QUASISTATIC FRACTURE USING NONLINEAR-NONLOCAL ELASTOSTATICS WITH EXPLICIT TANGENT STIFFNESS MATRIX

A PREPRINT

Patrick Diehl
Center for Computation & Technology, Louisiana State University, Baton Rouge, LA

Robert Lipton
Department of Mathematics, Louisiana State University, Baton Rouge, LA
Center for Computation & Technology, Louisiana State University, Baton Rouge, LA

April 4, 2022

ABSTRACT

We apply a nonlinear-nonlocal field theory for numerical calculation of quasistatic fracture. The model is given by a regularized nonlinear pairwise (RNP) potential in a peridynamic formulation. The potential function is given by an explicit formula with an explicit first and second derivatives. This fact allows us to write the entries of the tangent stiffness matrix explicitly thereby saving computational costs during the assembly of the tangent stiffness matrix. We validate our approach against classical continuum mechanics for the linear elastic material behavior. In addition, we compare our approach to a state-based peridynamic model that uses standard numerical derivations to assemble the tangent stiffness matrix. The numerical experiments show that for elastic material behavior our approach agrees with both classical continuum mechanics and the state-based model. The fracture model is applied to produce a fracture simulation for a ASTM E8 like tension test. We conclude with an example of crack growth in a pre-cracked square plate. For the pre-cracked plate, we investigated *load in force* (soft loading) and *load in displacement* (hard loading). Our approach is novel in that only bond softening is used as opposed to bond breaking. For the fracture simulation we have shown that our approach works with and without initial damage for two common test problems.

1 Introduction

Peridynamic theory (PD)[64, 66], is a non-local formulation of continuum mechanics that autonomously nucleates and propagates fracture. The formulation successfully captures qualitative features seen in a range of experiments[22, 21]. As of this writing, it appears that only a few quasi static PD simulations are available[21, 32, 54, 73, 70, 7, 40, 58, 25]. For quasi-static problems, there is no inertia and time is represented by a load parameter. One significant reason behind the paucity of quasi-static PD simulations is the additional computational expense in going from dynamic methods with explicit time integration $\mathcal{O}(n^2)$ to a quasi-static implementation with an implicit time integration $\mathcal{O}(n^4)$, where n is the number of discrete PD nodes. The major expense here is the assembly of the tangent stiffness matrix $\mathcal{O}(n^4)$.

Several methods were proposed to speed up the time integration. Finite element approaches (FEM) for PD[10, 24, 52] were applied and found to reduce the computational costs for the assembly of the tangent stiffness matrix to $\mathcal{O}(n^3)$. For purely elastic problems, Wang[71] developed a Galerkin method that exploits the matrix structure and reduces the costs of solving the matrix system from $\mathcal{O}(n^3)$ to $\mathcal{O}(n\log^2(n))$. In another direction, Chen[10] proposed a simplified model to reduce the computational costs to $\mathcal{O}(n)$ but with a reduced convergence rate of only first order for linear (FEM). Prakash[56] presents an algorithm using sparse matrices for the assembly of the tangent stiffness matrix instead of a dense matrix. The sparse algorithms scaled approximately linearly with the problem size, whereas the dense matrix algorithm scales non-linearly as one would expect. The performance of the sparse implementation is compared with an adaptive dynamic relaxation scheme (ADR) in[9, 67, 40]. It is found that a speed-up factor between 12 and 22 against the ADR solve is achievable. Shiihara[63] implemented the Fire algorithm[6] which

originated from molecular dynamics within the PD framework. The Fire algorithm converges about 100 time faster than the energy based relaxation method. Hu[31, 30] solved the linear equation system iteratively using the GMRES algorithm[60] in conjunction with the Arnoldi process[2, 61]. Another approach to speed-up nonlocal diffusion[35] or peridynamics[34] is the convolution-based method, which reduces the computational costs to $\mathcal{O}(nlog_2(n))$.

In this paper, we propose a different approach. We apply an explicit regularized nonlinear pairwise (RNP) potential[42, 48, 45] in our peridynamic formulation. The potential function is given by explicit formulas with explicit derivatives. This fact allows us to write the entries of the tangent stiffness matrix using explicit formulas, thereby saving computational costs during the assembly of the tangent stiffness matrix. The reduction of the computaitonal costs comes from the fact that we do not need to compute the numerical derivation for the entries of the tangent stiffness matrix. In the absence of fracture, numerical simulations using our model recover linear elastostatic material behavior. We compare simulations using our approach to those using a state-based peridynamic model with standard numerical derivations to assemble the tangent stiffness matrix. The numerical experiments show that for elastic material behavior, our approach agrees with both classical elastostatic continuum mechanics and the state-based model. Next, the model is applied to a fracture simulation for an ASTM E8 like tension test. The crack initiates at the upper grip section, where force is applied in agreement with theory. However, the crack growth becomes unstable, and the tangent stiffness matrix ceases to be invertible due to extensive localized damage - transverse to the loading direction - that occurs after a critical load step. We conclude with an example of crack growth in a pre-cracked square plate. Here, an external force is the load applied to the grips. For sufficiently large loads the crack starts to grow, and moves forward for a few mesh widths, upon reducing the load the crack grows further in a stable fashion for a few more load steps, however, beyond that even with increasingly small load steps the matrix is unstable. This numerical behavior aligns with the theory of quasistatic fracture mechanics[29, 11, 59]. Next, the load is applied as a displacement. Here, we observe stable crack growth for the pre-cracked square plate. Again, this numerical behavior aligns with the theory of quasistatic fracture mechanics [3]. Our approach is novel in that only bond softening is used, as opposed to bond breaking. We demonstrate that our approach works with and without initial damage for the two common test problems treated here.

The paper is organized as follows: A brief introduction to stable and unstable crack growth is presented in Section 2. The peridynamic model and quasistatic formulation is presented in Section 3. In Section 4, we demonstrate how to obtain the entries of the tangent stiffness matrix explicitly using Taylor series. In Section 5 we describe the new algorithm used to assemble the stiffness matrix and investigate its theoretical complexity. In Section 6, we present the results of our computational experiments. Here, we first display the savings in computational time using our RNP potentials versus non-differential potentials. We then present the validation of the implementation and present results for the mode-I fracture experiment. Concluding remarks and directions for future work are presented in Section 7.

2 On stable and unstable crack growth

The stability or instability of quasi-static crack growth is well known[3]. We briefly review the concepts required to explain stable and unstable crack growth. Stable quasi-static crack growth refers to a continuous increase in crack length with respect to loading; otherwise the crack is unstable and can grow without further applied load, *i.e.* the specimen breaks. In force control the load is applied using a force and in displacement control a displacement is applied to the specimen. A specimen containing a pre-crack of initial length a_0 is considered. Stable growth of the initial crack from its initial length in the quasi-static regime is determined by the material's crack growth resistance curve R. The crack resistance R is plotted with respect to the crack length l with $R = 0$ for $l \leq a_0$. The R curve can be determined experimentally using load in displacement as specified for example in ASTM Standard E 561[5]. Figure 1 sketches a rising R curve (thick black line) which is common for most materials. The crack can grow stably when the energy released G for an increment of crack growth is equal to the material resistance to crack growth R. However, depending on how G varies with the crack length, l, the crack growth is either stable or unstable. The condition for stable crack growth reads as

$$G = R \quad \text{and} \quad \frac{\partial G}{\partial l} \leq \frac{\partial R}{\partial l} \tag{1}$$

and the condition for unstable crack growth reads as

$$G = R \quad \text{and} \quad \frac{\partial G}{\partial l} > \frac{\partial R}{\partial l}. \tag{2}$$

The simplest case is a step function R curve corresponding to brittle materials. For $a_0 < l$ we have, $R = G_c$ where G_c is the critical energy release rate[26]. Here there is no crack growth for $G < G_c$ and the crack grows stably, if $G = G_c$ and $\frac{\partial G}{\partial l} \leq 0$.

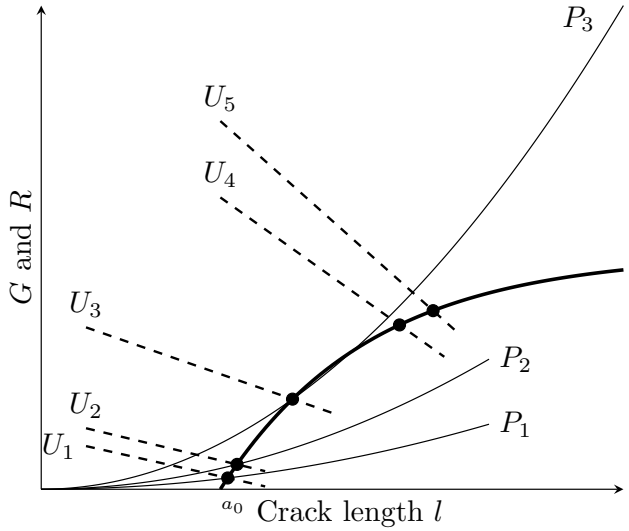


Figure 1: Sketch of a rising R curve (thick black line) which increases with the crack length l. The three thin black curves are the locus of admissible G and l corresponding to a fixed force P applied to the geometry. Here $P_1 < P_2 < P_3$. The crack is stable for force loads P_1, P_2 since the loci are non-tangential to the R curve, which indicates that the force must be increased for the crack develop further. Crack growth is unstable for P_3 since $G \geq R$, and the curve is tangent to the R curve hence no extra force is needed to grow the crack. The dotted curves show the locus of admissible G and l for prescribed displacements $\mathbf{U}_i, i = 1, \ldots, 5$ applied to the boundary. Here the displacements are increasing with i and the associated (G, l) loci are intersecting transversely to the R curve, which indicates that the load in displacement must be increased for the crack develop further. This corresponds to Equations (1), (2) and the crack growth tends to be more stable than in force control. This figure was adapted from Figure 2.11 from the reference[3].

The crack growth stability with respect to force control (load in force) and displacement control (load in displacement) is examined in the context of the R curve. Looking at Equation (1) and Equation (2) the stability of crack growth depends on the change in G with respect to l. Figure 1 sketches the R curve in thick black with respect to the crack length l. The loci of admissible G and l corresponding to a fixed force P and fixed displacement $\mathbf{u}$ are considered. Figure 1 examines these loci and their relation to the R curve, and shows schematically that stable crack growth is easier to obtain under displacement control as opposed to force control. For more details about force/displacement control and R curves, we refer to[3].

3 Peridynamic model and quasistatic formulation

The appeal of peridynamic models is that fracture appears as an emergent phenomenon, eliminating the need for supplemental kinetic relations describing crack growth. The displacement field inside the body for points $\boldsymbol{x}$ at time t is written $\boldsymbol{u}(\boldsymbol{x}, t)$. The peridynamic model is described simply by the balance of linear momentum of the form

$$\rho \boldsymbol{u}_t(\boldsymbol{x}, t) = \int_{\mathcal{H}_\epsilon(\boldsymbol{x})} \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x})\, d\boldsymbol{y} + \boldsymbol{b}(\boldsymbol{x}, t) \tag{3}$$

where $\mathcal{H}_\epsilon(\boldsymbol{x})$ is a neighborhood of $\boldsymbol{x}$, ρ is the density, $\boldsymbol{b}$ is the body force density field, and $\boldsymbol{f}$ is a material-dependent constitutive law that represents the force density that a point $\boldsymbol{y}$ inside the neighborhood exerts on $\boldsymbol{x}$ as a result of the deformation field. The radius ϵ of the neighborhood is referred to as the *horizon*. We assume $\mathcal{H}_\epsilon(\boldsymbol{x})$ is a ball of radius ϵ centered at $\boldsymbol{x}$. Here, all points satisfy the same field equations (3). Assuming the rate of loading is slow enough that the effect of inertia term is negligible, we arrive at the quasi-static formulation which states that the body is in equilibrium at all times, hence $\boldsymbol{u}(\boldsymbol{x}, t)$ is the solution of

$$\int_{\mathcal{H}_\epsilon(\boldsymbol{x})} \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x})\, d\boldsymbol{y} + \boldsymbol{b}(\boldsymbol{x}, t) = \mathbf{0}. \tag{4}$$

Here the parameter t can now be interpreted as a load parameter.

In this work, we focus on a numerical approximation of (4) when the force density $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x})$ is given by the regularized nonlinear pairwise (RNP) model introduced in [42] and further studied in [43, 37, 38, 48, 45]. This is a two point

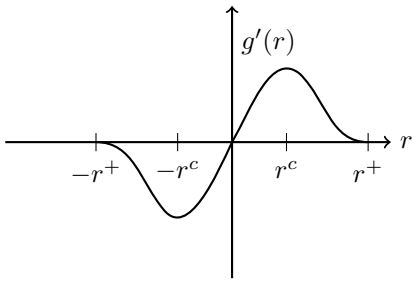


Figure 2: Cohesive force. The force goes smoothly to zero at $\pm r^+$.

density and the force between the two points, $\boldsymbol{f}(\boldsymbol{y},\boldsymbol{x})$ is referred to as a bond force between $\boldsymbol{y}$ and $\boldsymbol{x}$. For small strains the nonlocal force is linearly elastic but for larger strains the force begins to soften and then approaches zero after reaching a critical strain, see Figure 2. The associated nonlocal dynamics is called *cohesive dynamics*. This will be shown to be an ordinary state based model. This type of modeling can be extended to more general state based peridynamics models, see[47].

