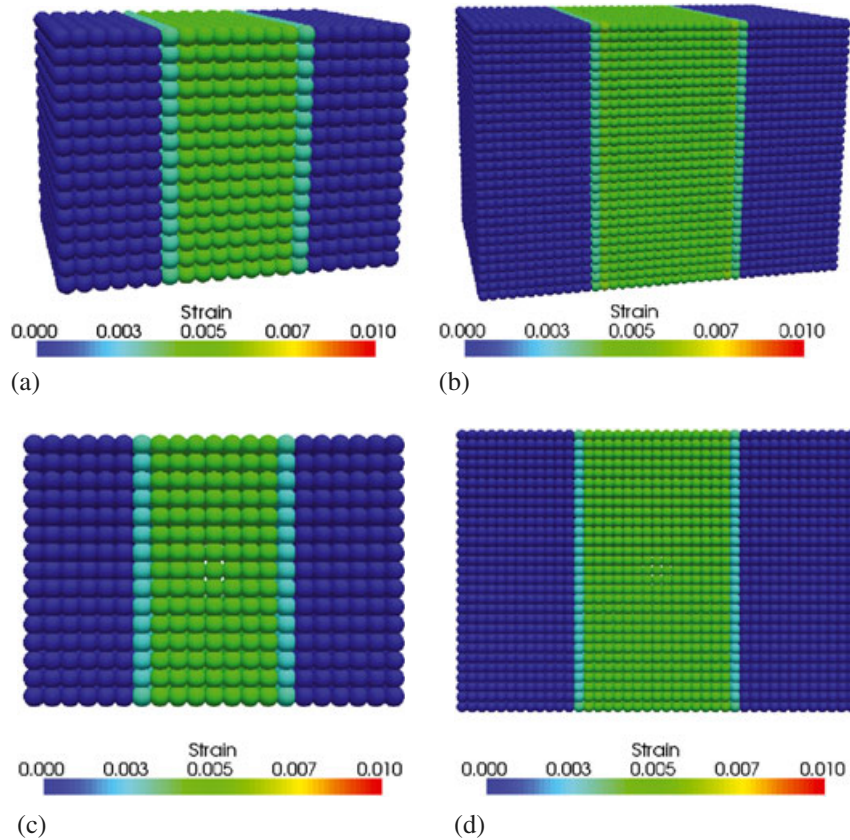


Figure 14. Strain ε_x at time $t = 1.0 \times 10^{-3}$ s, denoted by I in Figure 13(b). The horizon $\delta = 2\Delta x$. (a) $\Delta x = 0.50$ mm; (b) $\Delta x = 0.25$ mm; (c) the cross section at $z = 3.5$ mm, $\Delta x = 0.50$ mm; (d) the cross section at $z = 3.5$ mm, $\Delta x = 0.25$ mm.

particles in a $21 \times 15 \times 15$ grid, and the horizon $\delta = 2\Delta x$. Young's modulus E is 70 GPa, Poisson's ratio ν is 0.25, and the density ρ is 2700 kg/m³. Interparticle bonds are assumed to yield at the bond stretch $s_y = 0.004$, and the critical stretch for bonds to break is set as $s_0 = 0.02$. The calculation time-step is $dt = 5 \times 10^{-8}$ s.

The peridynamics results are compared with the results by FEA using a commercial FEA software, LS-DYNA (Livermore Software Technology Corp., Livermore, CA, USA) [49]. In the FEA, the identical boundary conditions to those of the peridynamic model are imposed, and the time-step dt is automatically calculated. The material model used in the FEA is identical to the material model in peridynamics: Young's modulus E is 70 GPa, and Poisson's ratio ν is 0.25. The yield stress σ_y is 280 MPa obtained by multiplying the yield bond stretch $s_y = 0.004$ and Young's modulus. The limit strain is set to be 0.020, and the corresponding plastic strain is 0.016. Figure 17(a) shows the material properties implemented in LS-DYNA.

