Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.026922

ARTICLE

ABAQUS and ANSYS Implementations of the Peridynamics-Based Finite Element Method (PeriFEM) for Brittle Fractures

Fei Han[*]**, Zhibin Li, Jianyu Zhang, Zhiying Liu, Chen Yao and Wenping Han**

State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics,
Dalian University of Technology, Dalian, 116023, China

*Corresponding Author: Fei Han. Email: hanfei@dlut.edu.cn

Received: 03 October 2022 Accepted: 01 December 2022

ABSTRACT

In this study, we propose the first unified implementation strategy for peridynamics in commercial finite element method (FEM) software packages based on their application programming interface using the peridynamics-based finite element method (PeriFEM). Using ANSYS and ABAQUS as examples, we present the numerical results and implementation details of PeriFEM in commercial FEM software. PeriFEM is a reformulation of the traditional FEM for solving peridynamic equations numerically. It is considered that the non-local features of peridynamics yet possesses the same computational framework as the traditional FEM. Therefore, this implementation benefits from the consistent computational frameworks of both PeriFEM and the traditional FEM. An implicit algorithm is used for both ANSYS and ABAQUS; however, different convergence criteria are adopted owing to their unique features. In ANSYS, APDL enables users to conveniently obtain broken-bond information from UPFs; thus, the convergence criterion is chosen as no new broken bond. In ABAQUS, obtaining broken-bond information is not convenient for users; thus, the default convergence criterion is used in ABAQUS. The codes integrated into ANSYS and ABAQUS are both verified through benchmark examples, and the computational convergence and costs are compared. The results show that, for some specific examples, ABAQUS is more efficient, whereas the convergence criterion adopted in ANSYS is more robust. Finally, 3D examples are presented to demonstrate the ability of the proposed approach to deal with complex engineering problems.

KEYWORDS

PeriFEM; peridynamics; ANSYS; ABAQUS; damage; fracture

1 Introduction

Establishing mathematical models for natural phenomena is the second paradigm in the scientific investigation [1]. More specifically, in the engineering and physics fields, many problems are traditionally summarized as a (group of) partial differential equation(s). However, the analytical solutions of these equations are usually difficult to obtain, and thus, we have to rely on numerical methods [2]. The finite element method (FEM) is one of the most popular numerical methods due to its standardized analysis process and applicability to a wide variety of problems.

However, the development of the FEM has not been smooth sailing, and it is said that even a well-known journal at that time shunned papers on the finite element method for many years [3]. In the 1960s, things turned around benefiting from Ed Wilson's liberal distribution of his first programs, making more and more people realize the value of the FEM and devote themselves to this field. Then in the next few decades, numerous FEM software sprang up, which greatly promoted the application of the FEM in engineering practice. In particular, commercial FEM software uses interactive windows to make applications more convenient and their appearance also makes the finite element analysis easier.

With the development of cutting-edge science and technologies over the past few years, the service environment of engineering structures has become increasingly complicated, and their analysis is facing new challenges, such as fracture. For these challenges, traditional commercial FEM software is probably to be ineffective. There are two main strategies to approach these problems: (1) developers integrate advanced techniques into the software directly or (2) users incorporate advanced techniques into the software using a programmable interface. The first option may be more suitable for more mature technology because software development is expensive in terms of time and funding. The second option is better able to adapt to the most advanced techniques and accumulate experience for the first option. For example, before ABAQUS launched a version containing the extended finite element method (XFEM), Fang et al. [4] had implemented XFEM simulations in it. For other methods that have received widespread attention, such as the cohesive zone model and the phase field model of fracture [5,6], there are also related works in which these methods were implemented in existing software. For example, Lindgaard et al. [7] presented a cohesive zone finite element method implemented via user programmable features (UPFs) in ANSYS; and Msekh et al. [8–10] incorporated the phase field model of fracture with ABAQUS via subroutines.