The tensile strain S between two points $\boldsymbol{x},\boldsymbol{y}$ in the domain D along the direction $\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}$ is defined as

$$S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t)) = \frac{\boldsymbol{u}(\boldsymbol{y},t)-\boldsymbol{u}(\boldsymbol{x},t)}{|\boldsymbol{y}-\boldsymbol{x}|}\cdot\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}, \tag{5}$$

where $\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}} = \frac{\boldsymbol{y}-\boldsymbol{x}}{|\boldsymbol{y}-\boldsymbol{x}|}$ is a unit vector and "$\cdot$" is the dot product. The critical strain $S_c > 0$ for which the force begins to soften is given by

$$S_c = \frac{r^c}{\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}}, \tag{6}$$

and the strength domain is described by the strain for which the force does not soften and is given by

$$-S_c < S < S_c. \tag{7}$$

The strain at which the force goes to zero is S_+

$$S_+ = \frac{r^+}{\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}}. \tag{8}$$

The nonlocal force in the RNP model can be defined in terms of a double well potential. The bond potential is a function of the strain and is defined for all $\boldsymbol{x},\boldsymbol{y}$ in D by

$$\mathcal{W}^\epsilon(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t))) = J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)\frac{1}{\epsilon^{d+1}\omega_d|\boldsymbol{y}-\boldsymbol{x}|}g(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t))) \tag{9}$$

where $\mathcal{W}^\epsilon(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t)))$ is the pairwise force potential per unit length between two points $\boldsymbol{x}$ and $\boldsymbol{y}$. It is described in terms of its potential function g, given by

$$g(r) = h(r^2) \tag{10}$$

where h is concave. Here ω_d is the volume of the unit sphere in dimension d, and $\epsilon^d\omega_d$ is the volume of the horizon $\mathcal{H}_\epsilon(\boldsymbol{x})$. The influence function $J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)$ is a measure of the influence that the point $\boldsymbol{y}$ has on $\boldsymbol{x}$. Only points inside the horizon can influence $\boldsymbol{x}$ so $J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)$ is nonzero for $|\boldsymbol{y}-\boldsymbol{x}| < \epsilon$ and zero otherwise. We take J^ϵ to be of the form: $J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|) = J(\frac{|\boldsymbol{y}-\boldsymbol{x}|}{\epsilon})$ with $J(r) = 0$ for $r \geq 1$ and $0 \leq J(r) \leq M < \infty$ for $r < 1$. Here $M < \infty$ is an upper bound on J. One of the examples of g is:

$$g(r) = C(1-\exp[-\beta r]) \tag{11}$$

where C,β are material dependent parameters.

In the RNP model, the bond force $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})$ is the derivative of the bond potential given by

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = 2\partial_S\mathcal{W}^\epsilon(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t)))\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}, \tag{12}$$

where

$$\partial_S\mathcal{W}^\epsilon(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t))) = \frac{1}{\epsilon^{d+1}\omega_d}\frac{J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)}{|\boldsymbol{y}-\boldsymbol{x}|}\partial_S g(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t))). \tag{13}$$

The potential energy of the motion is given by

$$PD^{\epsilon}(\boldsymbol{u}) = \int_D \int_{\mathcal{H}_\epsilon(\boldsymbol{x})\cap D} |\boldsymbol{y}-\boldsymbol{x}| \mathcal{W}^{\epsilon}(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t)))\, d\boldsymbol{y} d\boldsymbol{x}, \tag{14}$$

and define the energy

$$\mathrm{E}(\boldsymbol{u},t) = -PD^{\epsilon}(\boldsymbol{u}) + \int_D \boldsymbol{b}\cdot\boldsymbol{u}\, d\boldsymbol{x}.$$

On taking the first variation of the energy we see that the stationary point is the solution of the Euler-Lagrange equation (4) given by the peridynamic force

$$\mathcal{L}(\boldsymbol{u})(\boldsymbol{x}) = \int_{\mathcal{H}_\epsilon(\boldsymbol{x})} \boldsymbol{f}(\boldsymbol{y},\boldsymbol{x})\, d\boldsymbol{y} = \int_{\mathcal{H}_\epsilon(\boldsymbol{x})} \left(\mathbf{T}(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}) - \mathbf{T}(\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y})\right)\, d\boldsymbol{y}. \tag{15}$$

From (12) one sees that this is a bond-based or ordinary state based material model[66] with

$$\mathbf{T}(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}) = \partial_S \mathcal{W}^{\epsilon}(S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}(t)))\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}, \qquad \mathbf{T}(\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y}) = \partial_S \mathcal{W}^{\epsilon}(S(\boldsymbol{x},\boldsymbol{y},\boldsymbol{u}(t)))\boldsymbol{e}_{\boldsymbol{x}-\boldsymbol{y}}. \tag{16}$$

We assume at $t=0$ that there exists a displacement $\boldsymbol{u}_0 = \boldsymbol{u}(\boldsymbol{x},0)$ and body force $\boldsymbol{b}(\boldsymbol{x},0)$ for which the displacement is at equilibrium

$$\mathcal{L}(\boldsymbol{u}_0)(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x},0) = \boldsymbol{0}, \tag{17}$$

and the quasistatic evolution $\boldsymbol{u}(\boldsymbol{x},t) : D\times[0,T] \to \mathbb{R}^2$ is given by

$$\mathcal{L}(\boldsymbol{u})(\boldsymbol{x},t) + \boldsymbol{b}(\boldsymbol{x},t) = \boldsymbol{0}. \tag{18}$$

for a prescribed load path $\boldsymbol{b}(\boldsymbol{x},t) : D\times[0,T] \to \mathbb{R}^2$. Here $\boldsymbol{b}(\boldsymbol{x},t)$ includes boundary traction

On the other hand, displacement loading occurs when the displacement is prescribed on the boundary of the body in the absence of body force. For nonlocal problems, the displacement loading is prescribed along a layer of finite thickness ϵ, surrounding the body. Denote the extension of the peridynamic domain D by $\overline{D}$. It is defined as

$$\overline{D} = \left\{\boldsymbol{x}\in\mathbb{R}^2 : |\boldsymbol{y}-\boldsymbol{x}| \le \epsilon \text{ for } \boldsymbol{y}\in D\right\}. \tag{19}$$

The peridynamic boundary D^{ϵ} of D is defined as

$$D^{\epsilon} = \overline{D}\setminus D. \tag{20}$$

An evolution $\boldsymbol{u}(\boldsymbol{x},t) : \overline{D}\times[0,T] \to \mathbb{R}^2$ for a prescribed boundary displacement $\boldsymbol{U}(\boldsymbol{x},t) : D^{\epsilon}\times[0,T] \to \mathbb{R}^2$, $0\le t<T$ satisfies

$$\begin{cases} \mathcal{L}(\boldsymbol{u})(\boldsymbol{x},t) = 0 & \text{on } D \\ \boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{U}(\boldsymbol{x},t) & \text{on } D^{\epsilon} \end{cases} \tag{21}$$

with $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{U}(\boldsymbol{x},0)$. The Dirichlet boundary conditions applied in the simulations are described in section 6.

In what follows, we propose and implement a new numerical approach to the quasistatic fracture evolution problem and choose a discrete load parameter with increment $\Delta > 0$. The load parameter is defined by $t = t_n = n\times\Delta$ with $0\le n\le N$. At each t_n the displacement $\boldsymbol{u}(\boldsymbol{x},t_n)$ is the solution of

$$\mathcal{L}(\boldsymbol{u})(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x},t_n) = \boldsymbol{0}. \tag{22}$$

3.1 Failure Zone - Process Zone

Both pure elastic response and quasistatic fracture emerges from the model. This model can be thought of as a mesoscopic model, where the model treats the fracture as a failure zone and one recovers a fracture surface by passing to the limit of vanishing horizon $\epsilon\to 0$. This has been shown for the dynamic case, where the failure zone converges to a surface as $\epsilon\to 0$ and the RNP model recovers the energy of linear elastic fracture mechanics as the peridynamic horizon goes to zero, see[42, 43]. The Kinetic relation for the RNP model converges to that of LEFM[39] and for prescribed crack paths the displacement field converges to the solution of the wave equation on time-dependent domains[44].

The failure zone represents the crack in the RNP model. It is characterized by the failure zone center-line. In an illustrative example, the failure zone center-line starts at a precrack on the left side of the specimen and propagates into the interior. The force between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ separated by the failure zone center-line is zero. The center-line is shown in figure 3 and the failure zone is the gray region in figure 3. For the boundary conditions chosen here, the failure is in tension and confined to a neighborhood of the $x_2 = 0$ axis of width 2ϵ. Just in front of the failure zone is the process zone, where the force between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ on either side of the $x_2 = 0$ axis is decreasing with

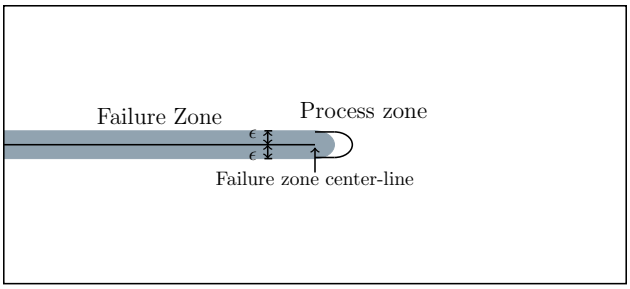


Figure 3: Depiction of a crack in this model: a failure zone is the gray shaded region and the process zone is the clear region inside the contour.

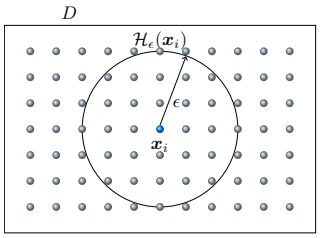


Figure 4: The material points at discrete positions $\boldsymbol{x}_i$ in the domain D in the reference configuration at time $t = 0$. For each discrete material point $\boldsymbol{x}_i$ the neighborhood $\mathcal{H}_\epsilon(\boldsymbol{x}_i) := \{j|\ |\boldsymbol{x}_j - \boldsymbol{x}_i| < \epsilon\}$ is computed. As an example, the neighborhood for the discrete node $\boldsymbol{x}_i$ is shown. The material point $\boldsymbol{x}_i$ exchanges force with all other discrete nodes within its neighborhood. This figure was adapted from the reference[14].

increasing strain. At the leading edge of the crack, one sees the force softening between points $\boldsymbol{x}$ and $\boldsymbol{y}$ and as the crack center-line moves forward passing between $\boldsymbol{x}$ and $\boldsymbol{y}$ the force between $\boldsymbol{x}$ and $\boldsymbol{y}$ decreases to zero, see figure 3. It needs to be stressed the failure zone and process zone emerge from the nonlocal dynamics and are not prescribed. The strain ratio with respect to critical strain at each node $\boldsymbol{x}$ is given by

$$d(\boldsymbol{x}) = \max\left(\frac{S(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{u}(t))}{r^c/\sqrt{|\boldsymbol{y} - \boldsymbol{x}|}}\right),\ \forall \boldsymbol{y} \in \mathcal{H}_\epsilon(\boldsymbol{x}). \tag{23}$$

When $d(\boldsymbol{x}) < 1$ the strain (stretched bonds) lie in the strength domain given by (7) and the material is behaving elastically, for $d(\boldsymbol{x}) = 1$ the maximal sustainable force corresponds to the strength of the material, and for $d(\boldsymbol{x}) > 1$ the collection of bonds have started to soften (i.e, the force between two points decreases) and have begun the failure process.

3.2 Discretization

To discretize the peridynamic equations of motion, a finite difference approximation is utilized. Figure 4 shows the domain D in the initial configuration at time $t = 0$. A set of mesh nodes $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^N \subset D$ is placed in the domain D. In this work, we consider an equidistant distance h between all mesh nodes in all directions. To each of the nodes, surrounding volumes $\{V_i \in \mathbb{R}\}_{i=1}^N$ is associated. These volumes are nonoverlapping $V_i \cap V_j = \emptyset$ and recover $\sum_{i=1}^N V_i \approx |D|$ the volume of the domain D. Each node $\boldsymbol{x}_i$ interacts with all neighbors within the finite neighborhood $\mathcal{H}_\epsilon(\boldsymbol{x}_i) = \{\boldsymbol{x}_j\,|\,|\boldsymbol{x}_j - \boldsymbol{x}_i| \leq \epsilon\}$. A common choice for the horizon is $\epsilon = mh$, where m is a constant $m > 1$ and h is the spacing between nodes.

4 Derivation of the explicit stiffness matrix

We seek a displacement field $\boldsymbol{u}$ that satisfies the quasistatic formulation, see Equation (18), which reads as

$$\mathcal{L}(\boldsymbol{u})(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x}) = 0.$$

The peridynamic force $\mathcal{L}$, see expression Equation 15, depends nonlinearly on the displacement $\boldsymbol{u}$. To make the above problem tractable and implementable in a computer, we apply the Newton's iterative method which is outlined in

Algorithm 1 Solver using a Newton method for the force control case, where the load is applied via the external force density $\boldsymbol{b}$.

Start with the initial guess of displacement $\boldsymbol{u}_0$
for $0 \leq k \leq N$ **do**
  Take $\boldsymbol{u}^k = \boldsymbol{u}^{k-1} + \Delta\boldsymbol{u}$
  Linearize $\mathcal{L}$ such that $\mathcal{L}(\boldsymbol{u}^k) = \mathcal{L}(\boldsymbol{u}^{k-1}) + H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}]$
  Solve for increment $\Delta\boldsymbol{u}$ using

$$H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}] + \mathcal{L}(\boldsymbol{u}^{k-1}) + \boldsymbol{b}(\boldsymbol{x}) = 0. \tag{24}$$

  if $\|\boldsymbol{u}^k - \boldsymbol{u}^{k-1}\| < \delta$ **then**
    Exit loop
  end if
end for

Algorithm 2 Solver using a Newton method for the displacement control case, where the load is applied via a prescribed displacement on an extension of the domain D

Extend the domain and applied the prescribed displacement $\mathbf{w}$ in the extension, see Figure 5a.
Start with the initial guess of displacement $\boldsymbol{u}_0 = 0$
for $0 \leq k \leq N$ **do**
  Take $\boldsymbol{u}^k = \boldsymbol{u}^{k-1} + \Delta\boldsymbol{u}$
  Linearize $\mathcal{L}$ such that $\mathcal{L}(\boldsymbol{u}^k) = \mathcal{L}(\boldsymbol{u}^{k-1}) + H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}]$
  Solve for increment $\Delta\boldsymbol{u}$ using

$$H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}] + \mathcal{L}(\boldsymbol{u}^{k-1} + \mathbf{w}) = 0. \tag{25}$$

  if $\|\mathbf{w}\| - \|\boldsymbol{u}^k\| < \delta$ **then**
    Exit loop
  end if
end for

Algorithm 1 for the force control case, where the load is applied via the external force density $\boldsymbol{b}$; and in Algorithm 2 for the displacement control case, where the load is applied via a prescribed displacement on an extension of the domain D.