Figure 17(b) compares the average stress on the cross section at $x = 5.0$ mm by peridynamics and FEA. After the yielding, the cross-sectional stress continues to increase as the boundary displacement u_x increases in the peridynamic model, whereas the FE model shows almost perfect plasticity. The cross-sectional stress demonstrates a trilinear response in the peridynamics. As shown in Figure 17(b), the end of the second piecewise linear line approximately corresponds to the displacement $u_x = 0.017$ mm, and the corresponding stress is $\sigma_x = 420$ MPa. Again, for the stress of a particle in the peridynamic model, the total bond forces acting through the cross section of a particle are calculated. Figure 17(c) shows the comparison of the stress σ_x of the element that has

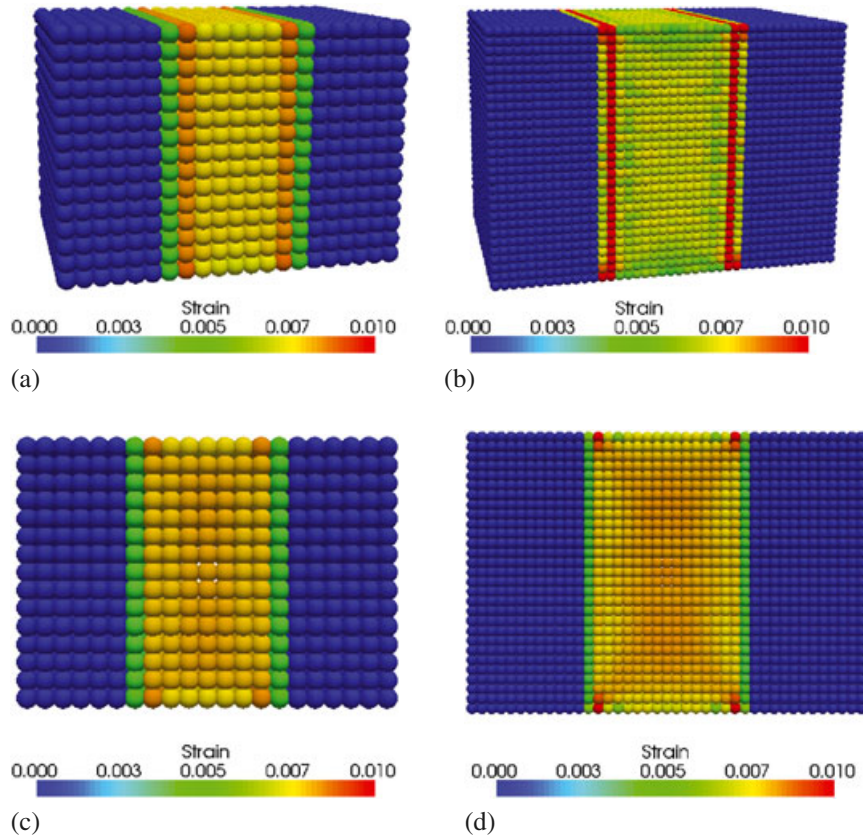


Figure 15. Strain ε_x at time $t = 1.6 \times 10^{-3}$ s, denoted by II in Figure 13(b). The horizon $\delta = 2\Delta x$. (a) $\Delta x = 0.50$ mm; (b) $\Delta x = 0.25$ mm; (c) the cross section at $z = 3.5$ mm, $\Delta x = 0.50$ mm; (d) the cross section at $z = 3.5$ mm, $\Delta x = 0.25$ mm.

a node at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm and $z = 3.5$ mm) and the stress σ_x of the peridynamic particle located at the center of the peridynamic bar. Similar to the average stress on the cross section, the stress of the peridynamic particle at the center of the bar shows a trilinear response whereas the stress in the element by the FEA has a bilinear response. It is noticed that the outer bonds of a particle yield earlier than the inner bonds in the peridynamic model, and the stiffness of the peridynamic model gets closer to the value of the FE model after all the bonds yield, as shown in Figure 17(c). The nodal strains ε_x at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm, and $z = 3.5$ mm) obtained by peridynamics and FEA are compared in Figure 17(d). The cause of the differences might be the scales at which the material model is defined. Therefore, to bridge the different scales, it is proposed to modify the scheme to define the material in the FEA rather than using the identical material model to peridynamics. The material response obtained in Figure 17(b) is interpolated to a trilinear line as shown in Figure 18(a). The point B is determined from the macroscopic response of the peridynamic model.