Peridynamics has also attracted considerable research attention as a new non-local theory [11,12], and has been applied to many engineering problems, such as the stress analysis in the vicinity of the crack tip [13], fracture modelling of reinforced concrete [14,15], and the inelastic fracture [16], etc. The governing equations in peridynamics are integral–differential equations rather than partial differential equations. Therefore, discontinuities are naturally tolerated, which offers considerable advantages when simulating fracture problems. Thus, incorporating peridynamics into commercial FEM software could considerably promote its engineering applications. However, because of the integration aspect of peridynamic equations, only a few studies have used element-based methods to solve them [17–19]; most studies use particle-based methods that can directly approximate the integration as a Riemann summation [20–22], which is hardly compatible with commercial FEM software. To the best of the authors' knowledge, only a few studies have implemented peridynamics in commercial FEM software [23–26]. Diyaroglu et al. [23] used a truss element to present the peridynamic bond [27], and peridynamics simulations were implemented in ANSYS. Huang et al. [24] defined the collection of all peridynamic particles in a horizon as a new element and conducted peridynamics simulations in ABAQUS. However, this type of element is undefined around the boundary, and thus a coupled model had to be used. Bie et al. [25] overcame the aforementioned problems around the boundary and implemented dual peridynamics in ABAQUS. However, they had to define many types of elements. Anicode et al. [26] used native MATRIX27 elements to perform peridynamic analysis in ANSYS. Notably, in existing reports, although peridynamics was implemented in FEM software, most of them used particles and regular grid for spatial discretization. On the other hand, an element-based implementation of peridynamics in commercial FEM software is still inadequate, particularly with irregular meshes. To our knowledge, Ren et al. [28,29] achieved the element-based peridynamics through a packaged peridynamic module in LS-DYNA. In this work, we consider the implementation based on the Application Programming Interface (API) of the software, which will be beneficial to the users for their specific requirements.

In this study, we propose a unified implementation technique for peridynamics in commercial FEM software based on the peridynamics-based finite element method (PeriFEM) [30,31]. PeriFEM was established in [30], and in [31], an adaptive continuous/discrete element technique was proposed as a supplement to [30]. However, both Han et al. [30] and Li et al. [31] employed an in-house code. In contrast, in this work, we concentrate on the implementation of PeriFEM in commercial FEM software based on its application programming interface (API) [32], which should be beneficial to users by better adapting to their specific requirements. PeriFEM is an element-based method in which only two types of elements are required. More importantly, the computational framework of PeriFEM is consistent with that of traditional FEM; thus, it can be conveniently implemented in any commercial FEM software with a programmable interface. Therefore, we implemented peridynamics simulations in ANSYS and ABAQUS using PeriFEM. The implementation process in these two pieces of software is similar, which shows that PeriFEM can be easily extended to other commercial FEM software packages. In addition, two different convergence criteria for the implicit iteration process are used in ANSYS and ABAQUS.

The rest of this paper is organized as follows. Section 2 reviews the basic formulations of bond-based peridynamics. Section 3 introduces PeriFEM in detail, including the definition of elements, shape functions, and establishment of linear equations. Section 4 is devoted to the numerical algorithm of PeriFEM and two convergence criteria. Section 5 describes the implementation of PeriFEM in ANSYS and ABAQUS. Numerical examples are presented in Section 6 to verify the proposed algorithms and programs. Finally, conclusions are drawn in Section 7.

2 Review of Peridynamics

Peridynamics is a reformulation of elasticity theory for discontinuities and long-range forces [11], which assumes that a point in the peridynamic continuum can interact with all points in its neighborhood through bonds. According to the mode of action of the force associated with the bonds, peridynamic formulations can be classified into bond-based and state-based peridynamics [12]. Here, we focus on the first one in the quasi-static case, and all mentions of “peridynamics” in subsequent sections refer to bond-based peridynamics.