We next derive the formula for $H(\boldsymbol{u})[\boldsymbol{w}]$. Substituting Equation 12 into Equation 15, and taking $\boldsymbol{u} = \boldsymbol{u}_0 + \Delta\boldsymbol{u}$, we have

$$\begin{aligned}&\mathcal{L}(\boldsymbol{u}_0 + \Delta\boldsymbol{u})(\boldsymbol{x})\\ &= \frac{2}{\epsilon^{d+1}\omega_d}\int_{D\cap\mathcal{H}_\epsilon(\boldsymbol{x})} \frac{J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)}{\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}} g'(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x},\boldsymbol{u}_0+\Delta\boldsymbol{u}))\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}d\boldsymbol{y}.\end{aligned} \tag{26}$$

From the definition of S in Equation 5, we have

$$S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0+\Delta\boldsymbol{u}) = S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0) + S(\boldsymbol{y},\boldsymbol{x};\Delta\boldsymbol{u}). \tag{27}$$

Let us do a Taylor expansion for the first derivative of the potential h

$$\begin{aligned}g'(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0+\Delta\boldsymbol{u})) &= g'(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0))\\ &\quad + g''(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0))\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\Delta\boldsymbol{u})\\ &\quad + O(||\Delta\boldsymbol{u}||^2).\end{aligned} \tag{28}$$

Substituting the above in Equation 26, we get

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}_0+\Delta\boldsymbol{u})(\boldsymbol{x}) &= \mathcal{L}(\boldsymbol{u}_0)(\boldsymbol{x})\\ &\quad + \frac{2}{\epsilon^{d+1}\omega_d}\int_{D\cap\mathcal{H}_\epsilon(\boldsymbol{x})} \frac{J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)}{\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}} g''(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0))\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\Delta\boldsymbol{u})\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}d\boldsymbol{y}\\ &\quad + O(||\Delta\boldsymbol{u}||^2).\end{aligned} \tag{29}$$

Denoting the first order term above as $H(\boldsymbol{u})[\boldsymbol{w}]$ as follows

$$H(\boldsymbol{u}_0)[\Delta\boldsymbol{u}](\boldsymbol{x}) = \frac{2}{\epsilon^{d+1}\omega_d}\int_{D\cap\mathcal{H}_\epsilon(\boldsymbol{x})} J^\epsilon(|\boldsymbol{y}-\boldsymbol{x}|)g''(\sqrt{|\boldsymbol{y}-\boldsymbol{x}|}S(\boldsymbol{y},\boldsymbol{x};\boldsymbol{u}_0))S(\boldsymbol{y},\boldsymbol{x};\Delta\boldsymbol{u})\boldsymbol{e}_{\boldsymbol{y}-\boldsymbol{x}}d\boldsymbol{y} \tag{30}$$

we can see that $H(\boldsymbol{u}_0)[\Delta\boldsymbol{u}]$ is linear in $\Delta\boldsymbol{u}$.

Discretization of quasistatic formulation and tangent matrix The discrete approximation of Equation 24 reads

$$H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}](\boldsymbol{x}_i) = -\boldsymbol{b}(\boldsymbol{x}_i) - \mathcal{L}(\boldsymbol{u}^{k-1})(\boldsymbol{x}_i), \tag{31}$$

where

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}^{k-1})(\boldsymbol{x}_i) = \frac{2}{\epsilon^{d+1}\omega_d}&\sum_{\substack{\boldsymbol{x}_j\in H_\epsilon(\boldsymbol{x}_i),\\ \boldsymbol{x}_j\neq\boldsymbol{x}_i}} \\ &\frac{J^\epsilon(|\boldsymbol{x}_j-\boldsymbol{x}_i|)}{\sqrt{|\boldsymbol{x}_j-\boldsymbol{x}_i|}}g'(\sqrt{|\boldsymbol{x}_j-\boldsymbol{x}_i|}S(\boldsymbol{x}_j,\boldsymbol{x}_i;\boldsymbol{u}^{k-1}))\boldsymbol{e}_{\boldsymbol{x}_j-\boldsymbol{x}_i}V_j \\ H(\boldsymbol{u}^{k-1})[\Delta u](\boldsymbol{x}_i) = \frac{2}{\epsilon^{d+1}\omega_d}&\sum_{\substack{\boldsymbol{x}_j\in H_\epsilon(\boldsymbol{x}_i),\\ \boldsymbol{x}_j\neq\boldsymbol{x}_i}} \\ &J^\epsilon(|\boldsymbol{x}_j-\boldsymbol{x}_i|)g''(\sqrt{|\boldsymbol{x}_j-\boldsymbol{x}_i|}S(\boldsymbol{x}_j,\boldsymbol{x}_i;\boldsymbol{u}^{k-1}))S(\boldsymbol{x}_j,\boldsymbol{x}_i;\Delta\boldsymbol{u})\boldsymbol{e}_{\boldsymbol{x}_j-\boldsymbol{x}_i}V_j.\end{aligned} \tag{32}$$

We substitute the definition of $S(\boldsymbol{x}_j,\boldsymbol{x}_i;\Delta\boldsymbol{u})$ in $H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}]$ to write

$$\begin{aligned}&H(\boldsymbol{u}^{k-1})[\Delta\boldsymbol{u}](\boldsymbol{x}_i) \\ &= \frac{2}{\epsilon^{d+1}\omega_d}\sum_{\substack{\boldsymbol{x}_j\in H_\epsilon(\boldsymbol{x}_i),\\ \boldsymbol{x}_j\neq\boldsymbol{x}_i}} \frac{J^\epsilon(|\boldsymbol{x}_j-\boldsymbol{x}_i|)}{|\boldsymbol{x}_j-\boldsymbol{x}_i|}g''(\sqrt{|\boldsymbol{x}_j-\boldsymbol{x}_i|}S(\boldsymbol{x}_j,\boldsymbol{x}_i;\boldsymbol{u}^{k-1}))\boldsymbol{E}_{\boldsymbol{x}_j-\boldsymbol{x}_i}(\Delta\boldsymbol{u}(\boldsymbol{x}_j)-\Delta\boldsymbol{u}(\boldsymbol{x}_i))V_j,\end{aligned} \tag{33}$$

where

$$\boldsymbol{E}_{\boldsymbol{x}_j-\boldsymbol{x}_i} = \boldsymbol{e}_{\boldsymbol{x}_j-\boldsymbol{x}_i}\otimes\boldsymbol{e}_{\boldsymbol{x}_j-\boldsymbol{x}_i}, \tag{34}$$

is the second-order tensor. The discrete problem Equation 31 can be represented as

$$\mathbb{K}(U^{k-1})\Delta U = F^{k-1}, \tag{35}$$

where $U^{k-1}, \Delta U, F^{k-1} = F(U^{k-1})$ is the discrete displacement vector at previous iteration $k-1$, the increment of the displacement in the current iteration, and force vector. Following Equation 31, we have

$$F_i^{k-1} = F(U^{k-1})_i = -\boldsymbol{b}(\boldsymbol{x}_i) - \mathcal{L}(U^{k-1})(\boldsymbol{x}_i). \tag{36}$$

It remains to show the form of the tangent matrix $\mathbb{K}$ which depends on the solution U^{k-1} from the previous iteration.

$$\mathbb{K}(\mathbf{u}) = \begin{bmatrix} \mathbb{K}_{1,1} & \mathbb{K}_{1,2} & \dots & \mathbb{K}_{1,N-1} & \mathbb{K}_{1,N} \\ \mathbb{K}_{2,1} & \mathbb{K}_{2,2} & \dots & \mathbb{K}_{2,N-1} & \mathbb{K}_{2,N} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbb{K}_{N-1,1} & \mathbb{K}_{N,2} & \dots & \mathbb{K}_{N-1,N-1} & \mathbb{K}_{N,N-1} \\ \mathbb{K}_{N,1} & \mathbb{K}_{N,2} & \dots & \mathbb{K}_{N-1,N} & \mathbb{K}_{N,N} \end{bmatrix}, \tag{37}$$

where each entry $\mathbb{K}_{ij}$ is a $d\times d$ matrix, where $d = 1,2,3$ is the dimension of the problem. Note, we suppress the notation of $\mathbb{K}$ on U^{k-1}. From Equation 33, we have

$$\mathbb{K}_{ij} = \begin{cases} \boldsymbol{A}_{ij}, & \text{if } i\neq j, \\ \sum_{\substack{\boldsymbol{x}_k\in\mathcal{H}_\epsilon(\boldsymbol{x}_k),\\ \boldsymbol{x}_j\neq\boldsymbol{x}_i}}\boldsymbol{A}_{ik}, & \text{if } i=j, \end{cases} \tag{38}$$

where second order tensor $\boldsymbol{A}_{ij}$ is given by

$$\boldsymbol{A}_{ij} \equiv \boldsymbol{A}_{ij}(\boldsymbol{u}^{k-1}) = \frac{2}{\epsilon^{d+1}\omega_d} \frac{J^\epsilon(|\boldsymbol{x}_j - \boldsymbol{x}_i|)}{|\boldsymbol{x}_j - \boldsymbol{x}_i|} g''(\sqrt{|\boldsymbol{x}_j - \boldsymbol{x}_i|} S(\boldsymbol{x}_j, \boldsymbol{x}_i; \boldsymbol{u}^{k-1})) \otimes \boldsymbol{E}_{\boldsymbol{x}_j - \boldsymbol{x}_i} V_j, \tag{39}$$

when $i \neq j$, and $\boldsymbol{A}_{ii} = \boldsymbol{0}$. Here by the notation $S(\boldsymbol{x}_j, \boldsymbol{x}_i; \boldsymbol{u}^{k-1})$ we mean

$$S(\boldsymbol{x}_j, \boldsymbol{x}_i; \boldsymbol{u}^{k-1}) = \frac{\boldsymbol{u}_j^{k-1} - \boldsymbol{u}_i^{k-1}}{|\boldsymbol{x}_j - \boldsymbol{x}_i|} \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{|\boldsymbol{x}_j - \boldsymbol{x}_i|}, \tag{40}$$

for discrete problems. Note, that the same approach can be used to derive the explicit stiffness for the state-based RNP model[46] with an additional linearization of the hydro-static force.

5 Algorithm

The algorithm for solving the system $\mathbb{K}(u)x = F$ is similar to the one described in[20, 49] using a Newton method, see Algorithm 3. Note that using a generic peridynamic material model, assembling the tangent stiffness matrix, see Line 5, can become computationally expensive since the entries are approximated using numerical schemes, *e.g.* central differences,

$$\mathbb{K}_{ij}(u) \approx \frac{\mathcal{L}^T(u + \epsilon)(x_i) - \mathcal{L}^T(u - \epsilon)(x_i)}{2\tau} \tag{41}$$

by perturbing the displacement. One is referred to[8] where a comparison of different methods are given. These include automatic differentiation, central difference, finite difference, and the complex step scheme[50, 51], for calculating tangent stiffness matrices in a massively parallel fashion. The relative accuracy of these schemes for computing the tangent stiffness matrix can be found in[8]. The perturbation approach requires the PD forces to be evaluated for a perturbed displacement $u \pm \tau$ over all directions, which is only computationally feasible in one dimension. For higher dimensions or numerous discrete nodes, the computational costs increase heavily, since the theoretical complexity of the algorithm is $\mathcal{O}(n^4)$. Meaning that for all n nodes the perturbed forces for all n neighbors have to be computed. Since the evaluation of the force costs n^2 operations, we have $n^2(d \cdot n^2 + d \cdot n^2)$ operations in total. On the other hand, for the RNP softening model the entries of the tangent stiffness matrix can be computed without perturbing the displacement of each node in all directions using Equation (38). Hence, the evaluation of the perturbed forces is not needed any more and this reduces the computational cost for the RNP model to $\mathcal{O}(n^3)$.

5.1 Force control

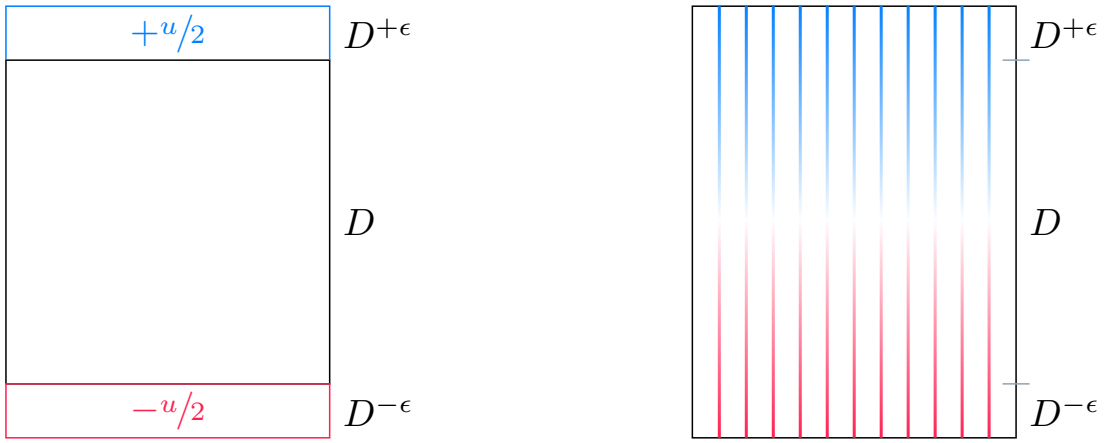
Algorithm 3 sketches the solver for the bond-based softening model for the force control case, where the load is applied via the external force density $\boldsymbol{b}$. The solver needs the external force density $\mathbf{b}$, the tolerance δ, and the perturbation τ. A guess for the initial displacement can be zero or some small perturbation in the size of τ. The initial guess of the displacement is used to compute the residual using Equation (36) and if the residual $r < \delta$ is smaller than the tolerance the initial guess is the solution. If not, iteration of Newton steps are done to get the solution of the displacement until the residual r is less than the tolerance δ. The tangent stiffness matrix is assembled using Equation (38). To avoid a singular matrix, all entries of the stiffness matrix and the force vector that are prescribed zero displacement due to the clamped boundary conditions are removed in Line 6 and Line 7. After that, the reduced system is solved. Note that this is just a single Newton iteration and for each successive load step, Algorithm 3 is repeated with $\boldsymbol{u}$ from the previous load step as the current initial guess $\boldsymbol{u}_0$ and a larger external load $\boldsymbol{u}$ as shown in Algorithm 1. These steps are not shown in the discrete algorithm for simplification.