A piecewise linear plastic model available in LS-DYNA [49] is used in the FE simulation in which the effective stress versus effective plastic strain $\varepsilon_{\text{eff}}^p$ curve is defined, and the effective plastic strain is given as $\varepsilon_{\text{eff}}^p = \int_0^t \sqrt{\frac{2}{3} \dot{\varepsilon}_{ij}^p \dot{\varepsilon}_{ij}^p} dt$ [50]. The plastic strain starts to increase after the strain reaches the yield strain $\varepsilon = 0.004$. The effective plastic strain can be approximated by integrating the strain rate after yielding, imposed by displacements at both ends. By evaluating the integration, the effective plastic strain $\varepsilon_{\text{eff}}^p$ corresponding to $\sigma_x = 420$ MPa is obtained as $\varepsilon_{\text{eff}}^p = 0.0037$. The average stress on the cross section at $x = 5.0$ mm and the stress at the center point of the section are plotted

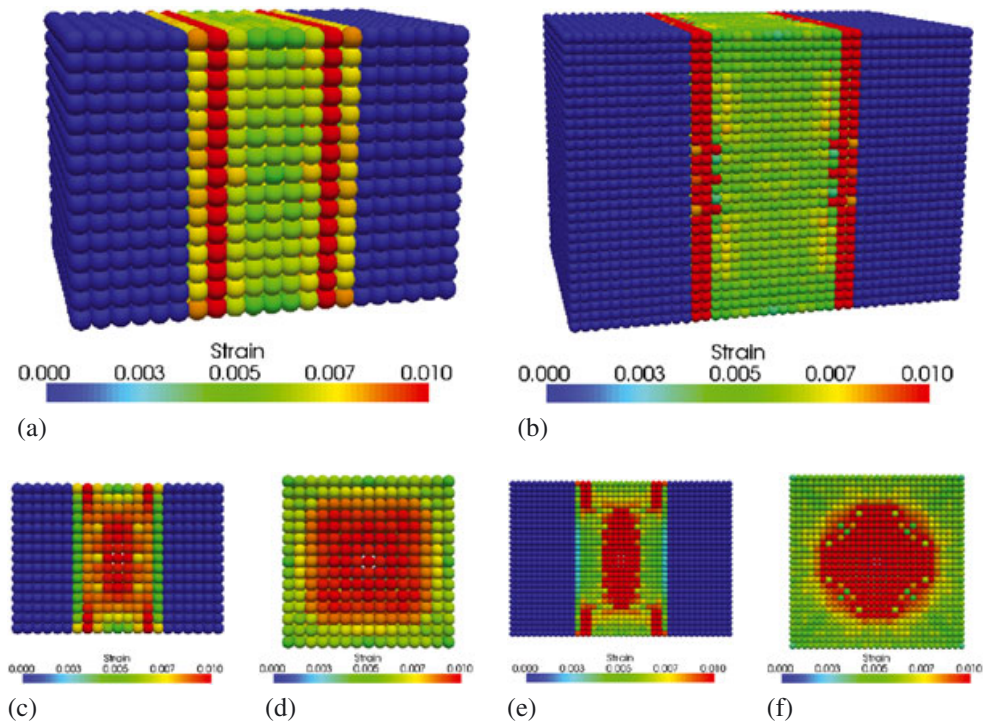


Figure 16. Strain ε_x at time $t = 1.8 \times 10^{-3}$ s, denoted by III in Figure 13(b). The horizon $\delta = 2\Delta x$. (a) $\Delta x = 0.50$ mm; (b) $\Delta x = 0.25$ mm; (c) the cross section at $z = 3.5$ mm, $\Delta x = 0.50$ mm; (d) the cross section at $x = 5.0$ mm, $\Delta x = 0.50$ mm; (e) the cross section at $z = 3.5$ mm, $\Delta x = 0.25$ mm; (f) the cross section at $x = 5.0$ mm, $\Delta x = 0.25$ mm.