Similar to classical elasticity theory, there are three groups of basic equations in peridynamics, i.e., the equilibrium, constitutive, and kinematic equations. These can be respectively expressed as

$$\int_{H_{\delta(\boldsymbol{x})}} \boldsymbol{f}(\boldsymbol{\xi})\,\mathrm{d}V_{\xi} + \boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{0}, \quad \forall \boldsymbol{x} \in \Omega, \tag{1a}$$

$$\boldsymbol{f}(\boldsymbol{\xi}) = \boldsymbol{C}(\boldsymbol{\xi}) \cdot \boldsymbol{\eta}(\boldsymbol{\xi}), \quad \forall \boldsymbol{x}' \in H_{\delta(\boldsymbol{x})}, \boldsymbol{x} \in \Omega, \tag{1b}$$

$$\boldsymbol{\eta}(\boldsymbol{\xi}) = \boldsymbol{u}(\boldsymbol{x}') - \boldsymbol{u}(\boldsymbol{x}), \quad \forall \boldsymbol{x}', \boldsymbol{x} \in \Omega, \tag{1c}$$

where $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ is the reference configuration, $\boldsymbol{x}$ and $\boldsymbol{x}'$ are points in Ω, $H_{\delta(\boldsymbol{x})}$ is the neighborhood of point $\boldsymbol{x}$ with a cut-off radius δ, $\boldsymbol{b}(\boldsymbol{x})$ is the external body force, $\boldsymbol{\xi} = \boldsymbol{x}' - \boldsymbol{x}$ is a relative position vector referred to as a bond, and $\boldsymbol{\eta}(\boldsymbol{\xi})$ and $\boldsymbol{f}(\boldsymbol{\xi})$ are the measure of deformation and the pairwise force vector of bond $\boldsymbol{\xi}$, respectively. In addition to these basic equations, there are displacement boundary conditions in peridynamics, which can be expressed as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^{*}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_{u}, \tag{2}$$

here, $\partial\Omega_u$ refers to a part of the boundary of Ω where a prescribed displacement is applied. In addition, traction boundary conditions are not considered directly in peridynamics [33].

$\boldsymbol{C}(\boldsymbol{\xi})$ in Eq. (1b) is the micro-modulus tensor, which is defined as [11]

$$\boldsymbol{C}(\boldsymbol{\xi}) = c(\boldsymbol{\xi})\boldsymbol{\xi} \otimes \boldsymbol{\xi}, \tag{3}$$

where $c(\boldsymbol{\xi})$ denotes the micro-modulus coefficient. Furthermore, $c(\boldsymbol{\xi})$ can be multiplicatively decomposed into a bond length-related part $\hat{c}(\|\boldsymbol{\xi}\|)$ and a non-related part $\tilde{c}(\theta,\varphi)$ (where θ and φ are the polar and azimuthal angles of $\boldsymbol{\xi}$ in spherical coordinates, respectively), i.e.,

$$c(\boldsymbol{\xi}) = \hat{c}(\|\boldsymbol{\xi}\|)\tilde{c}(\theta,\varphi), \tag{4}$$

where $\hat{c}(\|\boldsymbol{\xi}\|)$ can be a function of $\|\boldsymbol{\xi}\|$ in the form of a constant (e.g., $\hat{c}(\|\boldsymbol{\xi}\|) = 1$ [21]), linear (e.g., $\hat{c}(\|\boldsymbol{\xi}\|) = 1 - \|\boldsymbol{\xi}\|/\delta$ [34]), or exponential function (e.g., $\hat{c}(\|\boldsymbol{\xi}\|) = e^{-\|\boldsymbol{\xi}\|/l}$ [35], where l is a characteristic length related to the fracture process zone, see [36]), among others. Here, we adopted an exponential function. $\tilde{c}(\theta,\varphi)$ can be determined according to the deformation energy equivalence between peridynamics and classical continuum mechanics. For isotropic materials, $\tilde{c}(\theta,\varphi)$ degenerates to a constant $\tilde{c}^0$ related to Young's modulus E, i.e.,

$$\tilde{c}(\theta,\varphi) = \tilde{c}^0 = \begin{cases} \dfrac{3E}{\pi\int_0^\delta r^6 e^{-r/l}\mathrm{d}r}, & \text{for } d = 3, \\ \dfrac{3E}{\pi\int_0^\delta r^5 e^{-r/l}\mathrm{d}r}, & \text{for } d = 2 \text{ and plane-stress}, \\ \dfrac{16E}{5\pi\int_0^\delta r^5 e^{-r/l}\mathrm{d}r}, & \text{for } d = 2 \text{ and plane-strain}, \\ \dfrac{E}{\int_0^\delta r^4 e^{-r/l} dr}, & \text{for } d = 1. \end{cases} \tag{5}$$