5.2 Displacement control

Algorithm 4 outlines the solver used for the bond-based softening model for the displacement control case, where the load is applied via the displacement $\boldsymbol{u}$ instead of the external force $\boldsymbol{b}$ as in the force control. To apply the load in displacement, the domain D is extended with a layer of $\pm\epsilon$ in y-direction at the top $D^{+\epsilon}$ and bottom $D^{-\epsilon}$, respectively. Note that the extension of the domain to apply boundary conditions within peridynamics was applied in the following references[57, 65, 23, 1, 27, 53, 28]. Figure 5a shows the reference configuration of the domain D. A positive load displacement $+u/2$ is applied to the upper extension $D^{+\epsilon}$. A negative load displacement $-u/2$ is applied to the lower extension $D^{-\epsilon}$. The algorithm uses the tolerance δ and the perturbation τ as input. In Line 2 the reference configuration is extended by a layer of horizon size $\pm\epsilon$ and additional discrete PD nodes with the same nodal spacing are placed in the extension area. In Line 3 the initial displacement field $\boldsymbol{u}_0$ in the extended domain is guessed. A

Algorithm 3 Solver using a Newton method for the force control case, where the load is applied via the external force density $\boldsymbol{b}$.

1: Define the external force density b, tolerance δ, and perturbation τ
2: Guess the initial displacement $\mathbf{u}_0$
3: Compute the residual $r = \|\mathbf{F}\|$ //Equation (36)
4: **while** $r \geq \delta$ **do**
5: Assemble the tangent stiffness matrix $\mathbb{K}(\mathbf{u}) \in \mathbb{R}^{d \cdot N \times d \cdot N}$ //Equation (39)
6: Remove all columns/rows in $\mathbb{K}$ for nodes with prescribed zero displacement (clamped nodes)
7: Remove all entries in $\mathbf{F}$ for nodes with prescribed zero displacement (clamped nodes)
8: Solve the reduced system $\mathbb{K}\Delta\mathbf{u} = \mathbf{F}$ // Equation (24)
9: $\boldsymbol{u} += \Delta u$
10: $r = \|\mathbf{F}\|$
11: **end while**



(a) (b)

Figure 5: Extended domain for the displacement control: (a) reference configuration D with the extension $D^{+\epsilon}$ and $D^{-\epsilon}$ to apply the load in displacement $\mathbf{w}$. In this example, the assumption is made that a positive load in displacement is applied within the top extension and a negative load in displacement in the bottom extension. (b) sketch of the prescribed displacement $\mathbf{w}$ in y-direction and all values in the x-direction are zero. Here, the prescribed load in displacement is applied to the extension $D^{+\epsilon}$ and $D^{-\epsilon}$. To avoid high damage concentration at the interface of the extension and the domain, the prescribed displacement needs to be extended to the reference configuration D.

guess for the initial displacement can be zero or some small perturbation in the size of τ. In Line 4 the prescribed displacement is initialized. Here, the prescribed load in displacement is applied to the extension $D^{+\epsilon}$ and $D^{-\epsilon}$. For example, a constant displacement is applied to all discrete PD nodes in the extension. To avoid high damage concentration at the interface of the extension and the domain, the prescribed displacement needs to be extended to the reference configuration D. Figure 5b sketches an example field of the prescribed displacement $\mathbf{w}$ in y-direction and all values in the x-direction are zero. The displacement value is $+u/2$ and $-u/2$ in the extension and decays linearly to zero until reaching the middle of the plate. In this method, the displacement to be found $\boldsymbol{u}$ is determined by iteration so that the residual r is defined by

$$r = \left(\|\mathbf{w}^{D^{+\epsilon}}\| + \|\mathbf{w}^{D^{-\epsilon}}\|\right) - \left(\|\mathbf{u}^{D^{+\epsilon}}\| + \|\mathbf{u}^{D^{-\epsilon}}\|\right), \tag{42}$$

satisfies $r < \delta$. Note that the evaluation of the residual depends on the chosen prescribed displacement, $\mathbf{w}$ and in this case we ensure that the prescribed displacement in the extension matches the displacement solved in the Newton iteration of the current load step. Note that at this stage, the residual r is computed using the initial guess $\boldsymbol{u}_0$. If the initial guess of the displacement is not smaller than the tolerance δ a set of Newton iterations are started until the solution satisfies the tolerance, see Line 6. Inside the `while` loop all steps are identical to the ones for the force control (Algorithm 3), except that the force $\mathbf{F} = \mathcal{L}(\boldsymbol{u} + \mathbf{w})$ in Line 7 is evaluated differently. Note that all fields, *e.g.* force, displacements, and damage, need to be extended, thus the tangent stiffness matrix has a larger size of $m = \epsilon/h$.

Algorithm 4 Solver using a Newton method for the displacement control case, where the load is applied using a prescribed displacement $\boldsymbol{u}$ on an extension of the domain D of $\pm\epsilon$.

Define the tolerance δ, the load in displacement $\pm w$, and perturbation τ
Extend the domain D by a layer of horizon size $\pm\epsilon$ //see Figure 5a.
Guess the initial displacement $\mathbf{u}_0$
Initialize the prescribed displacement $\mathbf{w}$
Compute the residual $r = \left(\|\mathbf{w}^{D^{+\epsilon}}\| + \|\mathbf{w}^{D^{-\epsilon}}\|\right) - \left(\|\mathbf{u}_0^{D^{+\epsilon}}\| + \|\mathbf{u}_0^{D^{-\epsilon}}\|\right)$ //see Equation (42)
while $r \geq \delta$ **do**
    Compute $\mathbf{F} = \mathcal{L}(\boldsymbol{u} + \mathbf{w})$
    Assemble the tangent stiffness matrix $\mathbb{K}(\mathbf{u}) \in \mathbb{R}^{d\cdot(N+\epsilon/h)\times d\cdot(N+\epsilon/h)}$ //Equation (39)
    Remove all columns/rows in $\mathbb{K}$ for nodes with prescribed zero displacement (clamped nodes)
    Remove all entries in $\mathbf{F}$ for nodes with prescribed displacement (clamped nodes)
    Solve the reduced system $\mathbb{K}\Delta\mathbf{u} = \mathbf{F}$ // Equation (25)
    $\boldsymbol{u} += \Delta u$
    $r = \left(\|\mathbf{w}^{D^{+\epsilon}}\| + \|\mathbf{w}^{D^{-\epsilon}}\|\right) - \left(\|\mathbf{u}^{D^{+\epsilon}}\| + \|\mathbf{u}^{D^{-\epsilon}}\|\right)$ //see Equation (42)
end while

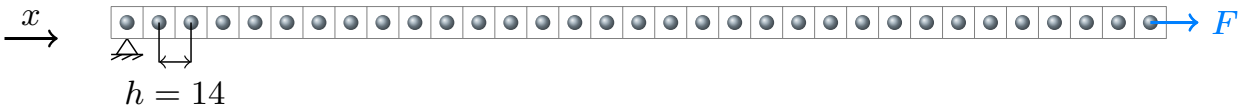


Figure 6: Sketch of the one dimensional bar benchmark test. The node on the left-hand side is clamped with respect to displacement ($u = 0$). A force F is applied on the node at the right-hand side. The figure was adapted from the reference[17].

6 Numerical examples

6.1 Linear elasticity

In this section, we validate our approach against classical continuum mechanics for the linear regime of the potential in Figure 2. Note that there is no damage involved since the strain between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ is less than critical, given by $r^c/\sqrt{|\boldsymbol{x}-\boldsymbol{y}|}$ and the material's strength is not exceeded, see Equation 7. These simulations are necessary to show that our approach recovers linear elasticity and gets results comparable to the approach using the numerical differentiation.

6.1.1 One-dimensional

For the one-dimensional case, the strain $\boldsymbol{\epsilon}_{\text{CCM}}$ from classical continuum mechanics (CCM) is recovered. The stress σ is defined as $\sigma = E \cdot \epsilon$ where E is the material's Young's modulus. The relation of Force F and stress reads as $\sigma = {}^{F}/_{A}$ with A as the area of the cross-section. Applying these two relations, the strain is obtained by $\boldsymbol{\epsilon} = {}^{\sigma}/_{\boldsymbol{E}} = {}^{\boldsymbol{F}}/_{(\boldsymbol{A}\cdot\boldsymbol{E})}$. Assuming a force F of 40N, a cross-section of 1m^2, and a Young's modulus E of 40GPa, the strain reads as $\boldsymbol{\epsilon_{CCM}} = 1 \times 10^{-8}$.

Figure 6 sketches the geometry for the one-dimensional model problem, which is used to recover the strain from classical continuum mechanics. The node on the left-hand side is clamped with respect to displacement. A load force is applied to the right-most node. The length of the bar is 16m. We chose $\epsilon = 3 \cdot h$ with $h = 1/4\text{m}$, a length $L = 16\text{m}$ and the tolerance was set to $\delta = 1 \times 10^{-11}$. To determine the bond-based material properties C and β in Equation (11), we used the relation $E = C \cdot \beta$. Since, we are interested in the linear elastic region of the potential, we can set $\beta = 1$ and thus $E = C = 40\text{GPa}$. For more details on the energy equivalence for the one-dimensional bond-based softening model, we refer to[37].

For the global strain, we get, $\boldsymbol{\epsilon} = 1.01 \times 10^{-8}$ which is close to the strain predicted by classical continuum mechanics $\boldsymbol{\epsilon_{CCM}} = 1 \times 10^{-8}$. As a second validation, the same discretized bar was simulated using Silling's state-based model[66] and the assembly of the tangent stiffness matrix using the numerical approximation of the derivative as in[49]. In that case, the predicted strain is $\boldsymbol{\epsilon} = 1.0 \times 10^{-8}$ using the author's C++ code[20, 36]. The python code

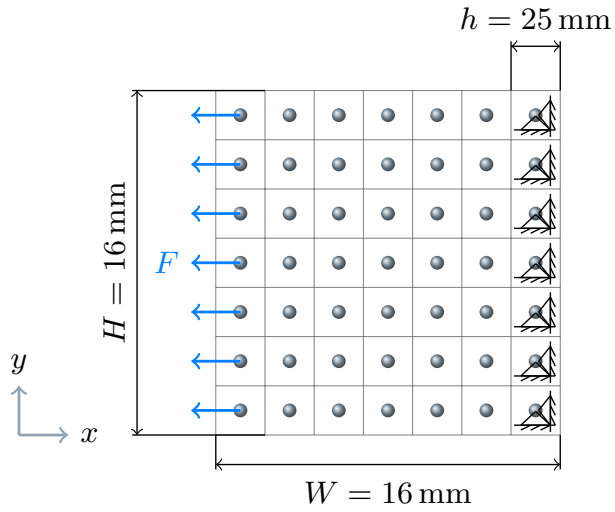


Figure 7: Sketch of the two-dimensional square plate for benchmarking. The displacement of the last layer of nodes is fixed in both directions. An external force in negative x-direction is applied to the first layer of nodes. This figure was adapted from the reference[16].

finished in 1.04s (24 iterations) using the presented approach, and the numerical approximation of the tangent stiffness matrix took 3.71s (31 iterations).

6.1.2 Two-dimensional

For the two-dimensional problem, the geometry in Figure 7, which is a square plate of $W = 16\,\mathrm{mm}$ by $L = 16\,\mathrm{mm}$, is used. The nodes on the right-hand side are clamped in displacement ($u_x = u_y = 0$). A body force $F = -40$N is applied on the first line of nodes on the left-hand side. The nodal spacing is chosen as $h = 0.1$ and the horizon is $\epsilon = 5h$. The material properties are chosen as Young's modulus $E = 4000$Pa, Poisson's ratio $\nu = 1/3$, and Energy release rate $G = 500\mathrm{J\,m^{-2}}$.

To validate our assembly of the stiffness matrix, we use the solution from classical continuum mechanics, see[20], derived by using the Airy stress function[62]. The following equations show the displacement obtained by classical continuum mechanics

$$\underline{u}_x(\boldsymbol{x}_i) = \frac{F}{EWT}\left(\mathbf{x}_{i_x} - W\right) \tag{43}$$
$$\underline{u}_y(\boldsymbol{x}_i) = -\frac{\nu F}{ET}\left(\frac{\mathbf{x}_{i_y}}{W} - \frac{1}{2}\right).$$

Figure 8 shows the solution of the displacement fields from Equation (44). The bond-based material parameters C and β in Equation (11) can be related to classical continuum mechanics as follows[12]

$$C = \frac{16}{6}\frac{K_{\mathrm{IC}}^2}{E}, \tag{44}$$
$$\beta = \frac{20\mu}{C}. \tag{45}$$

Figure 9a shows the displacement u_x obtained with our approach using a tolerance $\delta = 1 \times 10^{-6}$. The error with respect to CCM is overestimated by 13.7%. Figure 9b shows the displacement u_x obtained by assemble of the tangent stiffness matrix using the numerical approximation of derivative as in[49]. Note that the state-based model of Silling[66] instead of a bond-based model was used, however, the tolerance $\delta = 1 \times 10^{-6}$ was the same. Here, there is no error with respect to CCM. Note that one would have to do a convergence study for both quasi-static methods by varying the nodal spacing h and the horizon δ. However, this is out of the scope of the paper, since the focus of the paper is the explicit assembly of the stiffness matrix and validation of the approach to CCM and the approach to assemble the stiffness matrix using the numerical approximation of the derivative.