in Figures 18(b) and (c). The nodal strains at the center of the bar are compared in Figure 18(d), and the results show good agreement. As shown in Figures 18(b)–(d), the differences between the solutions by peridynamics and FEA are reduced substantially. The developed scheme can bridge the simulations at different scales at which the material model is defined. First, a mesoscale modeling is performed using peridynamics, and material properties are retrieved from the macroscopic material response. Then, the properties are described as a modified material model at the macroscale for the FEA. Strains ε_x by peridynamics and FEA when the displacement at the boundary $u_x = 0.025$ mm ($t = 2.5 \times 10^{-3}$ s) are compared in Figure 19, and the strains show almost identical results.

5. SUMMARY & CONCLUSION

A peridynamics code is developed for 3D simulations using numerical micromoduli for discretized peridynamic models. To enhance the efficiency in the calculation, we use the PGI Accelerator Programming model on an NVIDIA Tesla C1060 GPU for the parallelization. A benchmark problem of matrix multiplication is conducted to compare the wall-clock times between the parallelized algebraic computation on the GPU and the serial computation on the CPU. To eliminate the loop dependency in the peridynamics code for efficient parallelization, an algorithm that separates nested loops into independent loops is adopted in the GPU implementation. We compared the wall-clock times of the parallelized calculation in GPU and the serial calculation in CPU for different peridynamic models. As the number of particles in the peridynamic model increases, a significant speedup over the serial calculation is achieved in the GPU.

The responses of the peridynamic model for the horizons δ of $2\Delta x$, $3\Delta x$ and $4\Delta x$ are investigated for a brittle material. For the horizon $\delta = 2\Delta x$, the back-calculated Young's modulus matches the