For anisotropic materials, the calculation of $\tilde{c}(\theta,\varphi)$ is presented in the Appendix.

When material failure needs to be considered, the simplest method is to allow the bonds to break. Here, we use the criterion proposed in [21] to introduce material failure into the corresponding constitutive equation. The criterion is

$$\mu(\boldsymbol{\xi},t) = \begin{cases} 1, & \text{if } s(\boldsymbol{\xi},\tau) < s_{crit} \text{ for all } 0 \le \tau \le t, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where t and τ denote computational steps, μ is a history-dependent scalar-valued function that records whether $\boldsymbol{\xi}$ is broken, $s = \dfrac{\|\boldsymbol{\xi}+\boldsymbol{\eta}\| - \|\boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|}$ is the bond stretch, and s_{crit} is the critical bond stretch, which is generally considered to be related to the critical energy release rate [21,37]. Multiplying the right-hand side of Eq. (3) by $\mu(\boldsymbol{\xi},t)$ yields a constitutive equation that includes bond breaking (see [21], Eq. (18)). Based on the broken bonds, one can define the effective damage at each point $\boldsymbol{x}$ as [21]

$$\phi(\boldsymbol{x}) = \frac{\int_{H_\delta(\boldsymbol{x})}(1-\mu(\boldsymbol{\xi},t))\mathrm{d}V_\xi}{\int_{H_\delta(\boldsymbol{x})}\mathrm{d}V_\xi} \tag{7}$$

to determine macroscopic crack paths.

3 Peridynamics-Based Finite Element Method (PeriFEM)

Standardization of the analysis process is one of the most salient features of the FEM, which is also one of the main reasons for its wide use in engineering analysis. In our previous studies, we derived the linear equations of PeriFEM based on the principle of minimum potential energy [30] and the principle of virtual work [31]. In the present study, we will show the general steps of PeriFEM based on the general steps of the FEM, as described in [2]. The reader will find that the computational framework of PeriFEM is consistent with that of the FEM, which enables the implementation of PeriFEM in FEM software.

3.1 Discretization and Selection of Element Types

This section is devoted to the spatial discretization and the definition of elements in PeriFEM. There are two types of element in this method. One is the *local element* for local quantities, such as body force, and the other is the *peridynamic element* (non-local element) for non-local quantities, such as peridynamic long-range force.

The definition of local elements is a generalization of the *finite element* in the FEM. In the FEM, a configuration Ω is divided into a finite number of elements $\{e_i\}_{i=1}^{m}$, that is, $\Omega = \bigcup_{i=1}^{m} e_i$, where m is the element number. These elements are non-overlapping but share common edges and nodes with adjacent elements. An element that satisfies the above definitions is called a finite element. In PeriFEM, the configuration Ω is also discretized by a finite number of elements, that is, $\Omega = \bigcup_{i=1}^{m} e_i$. These elements are also non-overlapping but do not have to share edges or nodes. In other words, the only difference between a local element and a finite element is that the former has no requirement on whether nodes and edges are shared between adjacent elements. Therefore, any type of finite element in the FEM can be generalized to a local element and applied in PeriFEM.