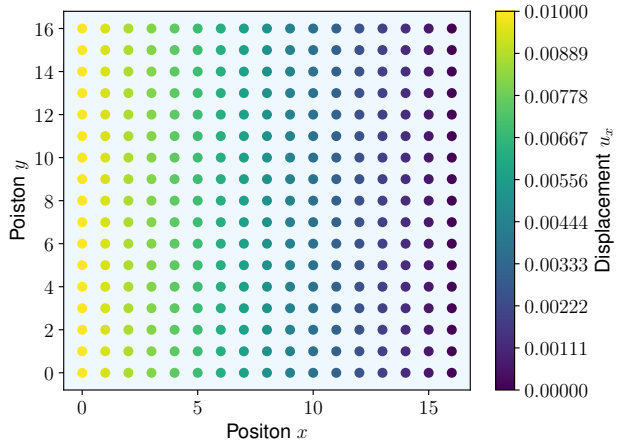
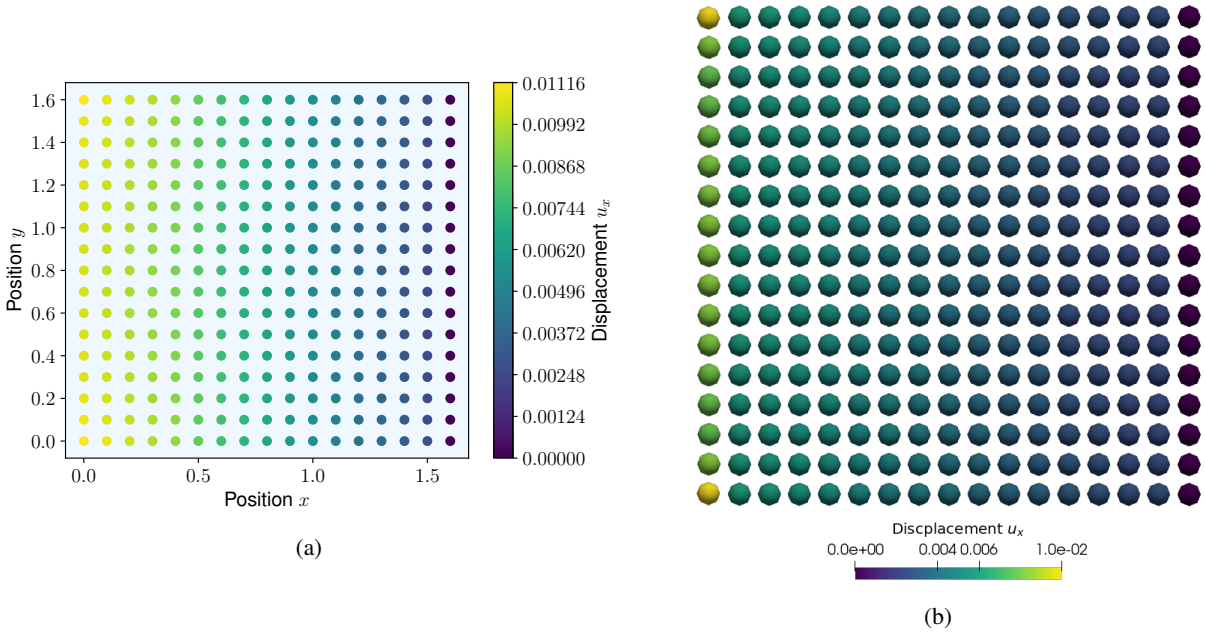


Figure 8: Displacement u_x (a) and u_y (b) at the nodes obtained by classical continuum mechanics from Equation (44).



(a)

(b)

Figure 9: Displacement u_x (a) obtained by our approach using a bond-based peridynamic model and u_x (b) obtained using the numerical derivation of the stiffness matrix and a state-based model using the author's C++ code. Adapted from[18].

The computation time for our approach employing the explicit stiffness matrix took 238 minutes using the prototype Python code with one CPU while for the computationally assembled stiffness matrix approach we used the author's C++ code, and it took 191 minutes using five CPUs. Thus, for this example, the explicit approach seems to be far more efficient than the standard numerical perturbation approach.

6.2 Damage

Previously, we validated our approach against classical continuum mechanics and now, we will show the performance of our method for fracture simulations. First, we investigate the influence of softening bonds on the condition number of the tangent stiffness matrix in Section 6.2.1. An ASTM E8 like tensile test is done to observe crack initiation in Section 6.2.2. Last, a notched square plate is used as an example for crack growth from a pre-crack in Section 6.2.3 for force control and in Section 6.2.4 for the displacement control, respectively.

6.2.1 User controlled damage

To study the influence of broken bonds on the matrix's condition number, the two-dimensional model problem in Section 6.1.2 is employed without the applied boundary conditions. A precrack from (0,0.8) to (0.8,0.8) is introduced. The precrack is modeled by excluding all peridynamic bonds that intersect this line from the neighborhood $\mathcal{H}_\epsilon(\boldsymbol{x})$. Figure 10a shows the grid with the initial crack, and all discrete PD nodes are colored with the number of neighbors. To analyze the influence of softening bonds on the condition number of the stiffness matrix K, bonds adjacent to the pre-crack are artificially softened to zero for three different scenarios: 1) the crack grows each time step by $1/2$ of the mesh size h, 2) grows at each step by exactly the mesh size, and 3) grows at each time step by $1.5\ h$. Figure 10 shows the artificial crack growth for the second case. Figure 11 shows the condition number $con(K)$ of the stiffness matrix K for all three cases. The condition number $con(K) = {}^{|K|_{L_2}}/_{|K^{-1}|_{L_2}}$ was obtained using NumPy, where the method described in[41] was implemented. For all three cases, we see that the condition number increases linearly per iteration. However, even for the fully cracked plate, the condition number increased from around 290 for the pre-cracked plate up to around 330.

6.2.2 Crack initiation for tensile testing with force control)

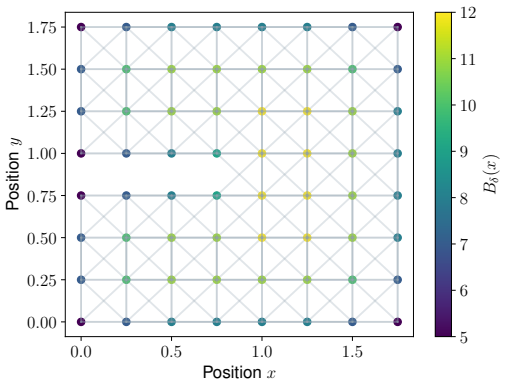
In the first two model problems, we focused on linear elasticity to showcase that our approach can reproduce the results where the cohesive force in Figure 2 stays in the linear regime. Note that these steps are necessary to validate this model, however, the interesting aspect of this approach is the part where the crack starts to initiate or grow. To showcase our approach, we now recover this important feature of peridynamics numerically. We use a flat test specimen for a tensile test according to standard *ASTM E8*[4]. Figure 12 shows the simplified geometry of the flat test specimen. Note that we rotated the specimen by, $90°$ and we used metric units. The nodes within the left-hand side are clamped in x-direction and y-direction. On the right-hand side a force of 4×10^6N is applied to the blue colored nodes. Note that we apply a force density in peridynamics by dividing the force by the area of the clamped region. The nodal spacing h is set to 1mm and the horizon ϵ is set to $4h$. The solver tolerance δ is set to 1×10^{-6}. For material properties, we chose $C = 3 \times 10^8$ and $\beta = 0.1$. Note that the same model was utilized in[12] for a tensile test with explicit time integration.

Figure 13 shows the evolution of the strain ratio in the tensile specimen. Figure 13a shows the strain ratio after one load step. Figure 13b shows that the strain ratio is over unity in a small region about the pre-crack and consequently highly localized damage after 50% of the load steps. Figure 13c shows the last load step before the explicit stiffness matrix becomes singular. We tried to continue beyond the last load step using smaller load steps, however, the matrix remained unstable. This behavior of unstable crack growth is depicted in the R curve in Figure 1. Once a load which has a (G, l) locus that intersects the R curve tangentially is reached, for example, P_3 in Figure 1, the crack growth becomes unstable and will continue to grow with no additional force. However, one can find some smaller loads in force, for example, P_1 and P_2, where the crack would grow a small length l as we increase the loading before becoming unstable. The simulation shows the crack initiation's location is near the re-entrant corner on the right side, where the force is applied. This is also the side (by design) where the crack initiates in the experiment. Figure 14 shows the displacement in y direction for the same load steps.

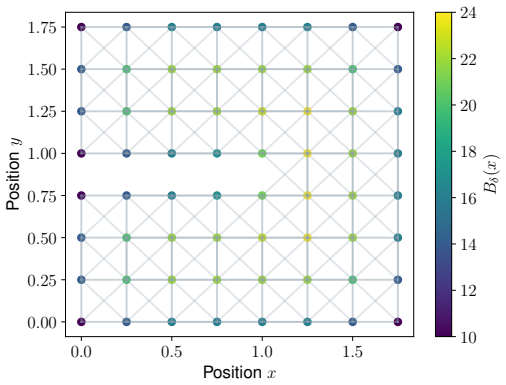
6.2.3 Pre-cracked plate with force control

Figure 15 shows a sketch of the pre-cracked square plate (15×15) with an initial crack of length 7.5. All nodes within the lower and upper right-hand side rectangle of horizon size ϵ are clamped in both directions. All nodes within the lower and upper left-hand side rectangle of horizon size $\epsilon \times 13\epsilon$ are loaded with the force of $\pm 4 \times 10^6$N in y-direction. The load in force refers to the so-called force control in the theory of fracture mechanics. The nodal spacing h was 0.2 and the horizon ϵ was $4 \cdot h = 0.8$. The tolerance was set to $\delta = 1 \times 10^{-5}$. All bonds between the PD nodes crossing the initial crack line were removed from the neighborhood $\mathcal{H}_\epsilon(\boldsymbol{x}_i)$. As material properties, we chose $C = 3 \times 10^8$ and $\beta = 0.1$.

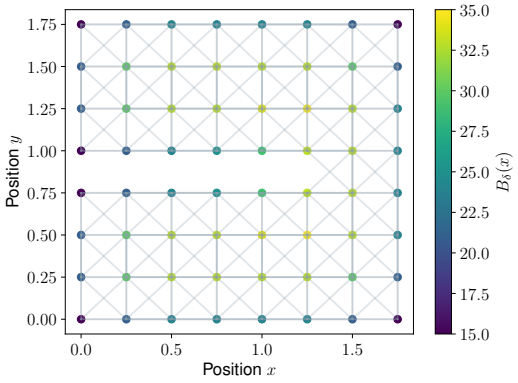
First, we look into the strain ratio while the bond stretch is far below one, see Figure 16. Note that the damage is plotted with respect to the reference configuration D. The white line indicates the initial crack, and all bonds intersecting this line are initially broken. The color map is chosen such that dark blue corresponds to zero strain ratio, light red indicates regions where the strain ratio is sufficient for some bonds within the horizon to fail, and dark red indicates that all bonds within the horizon are fully softened to zero. The strain ratio after one load step with $F_1 = \pm 4 \times 10^6$N is at 0.11 and no damage occurs. The maximum strain ratio after six load steps (b) with $F_6 = \pm 4 \times 10^6$N is 0.72 and the strains are still within the strength domain of the material. The highest strain ratios are localized around the tip of the initial crack.



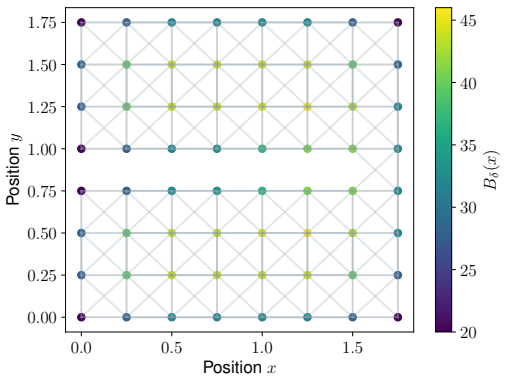
(a) Initial configuration



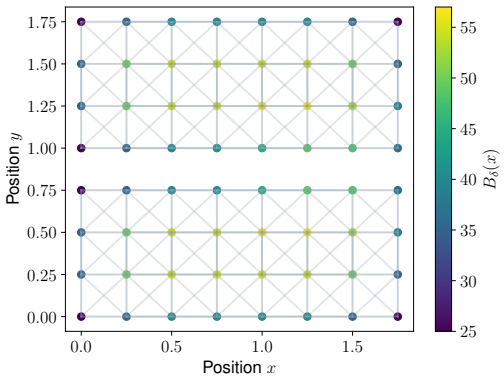
(b) Crack tip = h



(c) Crack tip = $2h$



(d) Crack tip = $3h$



(e) Crack tip = $4h$

Figure 10: The initial configuration of the plate with the initial crack (a). The consecutive images show the artificial growth of the crack, where the crack growth was controlled by softening all bonds along the initial crack. In the first iteration (b) the crack tip grows for one mesh width h. In the second iteration (c) for $2h$. In the third iteration (d) for $3h$. And in the last iteration, the plate is completely broken (e). The resulting condition number $con(K)$ for each iteration is shown in Figure 11.

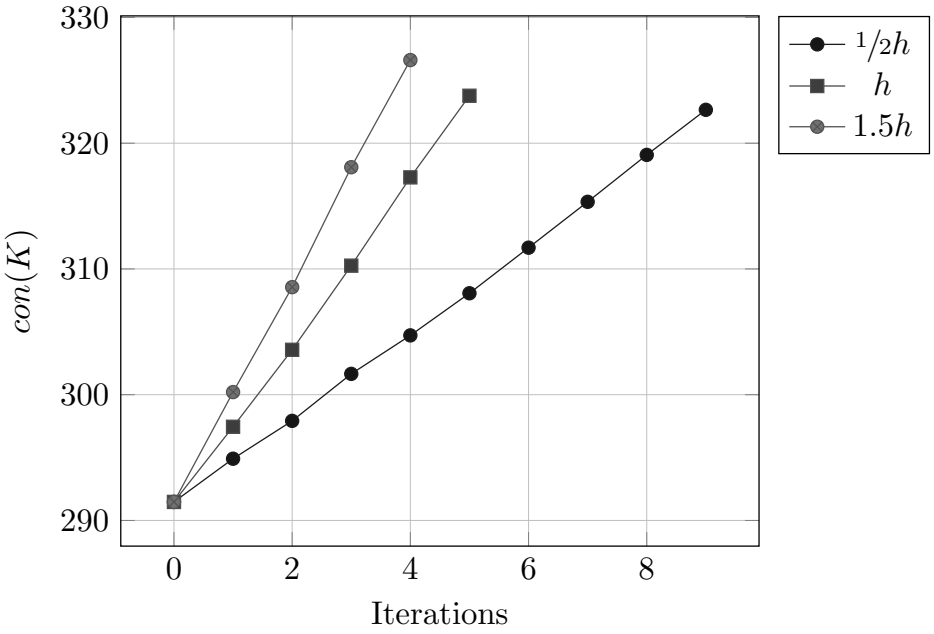


Figure 11: The condition number $con(K)$ of the tangent stiffness matrix K with respect to the crack growth. The initial configuration of the half-cracked square plate is shown in Figure 10a. At iteration zero, the condition number was computed for this configuration. To analyze the influence of soften bonds to the condition number of the stiffness matrix K, bonds along the initial crack are artificially softening to zero for three cases: 1) the crack grows $^1/_2$ of the mesh size h, 2) grows the mesh size, and 3) grows $1.5h$ per iteration.