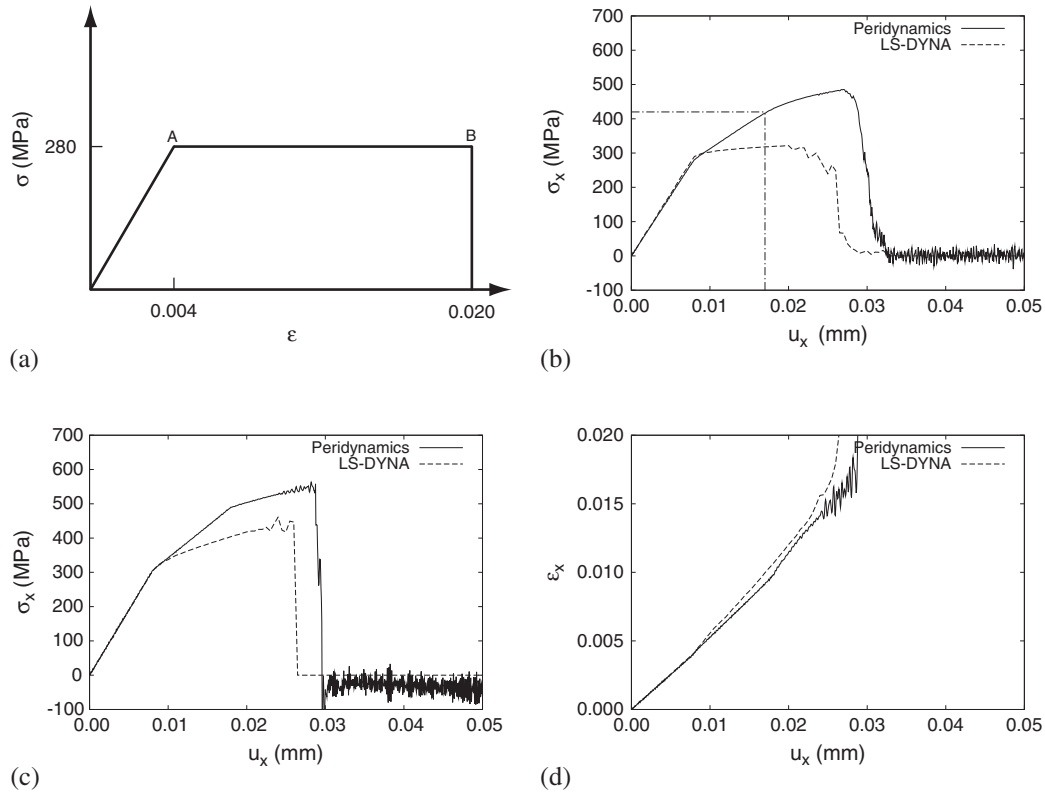


Figure 17. Comparison between peridynamic and FEA solutions. (a) The material model used in FEA, which corresponds to the material model for bonds in peridynamics; (b) Stress σ_x on the cross section at $x = 5.0$ mm; (c) Stress σ_x on an element, which has a node at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm and $z = 3.5$ mm) and the stress σ_x on the peridynamic particle at the center of the bar; (d) Strain ϵ_x at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm and $z = 3.5$ mm). u_x is the displacement of the boundary region as shown in Figure 12.

exact value of the material. On the other hand, for the horizon larger than $2\Delta x$, the back-calculated Young's moduli are smaller than the exact value. With the horizon $\delta = 2\Delta x$, the responses of the peridynamic bar discretized with different grid sizes ($\Delta x = 0.50$ mm and $\Delta x = 0.25$ mm) are compared, and numerical results of the back-calculated Young's modulus show good agreements. The strain distributions at different times in the simulations are demonstrated. After a large number of bonds reach the critical stretch, a failure core is formed inside the peridynamic model.

For a ductile material, the peridynamic stresses and strains are compared with the results by FEA using LS-DYNA. When the material model used in FEA is identical to that of peridynamics, the stresses of FEA show bilinear responses, whereas peridynamics demonstrates trilinear responses with the horizon $\delta = 2\Delta x$. In addition, the nodal strains by FEA and peridynamics show some difference at the center of the bar. Therefore, to bridge material models at different scales, the material properties are retrieved from the macroscopic response of the peridynamic model. These retrieved properties are used to modify the material model in FEA. By applying the proposed procedure, the differences between FEA and peridynamic solutions reduce substantially. This proposed scheme will be useful for coupling peridynamics and FEM that might be implemented in the future. Instead of directly applying the material model defined for peridynamic bonds to FEA in a coupled model, which may not lead to an accurate result, the plastic material model in FEA might be obtained from the macroscopic responses of peridynamic models.