The definition of peridynamic elements is based on local elements. In brief, a peridynamic element is composed of two local elements, as shown in Fig. 1 (note that the local elements are not limited to quadrilaterals for 2D and hexahedrons for 3D). For any two local elements, denoted as e_i and e_j, if they can interact with each other within the peridynamic horizon δ, then a peridynamic element denoted as $\bar{e}_k$ will be generated from them. In more formal terms, suppose the nodes of e_i and e_j are $\left[\mathrm{P}_{i_1}, \mathrm{P}_{i_2}, \cdots, \mathrm{P}_{in_i}\right]$ and $\left[\mathrm{P}_{j_1}, \mathrm{P}_{j_2}, \cdots, \mathrm{P}_{jn_j}\right]$, respectively, where $\mathrm{P}_{\alpha_\beta}$ $(\alpha = i, j; \beta = 1, 2, \cdots, n_\alpha) \in \{\mathrm{P}_l\}_{l=1}^{n}$, which are the global nodes of Ω after being discretized by local elements $\{e_i\}_{i=1}^{m}$. The nodes of $\bar{e}_k$ are $\left[\mathrm{P}_{k_1}, \mathrm{P}_{k_2}, \cdots, \mathrm{P}_{k_{\bar{n}_k}}\right] = \left[\mathrm{P}_{j_1}, \mathrm{P}_{j_2}, \cdots, \mathrm{P}_{jn_j}, \mathrm{P}_{i_1}, \mathrm{P}_{i_2}, \cdots, \mathrm{P}_{in_i}\right]$, where $\bar{n}_k = n_j + n_i$. Finally, a set of peridynamic elements $\{\bar{e}_k\}_{k=1}^{\bar{m}}$ is generated, where $\bar{m}$ denotes the number of peridynamic elements.

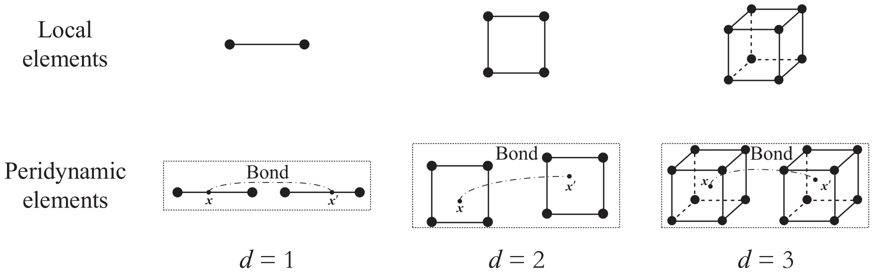


Figure 1: Schematic of local and peridynamic elements

3.2 Selection of a Displacement Function

Now, we introduce an approximation technique for calculating element displacement. Such a technique for local elements is similar to that used in the FEM, whereas for peridynamic elements it is based on the approximation of the corresponding local elements.

For any local element e_i, the possible displacements that can be approximated using the interpolation technique can be expressed as

$$\boldsymbol{u}_i(\boldsymbol{x}) = \boldsymbol{N}_i(\boldsymbol{x})\boldsymbol{d}_i, \tag{8}$$

where $\boldsymbol{N}_i(x)$ and $\boldsymbol{d}_i$ are respectively the local shape function matrix and the local nodal displacement vector of e_i, which are defined as

$$\boldsymbol{N}_i(\boldsymbol{x}) = \begin{bmatrix} N_{i_1}(\boldsymbol{x}) & 0 & 0 & N_{i_2}(\boldsymbol{x}) & 0 & 0 & \cdots & N_{in_i}(\boldsymbol{x}) & 0 & 0 \\ 0 & N_{i_1}(\boldsymbol{x}) & 0 & 0 & N_{i_2}(\boldsymbol{x}) & 0 & \cdots & 0 & N_{in_i}(\boldsymbol{x}) & 0 \\ 0 & 0 & N_{i_1}(\boldsymbol{x}) & 0 & 0 & N_{i_2}(\boldsymbol{x}) & \cdots & 0 & 0 & N_{in_i}(\boldsymbol{x}) \end{bmatrix}, \tag{9}$$

$$\boldsymbol{d}_i = \begin{bmatrix} u_{i_1} & v_{i_1} & w_{i_1} & u_{i_2} & v_{i_2} & w_{i_2} & \cdots & u_{in_i} & v_{in_i} & w_{in_i} \end{bmatrix}^T. \tag{10}$$

Here, N_{i_l} $(l = 1, 2, \cdots, n_i)$ is the shape function of node P_{i_l} and $u_{i_l}, v_{i_l}, w_{i_l}$ denote the X, Y, Z displacements of node P_{i_l}, respectively.