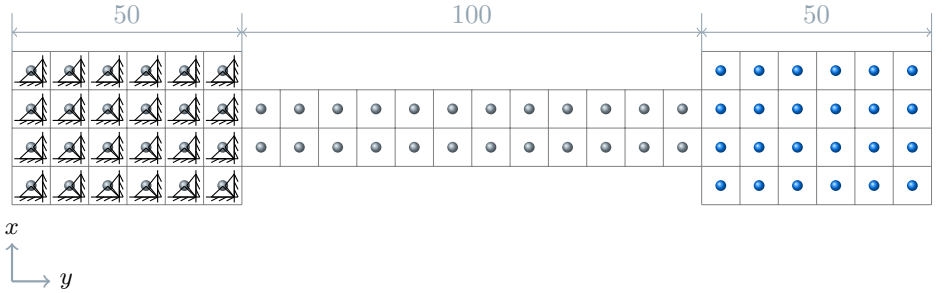


Figure 12: Sketch of the ASTM ES like tensile specimen. Note that we rotated the specimen by $90°$ and metric units were used. The nodes within the left-hand side are clamped in the x-direction and y-direction. On the blue colored nodes, a force of 4×10^6N is applied. Note that we apply a force density in peridynamic by dividing the force by the area of the clamped region. The figure was adapted from the reference[13].

Second, we look at the strain ratio while the bonds started to soften, but damage has not started yet, see Figure 17. The strain ratio after nine load steps (Figure 17a) with $F_9 = 36 \times 10^6$N shows that first bond stretches are at 140% and exceeding the strength domain and bonds are softening. Here, the bond softening is still localized in a layer of horizon size δ at the tip of the initial crack. The strain ratio after eleven load steps (Figure 17b) with $F_{11} = 44 \times 10^6$N shows, that the bonds are still softening, but damage has not yet begun.

Third, the bonds are softening damage growth has begun, the damaged zone is the location where the bonds are fully softened. The damage after twelve load steps with $F_{12} = 44.4$N is shown in (a) and after 21 load steps with $F_{21} = 44.436$N in (b), respectively. We had to reduce the load in force to $\pm 4 \times 10^5$N at load step 12 and further to $\pm 4 \times 10^3$N for the remaining load steps, since the large force resulted in an unstable matrix. At some point, the matrix remained unstable while decreasing the load further, which conforms with the theory that force control is more likely to result in unstable crack growth. Figure 19 shows the corresponding displacement field in y-directions at the discrete PD nodes (black = $min(u_y)$ and gray = $max(u_y)$).

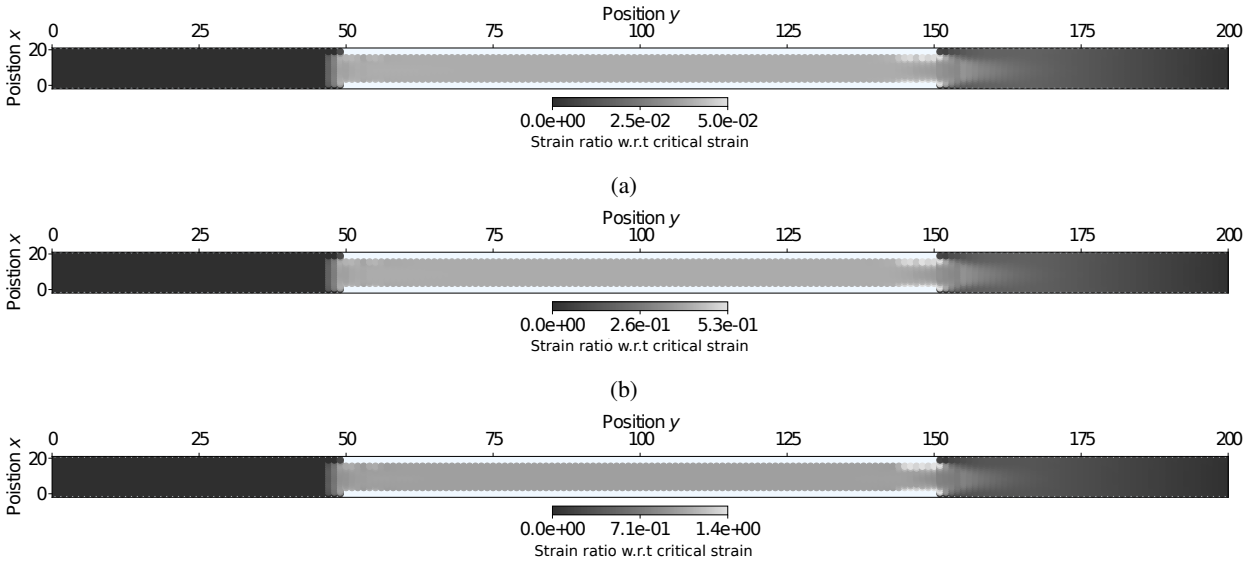


Figure 13: Strain ratio $d(\boldsymbol{x})$ w.r.t to critical strain for force control at the discrete PD nodes (black = undamaged and gray = damaged) for the following loading steps: (a) after the first load step, (b) after 10 load steps, and (c) after 20 load steps. Note that the damage is plotted with respect to the reference configuration D.

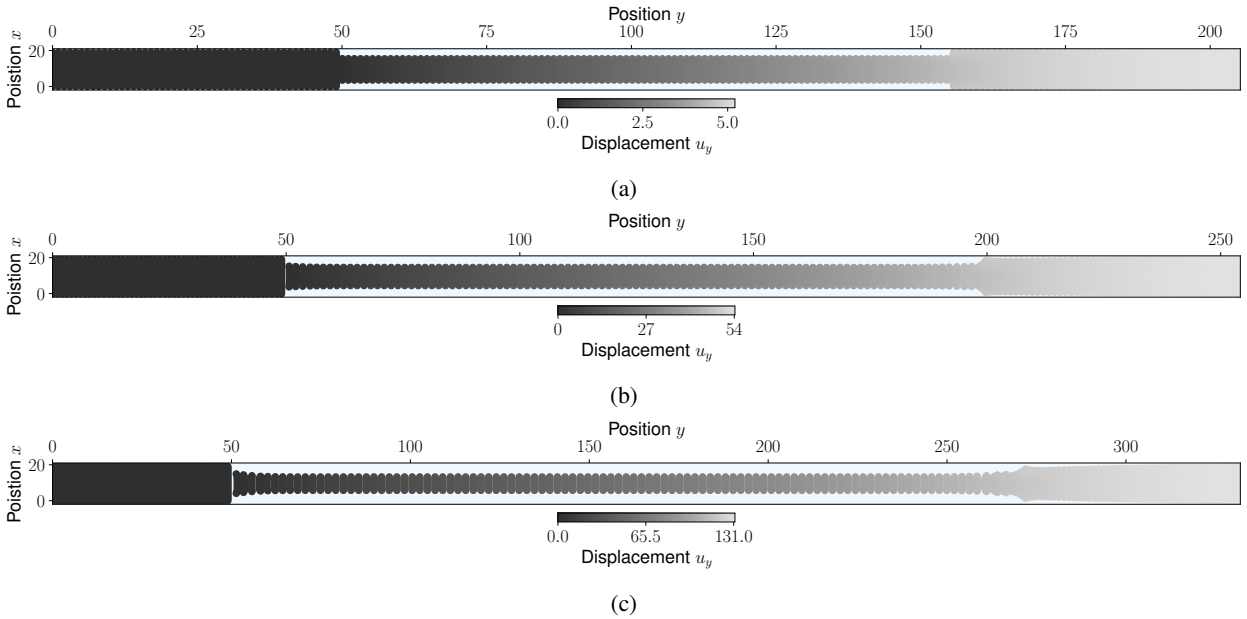


Figure 14: Displacement u_y for force control at the discrete PD nodes (black = 0 and gray = $max(u_y)$) for following loading steps: (a) after the first load step, (b) after 10 load steps, and (c) after 20 load steps. The nodes are plotted in the deformed configuration.

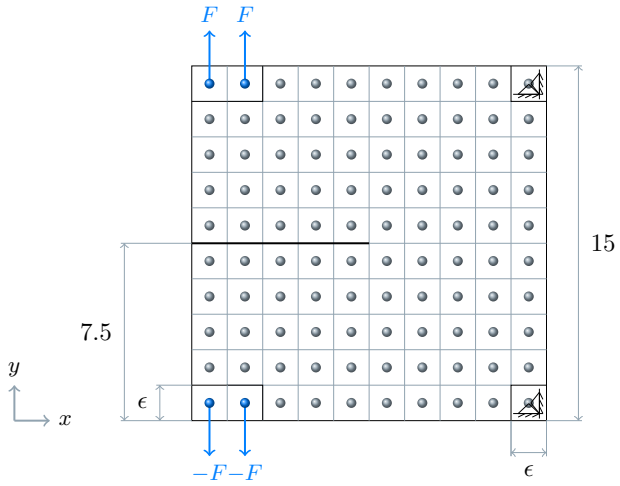
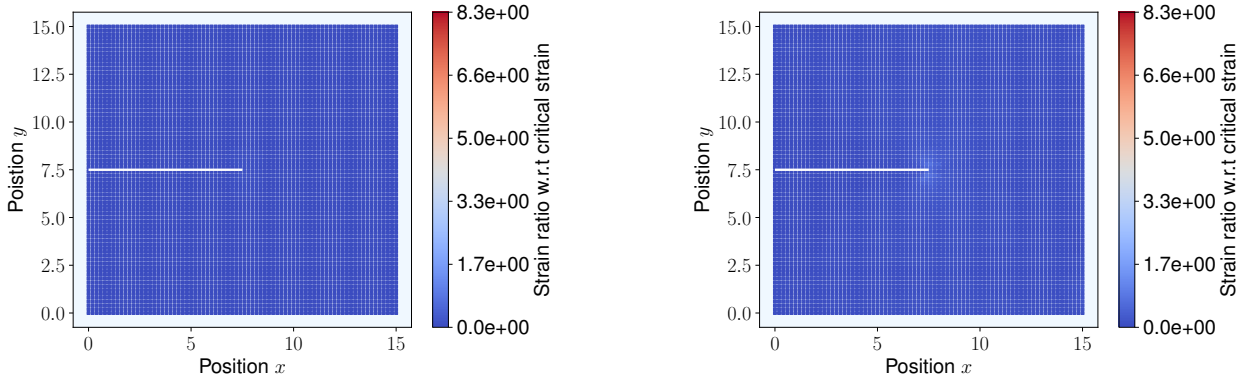


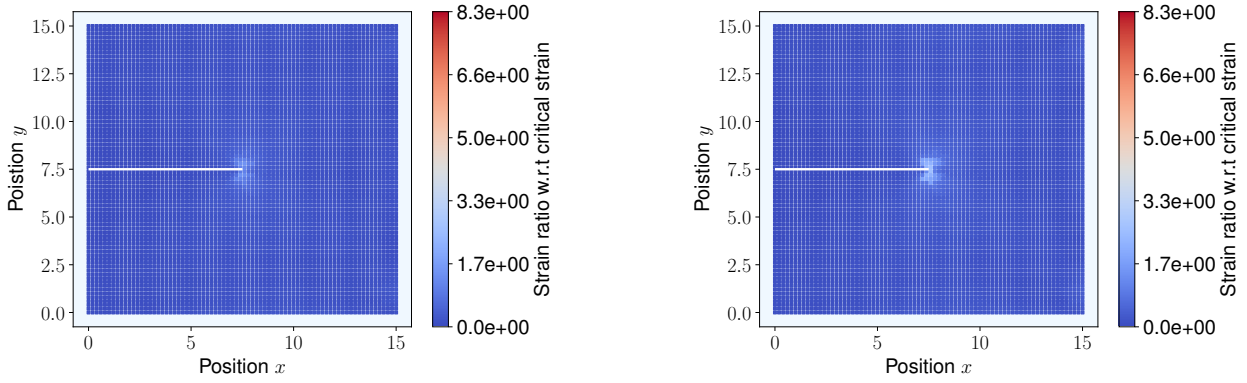
Figure 15: Sketch of the two-dimensional pre-cracked square plate with an initial crack from the mid of the left-hand side to the center of the plate. All nodes in a square of horizon size ϵ at the lower right and upper right corner are fixed in displacement in both directions. On the lower left and the upper left, an external force in y-direction is applied to all nodes within a rectangle of size $\epsilon \times 2\epsilon$. This figure was adapted from the reference[15].



(a) (b)

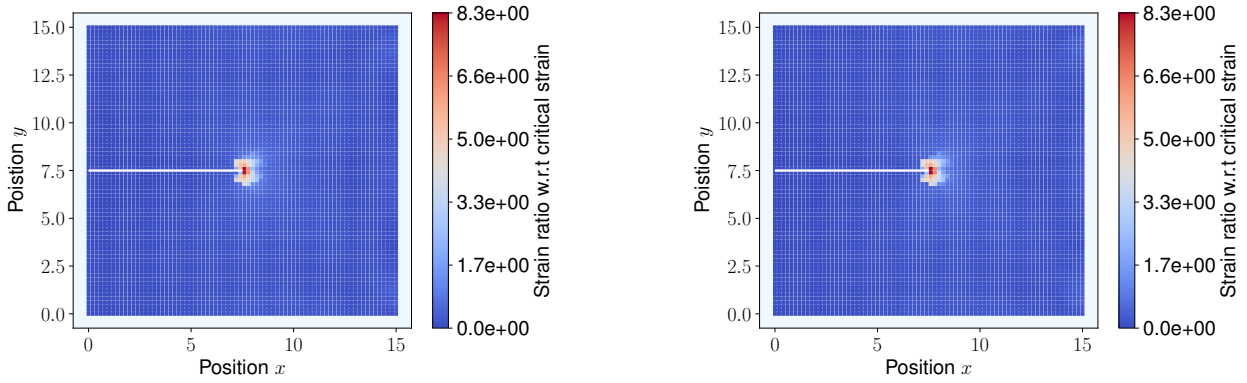
Figure 16: Plot of the strain ratio $d(\boldsymbol{x})$ w.r.t critical stretch for force control In this graphic the bond stretch is within the strength domain everywhere and the material behaves elastically across in the specimen. The color map is chosen such that dark blue corresponds to zero strain ratio, light red indicates regions where the strain ratio is sufficient for some bonds within the horizon to fail, and dark red indicates that all bonds within the horizon are fully softened to zero. The strain ratio after one load step with $F_1 = \pm 4 \times 10^6$N is at 0.11 and no damage occurs. The maximum strain ratio after six load steps (b) with $F_6 = \pm 4 \times 10^6$N is 0.72 and the strains are still within the strength domain of the material. The largest strains are localized around the tip of the initial crack. Note that the strain ratio is plotted with respect to the reference configuration D. The white line indicates the initial crack, and all bonds intersecting this line are initially broken.