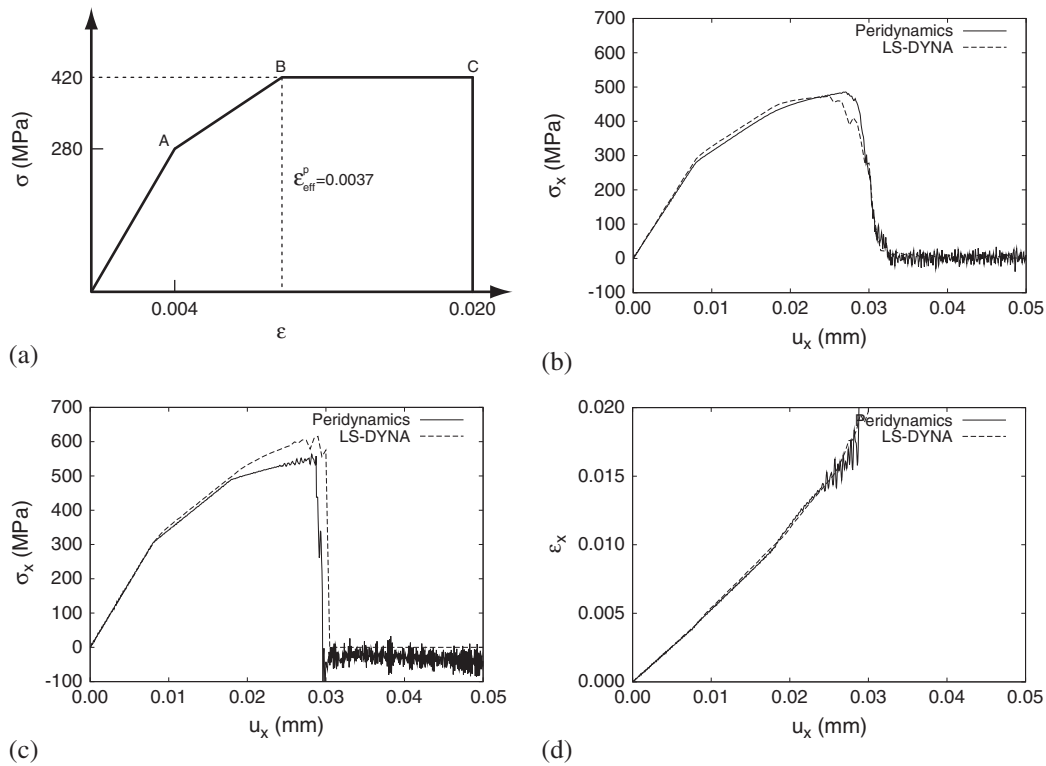


Figure 18. Comparison between peridynamic and FEA solutions. (a) The modified material model used in FEA to incorporate retrieved macroscopic material properties from the peridynamic model; (b) Stress σ_x on the cross section at $x = 5.0$ mm; (c) Stress σ_x on an element that has a node at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm and $z = 3.5$ mm) and the stress σ_x on the peridynamic particle at the center of the bar; (d) Strain ϵ_x at the center of the bar ($x = 5.0$ mm, $y = 3.5$ mm and $z = 3.5$ mm). u_x is the displacement of the boundary region as shown in Figure 12.

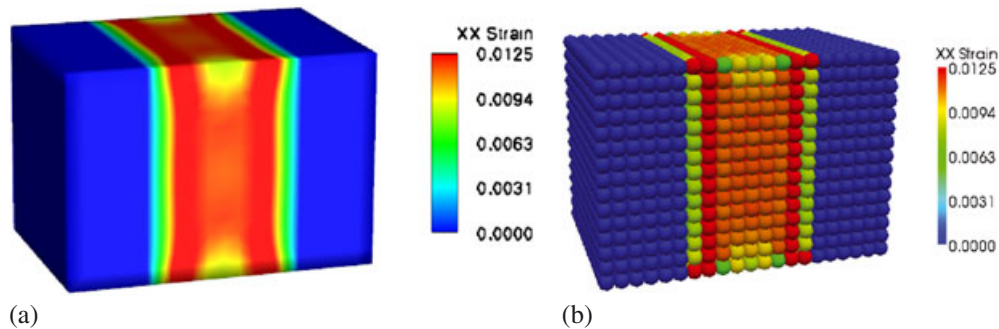


Figure 19. Strain ϵ_x when the boundary displacement $u_x = 0.025$ mm. (a) LS-DYNA and (b) peridynamics.

ACKNOWLEDGEMENTS

The authors thank Air Force Office of Scientific Research (Grant FA 9550-10-1-0222) for their generous support of this research.

REFERENCES

1. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**(5):601–620.
2. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**(1):131–150.