For any peridynamic element $\bar{e}_k$, assuming that it is generated from local elements e_i and e_j, the displacements within $\bar{e}_k$ can be approximated as

$$\bar{\boldsymbol{u}}_k(\boldsymbol{x}', \boldsymbol{x}) = \begin{bmatrix} \boldsymbol{u}_j(\boldsymbol{x}') \\ \boldsymbol{u}_i(\boldsymbol{x}) \end{bmatrix} = \bar{\boldsymbol{N}}_k(\boldsymbol{x}', \boldsymbol{x})\,\bar{\boldsymbol{d}}_k, \tag{11}$$

where

$$\bar{\boldsymbol{N}}_k(\boldsymbol{x}', \boldsymbol{x}) = \begin{bmatrix} \boldsymbol{N}_j(\boldsymbol{x}') & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N}_i(\boldsymbol{x}) \end{bmatrix}, \tag{12}$$

$$\bar{\boldsymbol{d}}_k = \begin{bmatrix} \boldsymbol{d}_j \\ \boldsymbol{d}_i \end{bmatrix} \tag{13}$$

are the peridynamic shape function matrix and the peridynamic nodal displacement vector of $\bar{e}_k$, respectively.

3.3 Force/Deformation Relationship

Now, we express the constitutive relation in terms of the unknown nodal displacements. Notably, the constitutive response stems from peridynamic long-range forces; thus, the expression of the constitutive relation is based on peridynamic elements.

First, the measure of deformation $\boldsymbol{\eta}$ that is associated with a peridynamic element $\bar{e}_k$ can be approximately expressed as

$$\bar{\boldsymbol{\eta}}_k(\boldsymbol{x}', \boldsymbol{x}) = \boldsymbol{u}_j(\boldsymbol{x}') - \boldsymbol{u}_i(\boldsymbol{x}) = \bar{\boldsymbol{B}}_k(\boldsymbol{x}', \boldsymbol{x})\,\bar{\boldsymbol{d}}_k, \tag{14}$$

where

$$\bar{\boldsymbol{B}}_k(\boldsymbol{x}', \boldsymbol{x}) = \bar{\boldsymbol{H}}\bar{\boldsymbol{N}}(\boldsymbol{x}', \boldsymbol{x}) \tag{15}$$

is the difference matrix of the shape function of $\bar{e}_k$ and

$$\bar{\boldsymbol{H}} = [\boldsymbol{I}, -\boldsymbol{I}] \tag{16}$$

is the difference operator matrix, with $\boldsymbol{I}$ being an identity matrix of dimension d.

Then, based on the constitutive equation, that is, Eq. (1b), the approximate pair force vector $\boldsymbol{f}$ can be expressed as

$$\bar{\boldsymbol{f}}_k(\boldsymbol{x}', \boldsymbol{x}) = \boldsymbol{D}(\boldsymbol{\xi})\bar{\boldsymbol{B}}_k(\boldsymbol{x}', \boldsymbol{x})\,\bar{\boldsymbol{d}}_k, \tag{17}$$

where

$$\boldsymbol{D}(\boldsymbol{\xi}) = c(\boldsymbol{\xi})\mu(\boldsymbol{\xi}, t)\begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1\xi_3 \\ \xi_2\xi_1 & \xi_2^2 & \xi_2\xi_3 \\ \xi_3\xi_1 & \xi_3\xi_2 & \xi_3^2 \end{bmatrix} \tag{18}$$

is the matrix form of the micro-modulus tensor $\boldsymbol{C}(\boldsymbol{\xi})$ for $d = 3$.