We like to share three remarks here. First, from the numerical perspective this behavior is seen in other quasi-static PD methods that once the crack starts to grow and bonds soften or drop to zero, the load step needs to be reduced to keep the condition number $con(K)$ of the tangent stiffness matrix feasible and the matrix solvable, see Figure 11. Second, from the physical perspective we know that once the crack starts to grow, the force required to drive the crack reduces. Thus, with the same force, the crack grows faster. Third, applying the load in force, the so-called force control, results in the crack only growing for a few load steps before becoming unstable. The simulations show that this is to be expected since for the force control most of the deformation is borne by the pre-crack opening displacement with very small strain seen in the intact material except at the tip of the precrack where material damage and strain is concentrated. This behavior of unstable crack growth is reflected in the theory, as depicted in the R curve in Figure 1. Once a load that intersects the R curve tangent is reached, for example, P_3 in Figure 1, the crack growth will remain



(a) (b)

Figure 17: Plot of the strain ratio $d(\boldsymbol{x})$ w.r.t critical stretch for force control the strain continues to grow near the pre-crack tip. The color map is chosen such that dark blue indicates zero strain ratio, lighter blue higher strain ratio and light red indicates the strain ratio has exceeded unity consequently the damage process started, and dark red indicates that all bonds within a horizon are fully softened to zero. The strain ratio after nine load steps (a) with $F_9 = 36 \times 10^6$N shows that strain ratio has exceeded unity by, 140% so bonds are softening. Here, bond softening is still localized in a layer of horizon size ϵ at the tip of the initial crack. The strain ratio after eleven load steps (b) with $F_{11} = 44 \times 10^6$N shows, that bonds are still softening, but growth of the damage region (crack growth) has not yet begun. The strain ratio is plotted with respect to the reference configuration D. The white line indicates the initial crack, and all bonds intersecting this line are initially broken.



(a) (b)

Figure 18: Plot of the strain ratio $d(\boldsymbol{x})$ w.r.t critical stretch for force control The bonds are softening and damage growth has begun. The damaged zone is the location where bonds are fully softened. The damage after twelve load steps with $F_{12} = 44.4 \times 10^6$N is shown in (a) and after 21 load steps with $F_{21} = 44.436 \times 10^6$N in (b), respectively. We had to reduce the load in force since the large force resulted in an unstable matrix. At some point, the matrix remained unstable while decreasing the load further, which conforms with the theory that force control is more likely to result in unstable crack growth. Note that the damage is plotted with respect to the reference configuration D. The white line indicates the initial crack, and all bonds intersecting this line are initially broken.

unstable while increasing the load in force. However, one can find some smaller loads in force, for example, P_1 and P_2, where the crack would grow a small length l.

6.2.4 Pre-cracked plate with displacement control

For the displacement control, the geometry of the precracked square plate in Figure 15 was modified to apply the load in displacement to the top and bottom of the plate according to Algorithm 4. Figure 20a shows the modified geometry for the displacement control. To the blue nodes, a positive load in displacement u and to the red nodes, a negative load in displacement $-u$ is applied. The lower and upper node of the last line of nodes are clamped to ensure a solvable

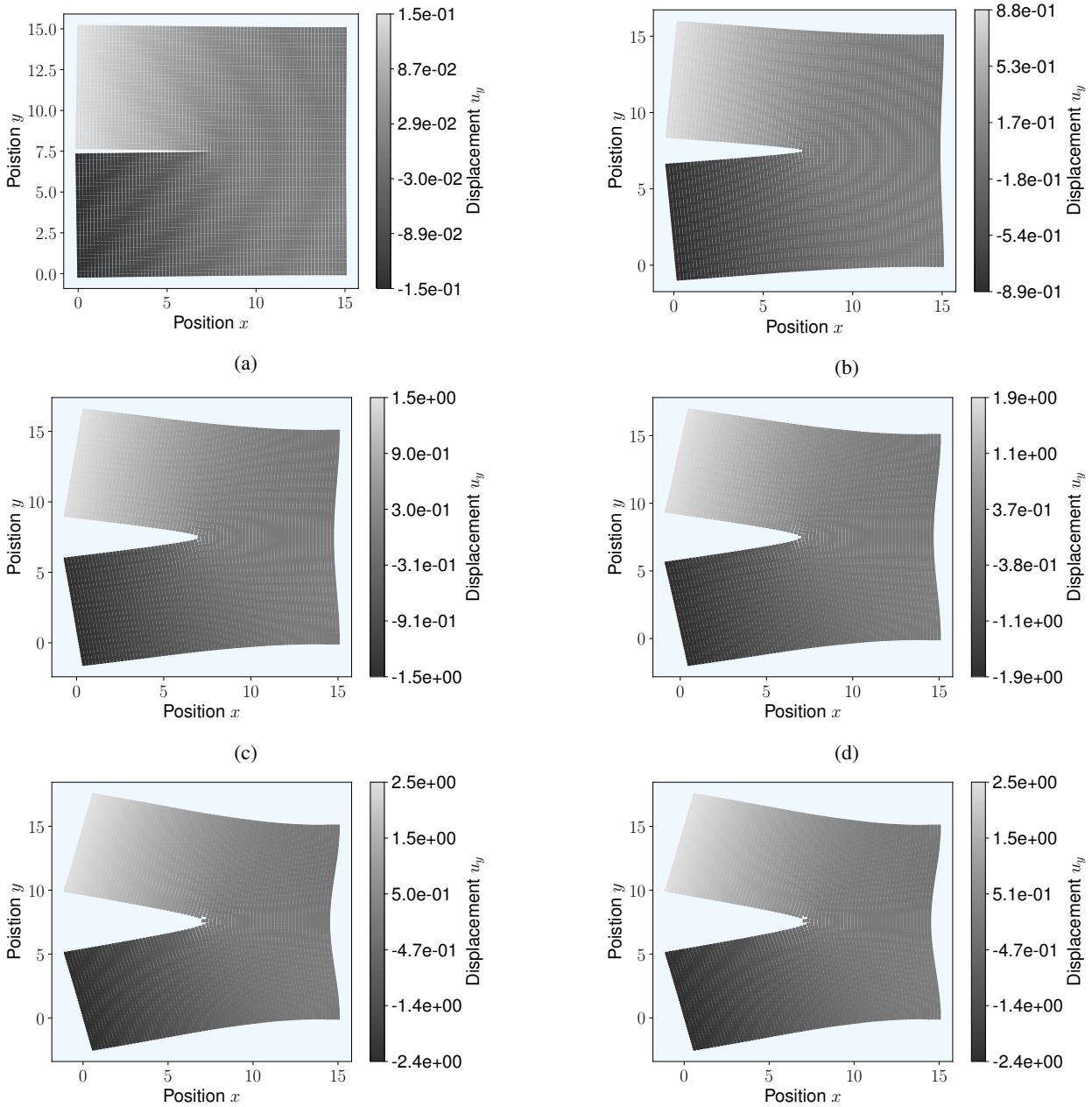
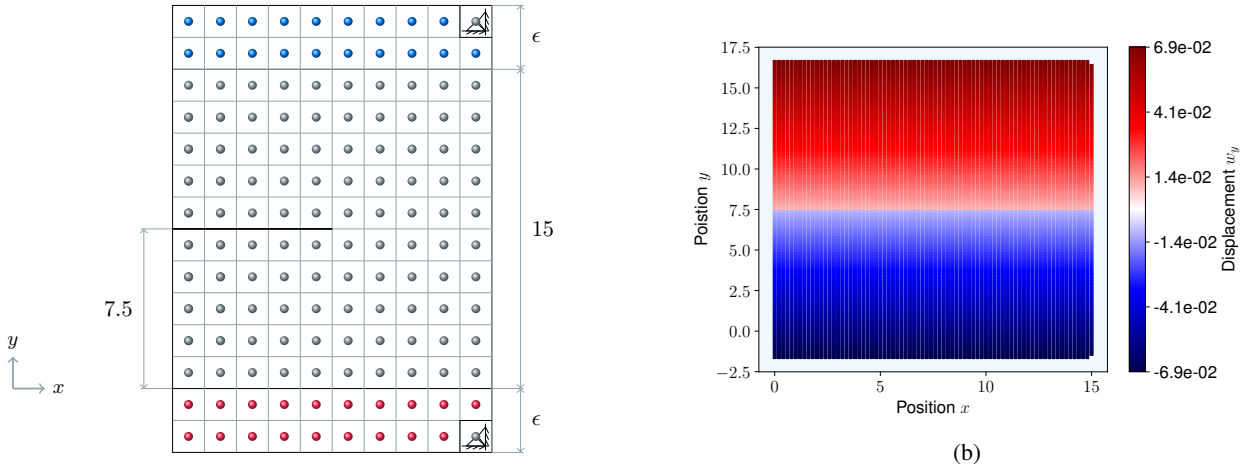
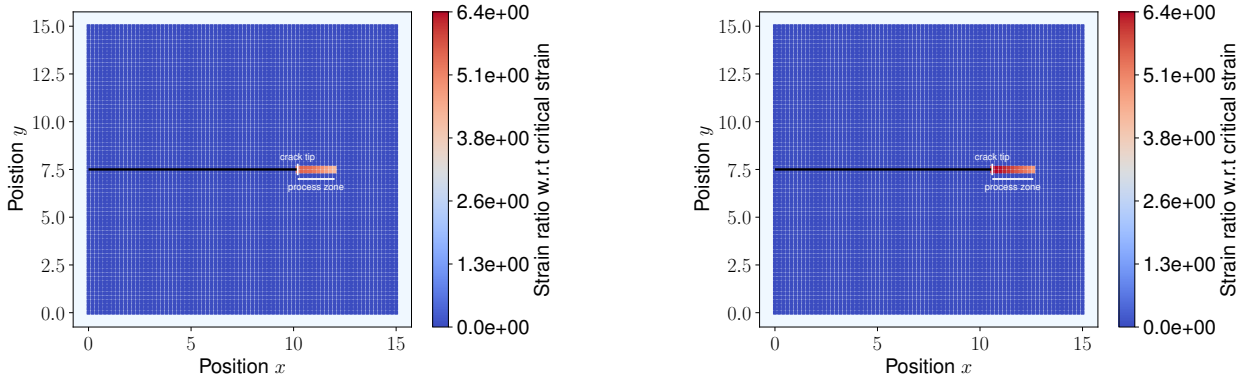


Figure 19: Displacement u_y for force control at the discrete PD nodes (black = $min(u_y)$ and gray = $max(u_y)$) for the following loading steps: An external force $F_1 = \pm 4 \times 10^6$N was applied for one load step (a), after six load steps (b), after nine load steps (c), and 11 load steps (d). An external force $F_2 = \pm 4 \times 10^5$N was applied for one load step (e) since the larger load resulted in an unstable matrix. After that, an external force $F_3 = \pm 4 \times 10^3$N was applied for nine time steps (f) before the matrix became unstable again.



(a) (b)

Figure 20: Extended domain for the displacement control: (a) of the extended domain where a load in positive displacement $\boldsymbol{u}$ is applied to the blue colored discrete nodes and a negative load $\boldsymbol{u}$ in displacement to the red colored discrete nodes. (b) shows the field of the prescribed displacement $\boldsymbol{w}_y$ in the y-direction. For more details, we refer to Algorithm 4.



(a) (b)

Figure 21: Plot of the strain ratio $d(\boldsymbol{x})$ for displacement control. The crack is indicated by the black crack zone center line and grows stably with increasing load. The white vertical line indicates the crack tip, which are the nodes with the maximal damage. The white horizontal line indicates the process zone where some bonds have softened to zero already, but not sufficiently many to create a surface increment over which a point in the upper half of the specimen is not influenced by forces from points in the lower half of the specimen. The damage after 11 load steps is shown in (a) and after 12 load steps in (b), respectively. Note that the damage is plotted with respect to the reference configuration D and the discrete nodes in the extended domains $D^{+\epsilon}$ and $D^{-\epsilon}$ are not shown

.

/ invertible tangent stiffness matrix. In this example, the maximal load of $\pm 6.9 \times 10^{-2}\mathrm{m}$ was applied to the lower and upper line of nodes, except the last clamped node. For the prescribed displacement field, $\mathbf{w}$ this maximal value decayed to 1×10^{-3} to the center of the plate. Due to numerical artifacts in the solver, the prescribed displacement should not be exactly zero at the center of the plate. Figure 20b shows the prescribed displacement field $\mathbf{w}$ at the first load step and for all successive load steps, the field increased linearly. All simulation parameters and material properties from the force control example were used, except the horizon ϵ was increased by a factor of two and $\epsilon = 8h$.

The simulation displayed in Figure 21 shows that the crack (indicated by the black crack zone center line) grows stably with increasing load. The white vertical line indicates the crack tip, which are the nodes with the maximal damage. The white horizontal line indicates the process zone where some bonds have softened to zero already, but not sufficiently many to create a surface increment over which a point in the upper half of the specimen is not influenced by forces from points in the lower half of the specimen. The damage after 11 load steps is shown in (a) and after 12

load steps in (b), respectively. The simulations again show that this is to be expected since for the displacement control the pre-crack opening displacement is small and the strain is now distributed uniformly within the intact material and stable crack growth can proceed. This is corroborated by the theory, recall Figure 1 where an example R curve with respect to the crack length is plotted. The dotted lines U_1 to U_5 sketch load in displacement. We see that these dotted lines run transversely to the R curve, which indicates that the load in displacement needs to be increased to develop the crack further. This leads to Equation 1 and the crack growth tends to be more stable as in force control.