3. Moës N, Belytschko T. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics* 2002; **69**(7):813–833.
4. Foulk JW, Allen DH, Helms KLE. Formulation of a three-dimensional cohesive zone model for application to a finite element algorithm. *Computer Methods in Applied Mechanics and Engineering* 2000; **183**(1–2):51–66.
5. Kozicki J, Tejchman J. Modelling of fracture process in concrete using a novel lattice model. *Granular Matter* 2008; **10**(5):377–388.
6. Gils MAJV, van Der Sluis O, Zhang GQ, Janssen JHJ, Voncken RMJ. Analysis of Cu/low-k bond pad delamination by using a novel failure index. *Microelectronics Reliability* 2007; **47**(2–3):179–186.
7. De S, Hong JW, Bathe KJ. On the method of finite spheres in applications: towards the use with ADINA and in a surgical simulator. *Computational Mechanics* 2003; **31**(1):27–37.
8. Hong JW, Bathe KJ. On analytical transformations for efficiency improvements in the method of finite spheres. *Computational fluid and solid mechanics* 2003:1990–1994.
9. Hong JW, Bathe KJ. Coupling and enrichment schemes for finite element and finite sphere discretizations. *Computers & Structures* 2005; **83**(17–18):1386–1395.
10. Liu GR, Liu MB. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Pub Co Inc: Hackensack, NJ, 2003.
11. Li S, Liu WK. Meshfree and particle methods and their applications. *Applied Mechanics Reviews* 2002; **55**:1.
12. Holian BL, Ravelo R. Fracture simulations using large-scale molecular dynamics. *Physical Review B* May 1995; **51**(17):11275–11288. DOI: 10.1103/PhysRevB.51.11275.
13. Chen JS, Pan C, Wu CT, Liu WK. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1–4):195–227.
14. Liu WK, Chen Y, Jun S, Chen JS, Belytschko T, Pan C, Uras RA, Chang CT. Overview and applications of the reproducing kernel particle methods. *Archives of Computational Methods in Engineering* 1996; **3**(1):3–80.
15. Silling SA, Lehoucq RB. Peridynamic theory of solid mechanics. *Advances in Applied Mechanics* 2010; **44**:73–168.
16. Silling SA. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* 2000; **48**(1):175–209.
17. Silling SA, Askari E. A meshfree method based on the peridynamic model of solid mechanics. *Computers & Structures* 2005; **83**(17–18):1526–1535.
18. Edelen DGB, Green AE, Laws N. Nonlocal continuum mechanics. *Archive for Rational Mechanics and Analysis* 1971; **43**(1):36–44.
19. Emmrich E, Weckner O. The peridynamic equation of motion in non-local elasticity theory. *III European Conference on Computational Mechanics. Solids, Structures and Coupled Problems in Engineering*, Lisbon, Springer, 2006; vol. 19.
20. Weckner O, Abeyaratne R. The effect of long-range forces on the dynamics of a bar. *Journal of the Mechanics and Physics of Solids* 2005; **53**(3):705–728.
21. Lehoucq RB, Silling SA. Force flux and the peridynamic stress tensor. *Journal of the Mechanics and Physics of Solids* 2008; **56**(4):1566–1577.
22. Emmrich E, Weckner O. On the well-posedness of the linear peridynamic model and its convergence towards the Navier equation of linear elasticity. *Communications in Mathematical Sciences* 2007; **5**(4):851–864.
23. Silling SA, Lehoucq RB. Convergence of peridynamics to classical elasticity theory. *Journal of Elasticity* 2008; **93**(1):13–37. DOI: 10.1007/s10659-008-9163-3.
24. Silling SA, Bobaru F. Peridynamic modeling of membranes and fibers. *International Journal of Non-Linear Mechanics* 2005; **40**(2–3):395–409.
25. Dayal K, Bhattacharya K. Kinetics of phase transformations in the peridynamic formulation of continuum mechanics. *Journal of the Mechanics and Physics of Solids* 2006; **54**(9):1811–1842. DOI: 10.1016/j.jmps.2006.04.001.
26. Demmie PN, Silling SA. An approach to modeling extreme loading of structures using peridynamics. *Journal of Mechanics of Materials and Structures* 2007; **2**(10):1921–1945.
27. Gerstle W, Sau N, Silling SA. Peridynamic modeling of plain and reinforced concrete structures. *SMIRT18: 18th International Conference on Structural Mechanics in Reactor Technology*, Beijing, 2005.
28. Bobaru F. 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* 2007; **15**:397.
29. Kilic B, Madenci E. Prediction of crack paths in a quenched glass plate by using peridynamic theory. *International Journal of Fracture* 2009; **156**(2):165–177.
30. Parks ML, Lehoucq RB, Plimpton SJ, Silling SA. Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications* 2008; **179**(11):777–783.
31. Macek RW, Silling SA. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design* 2007; **43**(15):1169–1178.
32. Silling SA, Epton M, Weckner O, Xu J, Askari E. Peridynamic states and constitutive modeling. *Journal of Elasticity* 2007; **88**(2):151–184. DOI: 10.1007/s10659-007-9125-1.
33. Silling SA. Linearized theory of peridynamic states. *Journal of Elasticity* 2010; **99**(1):85–111.
34. Warren TL, Silling SA, Askari A, Weckner O, Epton MA, Xu J. A non-ordinary state-based peridynamic method to model solid material deformation and fracture. *International Journal of Solids and Structures* 2009; **46**(5):1186–1195.

35. Foster JT, Silling SA, Chen WW. Viscoplasticity using peridynamics. *International Journal for Numerical Methods in Engineering* 2009; **81**(10):1242–1258.
36. Askari E, Xu J, Silling S. Peridynamic analysis of damage and failure in composites. *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada. Reston VA: AIAA*, 2006.
37. Mase GT, Smelser RE, Mase GE. *Continuum Mechanics for Engineers*. CRC: Boca Raton, FL, 1999.
38. Parks ML, Plimpton SJ, Lehoucq RB, Silling SA. Peridynamics with LAMMPS: A user guide. *Technical Report SAND 2008-1035*, Sandia National Laboratories, 2008.
39. Ercolessi F. A molecular dynamics primer. *Spring College in Computational Physics, ICTP, Trieste* 1997:24–25.
40. NVIDIA. *NVIDIA CUDA Programming Guide*, 2009. Version 2.3.1 edn.
41. Harris MJ. Fast fluid dynamics simulation on the GPU. *GPU gems* 2004; **1**:637–665.
42. Kruger J, Westermann R. Linear algebra operators for GPU implementation of numerical algorithms. In *ACM SIGGRAPH 2005 Courses*. ACM: New York, NY, 2005; 234.
43. Manavski S, Valle G. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC bioinformatics* 2008; **9**(Suppl 2):S10.
44. Phillips EH, Davis RL, Owens JD. Unsteady turbulent simulations on a cluster of graphics processors. *Proceedings of the 40th AIAA Fluid Dynamics Conference*, 2010. AIAA-2010-5036, 40th AIAA Fluid Dynamics Conference.
45. Wolfe M. Understanding the CUDA data parallel threading model. *Technical Report*, The Portland Group, 2010.
46. Wolfe M. The PGI accelerator programming model on NVIDIA GPUs. *Technical Report*, The Portland Group, 2009.
47. The Portland Group. *PGI User's Guide*, 2010th ed. The Portland Group, Inc: Lake Oswego, OR, 2010.
48. The Portland Group. *PGI Fortran & C Accelerator Programming Model*, 3rd ed., Vol. 1.2.1.2. The Portland Group, Inc: Lake Oswego, OR, 2010.
49. Livermore Software Technology Corporation. *LS-DYNA keyword user's manual*, 05 2007.
50. Hallquist JO. *LS-DYNA theory manual*. Livermore Software Technology Corporation: Livermore, California, March 2006.