7 Conclusion

We showed the explicit derivation of the tangent stiffness matrix for a regularized nonlinear pairwise (RNP) potential. First, we validated our approach against solutions from classical continuum mechanics for linear elasticity to validate the linear regime of the RNP potential. In addition, we validated our approach against a state-based model that uses the standard numerical derivations for assembling the stiffness matrix. Both validations showed good agreement for the linear regime without any damage. Furthermore, we could observe that the computational time of our approach was faster due to the reduced cost in the assembly of the tangent stiffness matrix. However, the Newton iterations per load step do not differ much. Second, the effect of softening bonds to the condition number of the tangent stiffness matrix $con(K)$ was investigated. Note that the crack growth was controlled by the user for this example. For a pre-crack square plate, the condition number increases linearly with respect to the softening bonds. However, the crack should not grow faster than one or one and a half mesh width per load step to keep the condition number small. Third, fracture simulations for a ASTM E8 like tension test and a pre-crack square plate were obtained for force control. Here, both simulations resulted in unstable crack growth as intended for force control. However, for the displacement control, we could show stable crack growth close to the boundary. Again, the novelty of our approach is that we soften the bonds and do not break them, which allows us an explicit derivation of the stiffness matrix.

To follow up on the proof of concept, the authors will implement the approach in their massive parallel C++ code to run larger simulations, which are needed for validation with experimental results and simulating more complex structures. Another aspect is to apply the method to three-dimensional structures. In addition, adaptive load steps are needed if one load step results in an unstable tangent stiffness matrix. Here, the authors like to investigate a more sophisticated choice of the reduced load steps based on numerical estimates or physical properties. Another aspect is to speed up the assembly of the stiffness matrix using the Fire algorithm[6] or Broyden-Fletcher–Goldfarb-Shanno (BFGS)[72] approach. A further challenge is to conduct a rigorous computational performance study of proposed methods to compare the reductions in computational cost. Here, all methods would need to run on the same computer hardware and one benchmark example would need to be defined for comparison. From the modeling aspect, the authors intend to apply the same numerical approach to the state-based RNP model[46] which includes dilatation forces.

Acknowledgments

This work was partly funded by DTIC Contract FA8075-14-D-0002/0007 and the Center of Computation & Technology at Louisiana State University. This material is based upon work supported by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant W911NF1610456 and the MURI Predicting and Controlling the Response of Particulate Systems through Grain-Scale Engineering - Cal-tech Prime W911NF1910245.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

Supporting information

The Python prototype of the implementation is available on GitHub[1] or on Zenodo[19]. Following Python packages were utilized: using NumPy[55, 68], SciPy[69], shapely, and Matplotlib[33].

[1] `https://github.com/diehlpk/AnalyticStiffnessPython`

References

[1] Burak Aksoylu, Fatih Celiker, and Orsan Kilicer. Nonlocal operators with local boundary conditions in higher dimensions. *Advances in Computational Mathematics*, 45(1):453–492, 2019.

[2] WE Amoldi. The principle of minimized iteration in the solution of the matrix eigenvafue problem, quart. a&. *J. Appl. Math*, 9:17–29, 1951.

[3] Ted L Anderson. *Fracture mechanics: fundamentals and applications*. CRC press, 2017.

[4] ASTM International. ASTM E8-99 Standard test methods for tension testing of metallic materials. `https://www.astm.org/Standards/E8.htm`, 2001.

[5] ASTM International. ASTM E561-20 Standard Test Method for KR Curve Determination. `http://www.astm.org/cgi-bin/resolver.cgi?E561`, 2020.

[6] Erik Bitzek, Pekka Koskinen, Franz Gähler, Michael Moseler, and Peter Gumbsch. Structural relaxation made simple. *Physical review letters*, 97(17):170201, 2006.

[7] Michael Breitenfeld. *Quasi-static non-ordinary state-based peridynamics for the modeling of 3D fracture*. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, 2014.

[8] Michael D Brothers, John T Foster, and Harry R Millwater. A comparison of different methods for calculating tangent-stiffness matrices in a massively parallel computational peridynamics code. *Computer Methods in Applied Mechanics and Engineering*, 279:247–267, 2014.

[9] AC Cassell and RE Hobbs. Numerical stability of dynamic relaxation analysis of non-linear structures. *International Journal for numerical methods in engineering*, 10(6):1407–1410, 1976.

[10] Xi Chen and Max Gunzburger. Continuous and discontinuous finite element methods for a peridynamics model of mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1237–1250, 2011.

[11] RH Dean and JW Hutchinson. Quasi-static steady crack growth in small-scale yielding. In *Fracture Mechanics*. ASTM International, 1980.

[12] P Diehl, R Lipton, and MA Schweitzer. Numerical verification of a bond-based softening peridynamic model for small displacements: deducing material parameters from classical linear theory. *INS Preprint No. 1630*, 2016.

[13] Patrick Diehl. Astm e8. Figshare, Nov 2020.

[14] Patrick Diehl. Emu-nodal discretization. Figshare, May 2020.

[15] Patrick Diehl. Pre-cracked square plate. Figshare, Nov 2020.

[16] Patrick Diehl. Validation 2d. Figshare, Jul 2020.

[17] Patrick Diehl. Validation of a one-dimensional bar. Figshare, 5 2020.

[18] Patrick Diehl. Displacement field for the two-dimensional benchmark. 10.6084/m9.figshare.15113508.v1, Aug 2021.

[19] Patrick Diehl. Quasistatic Fracture using Nonliner-Nonlocal Elastostatics with Explicit Tangent Stiffness Matrix. https://doi.org/10.5281/zenodo.5484312, September 2021.

[20] Patrick Diehl, Prashant K. Jha, Hartmut Kaiser, Robert Lipton, and Martin Lévesque. An asynchronous and task-based implementation of peridynamics utilizing hpx—the c++ standard library for parallelism and concurrency. *SN Applied Sciences*, 2(12):2144, Dec 2020.

[21] Patrick Diehl, Robert Lipton, Thomas Wick, and Mayank Tyagi. A comparative review of peridynamics and phase-field models for engineering fracture mechanics. *engrXiv*, 2021.

[22] Patrick Diehl, Serge Prudhomme, and Martin Lévesque. A review of benchmark experiments for the validation of peridynamics models. *Journal of Peridynamics and Nonlocal Modeling*, 1(1):14–35, 2019.

[23] Qiang Du. Nonlocal calculus of variations and well-posedness of peridynamics. In *Handbook of peridynamic modeling*, pages 101–124. Chapman and Hall/CRC, 2016.

[24] Etienne Emmrich and Olaf Weckner. The peridynamic equation and its spatial discretisation. *Mathematical Modelling and Analysis*, 12(1):17–27, 2007.

[25] Andris Freimanis and Ainars Paeglitis. Mesh sensitivity in peridynamic quasi-static simulations. *Procedia Engineering*, 172:284–291, 2017.

[26] Alan Arnold Griffith. Vi. the phenomena of rupture and flow in solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 221(582-593):163–198, 1921.

[27] Xin Gu, Erdogan Madenci, and Qing Zhang. Revisit of non-ordinary state-based peridynamics. *Engineering fracture mechanics*, 190:31–52, 2018.

[28] Max Gunzburger and Richard B Lehoucq. A nonlocal vector calculus with application to nonlocal boundary value problems. *Multiscale Modeling & Simulation*, 8(5):1581–1598, 2010.

[29] C Gurney and J Hunt. Quasi-static crack propagation. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 299(1459):508–524, 1967.

[30] Yile Hu, Hailong Chen, Benjamin W Spencer, and Erdogan Madenci. Thermomechanical peridynamic analysis with irregular non-uniform domain discretization. *Engineering Fracture Mechanics*, 197:92–113, 2018.

[31] YL Hu and Erdogan Madenci. Bond-based peridynamic modeling of composite laminates with arbitrary fiber orientation and stacking sequence. *Composite structures*, 153:139–175, 2016.

[32] Dan Huang, Guangda Lu, and Pizhong Qiao. An improved peridynamic approach for quasi-static elastic deformation and brittle fracture analysis. *International Journal of Mechanical Sciences*, 94:111–122, 2015.

[33] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[34] Siavash Jafarzadeh, Farzaneh Mousavi, Adam Larios, and Florin Bobaru. A general and fast convolution-based method for peridynamics: applications to elasticity and brittle fracture. *arXiv*, 2021.

[35] Siavash Jafarzadeh, Longzhen Wang, Adam Larios, and Florin Bobaru. A fast convolution-based method for peridynamic transient diffusion in arbitrary domains. *Computer Methods in Applied Mechanics and Engineering*, 375:113633, 2021.

[36] Prashant K. Jha and Patrick Diehl. Nlmech: Implementation of finite difference/meshfree discretization of nonlocal fracture models. *Journal of Open Source Software*, 6(65):3020, 2021.

[37] Prashant K. Jha and Robert Lipton. Numerical analysis of nonlocal fracture models in hölder space. *SIAM Journal on Numerical Analysis*, 56(2):906–941, 2018.

[38] Prashant K Jha and Robert Lipton. Numerical convergence of nonlinear nonlocal continuum models to local elastodynamics. *International Journal for Numerical Methods in Engineering*, 114(13):1389–1410, 2018.

[39] Prashant K Jha and Robert Lipton. Kinetic relations and local energy balance for lefm from a nonlocal peridynamic model. *International Journal of Fracture*, 226(1):81–95, 2020.

[40] B Kilic and Erdogan Madenci. An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics*, 53(3):194–204, 2010.

[41] David C Lay, Steven R Lay, and Judi J McDonald. *Linear algebra and its applications*. Pearson, 2016.

[42] Robert Lipton. Dynamic brittle fracture as a small horizon limit of peridynamics. *Journal of Elasticity*, 117(1):21–50, 2014.

[43] Robert Lipton. Cohesive dynamics and brittle fracture. *Journal of Elasticity*, 124(2):143–191, 2016.

[44] Robert Lipton and Prashant K. Jha. Nonlocal elastodynamics and fracture. *Nonlinear Differ. Equ. Appl.*, 28(23), 2021.

[45] Robert Lipton, Richard B Lehoucq, and Prashant K Jha. Complex fracture nucleation and evolution with nonlocal elastodynamics. *Journal of Peridynamics and Nonlocal Modeling*, 133(1):122–130, 2019.

[46] Robert Lipton, Eyad Said, and Prashant K Jha. Dynamic brittle fracture from nonlocal double-well potentials: A state-based model. *Handbook of Nonlocal Continuum Mechanics for Materials and Structures*, pages 1–27, 2018.

[47] Robert Lipton, Eyad Said, and Prashant K Jha. Free damage propagation with memory. *Journal of Elasticity*, 133:129–153, 2018.

[48] Robert Lipton, Stewart Silling, and Richard Lehoucq. Complex fracture nucleation and evolution with nonlocal elastodynamics. *arXiv preprint arXiv:1602.00247*, 2016.

[49] David Littlewood. Roadmap for software implementation. In *Handbook of Peridynamic Modeling*, pages 147–178. Chapman and Hall/CRC, 2016.

[50] James N Lyness and Cleve B Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, 1967.

[51] JN Lyness. Differentiation formulas for analytic functions. *Mathematics of Computation*, pages 352–362, 1968.

[52] Richard W Macek and Stewart A Silling. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design*, 43(15):1169–1178, 2007.

[53] Erdogan Madenci, Mehmet Dorduncu, Atila Barut, and Nam Phan. Weak form of peridynamics for nonlocal essential and natural boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 337:598–631, 2018.

[54] Yozo Mikata. Analytical solutions of peristatic and peridynamic problems for a 1d infinite rod. *International Journal of Solids and Structures*, 49(21):2887–2897, 2012.

[55] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[56] Naveen Prakash and Ross J Stewart. A multi-threaded method to assemble a sparse stiffness matrix for quasi-static solutions of linearized bond-based peridynamics. *Journal of Peridynamics and Nonlocal Modeling*, pages 1–35, 2020.

[57] Serge Prudhomme and Patrick Diehl. On the treatment of boundary conditions for bond-based peridynamic models. *Computer Methods in Applied Mechanics and Engineering*, 372:113391, 2020.

[58] Timon Rabczuk and Huilong Ren. A peridynamics formulation for quasi-static fracture and contact in rock. *Engineering Geology*, 225:42–48, 2017.

[59] JR Rice. Thermodynamics of the quasi-static growth of griffith cracks. *Journal of the Mechanics and Physics of Solids*, 26(2):61–78, 1978.

[60] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[61] Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126, 1981.

[62] Martin H Sadd. *Elasticity: theory, applications, and numerics*. Academic Press, 2009.

[63] Yoshinori SHIIHARA, Shoki TANAKA, and Nobuhiro YOSHIKAWA. Fast quasi-implicit nosb peridynamic simulation based on fire algorithm. *Mechanical Engineering Journal*, 6(3):18–00363, 2019.

[64] Stewart A Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175–209, 2000.

[65] Stewart A Silling and Ebrahim Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17-18):1526–1535, 2005.

[66] Stewart A Silling, M Epton, O Weckner, Ji Xu, and E23481501120 Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184, 2007.

[67] BHV Topping and AI Khan. Parallel computation schemes for dynamic relaxation. *Engineering computations*, 11(6):513–548, 1994.

[68] S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, 2011.

[69] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[70] Fei Wang, Yu'e Ma, Yanning Guo, and Wei Huang. Studies on quasi-static and fatigue crack propagation behaviours in friction stir welded joints using peridynamic theory. *Advances in Materials Science and Engineering*, 2019, 2019.

[71] Hong Wang and Hao Tian. A fast galerkin method with efficient matrix assembly and storage for a peridynamic model. *Journal of Computational Physics*, 231(23):7730–7738, 2012.

[72] Jian-Ying Wu, Yuli Huang, and Vinh Phu Nguyen. On the bfgs monolithic algorithm for the unified phase field damage theory. *Computer Methods in Applied Mechanics and Engineering*, 360:112704, 2020.

[73] Mirco Zaccariotto, Fabio Luongo, U Galvanetto, et al. Examples of applications of the peridynamic theory to the solution of static equilibrium problems. *The Aeronautical Journal*, 119(1216):677–700, 2015.