3.4 Deriving the Element Stiffness Matrix and Equations

In this section, we show how to derive the element stiffness matrix and linear equations for nodal displacement according to the principle of minimum potential energy.

For any peridynamic element $\bar{e}_k$ that is generated from local elements e_i and e_j, the potential energy [38] of $\bar{e}_k$ is

$$\Pi_k(\boldsymbol{u}) = \frac{1}{4}\int_{\bar{e}_k}\int_{\mathscr{H}_{\delta(\boldsymbol{x})}\cap\bar{e}_k} \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{x}) \cdot \boldsymbol{\eta}(\boldsymbol{x}', \boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}'}\mathrm{d}V_{\boldsymbol{x}} - \int_{ej} \boldsymbol{p}(\boldsymbol{x}') \cdot \boldsymbol{u}(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'} - \int_{ei} \boldsymbol{p}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}}, \tag{19}$$

where $\boldsymbol{p}$ is composed of external body forces and the long-range forces associated with the points that do not belong to the peridynamic element $\bar{e}_k$. This energy is unknown for a given element[1]. Furthermore, according to the contents of Sections 3.2 and 3.3, the potential energy is a function of the nodal displacements:

$$\begin{aligned}\Pi_k(\bar{\boldsymbol{d}}_k) &= \frac{1}{4}\int_{\bar{e}_k}\int_{\mathscr{H}_{\delta(\boldsymbol{x})}\cap\bar{e}_k} \bar{\boldsymbol{f}}_k(\boldsymbol{x}', \boldsymbol{x}) \cdot \bar{\boldsymbol{\eta}}_k(\boldsymbol{x}', \boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}'}\mathrm{d}V_{\boldsymbol{x}} - \int_{ej} \boldsymbol{p}(\boldsymbol{x}') \cdot \boldsymbol{u}_j(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'} - \int_{ei} \boldsymbol{p}(\boldsymbol{x}) \cdot \boldsymbol{u}_i(\boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}} \\ &= \frac{1}{4}\bar{\boldsymbol{d}}_k^T\bar{\boldsymbol{K}}_k\bar{\boldsymbol{d}}_k - \bar{\boldsymbol{d}}_k^T\bar{\boldsymbol{P}}_k,\end{aligned} \tag{20}$$

where

$$\bar{\boldsymbol{K}}_k = \int_{\bar{e}_k}\int_{\mathscr{H}_{\delta(\boldsymbol{x})}\cap\bar{e}_k} \bar{\boldsymbol{B}}_k^T(\boldsymbol{x}', \boldsymbol{x})\,\boldsymbol{D}(\boldsymbol{\xi})\bar{\boldsymbol{B}}_k(\boldsymbol{x}', \boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}'}\mathrm{d}V_{\boldsymbol{x}}, \tag{21}$$

$$\bar{\boldsymbol{P}}_k = \begin{bmatrix} \int_{e_j} \boldsymbol{N}_j^T(\boldsymbol{x}')\,\boldsymbol{p}(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'} \\ \int_{e_i} \boldsymbol{N}_i^T(\boldsymbol{x})\,\boldsymbol{p}(\boldsymbol{x})\,\mathrm{d}V_{\boldsymbol{x}} \end{bmatrix}. \tag{22}$$

Then, taking the first variation, we have

$$\frac{1}{2}\bar{\boldsymbol{K}}_k\bar{\boldsymbol{d}}_k = \bar{\boldsymbol{P}}_k. \tag{23}$$

[1] Although it is unknown for a given element, this does not matter as we only need the total external force vector after assembling the whole structure.

3.5 Assembling the Element Equations

With the element stiffness matrix at hand, we can now assemble the global stiffness matrix, global force vector, and global linear equations.

For convenience, we introduce the transform matrix of the degree of freedom for the nodes of $\bar{e}_k$, denoted by $\bar{\boldsymbol{G}}_k$. This matrix satisfies

$$\bar{\boldsymbol{d}}_k = \bar{\boldsymbol{G}}_k \boldsymbol{d}, \tag{24}$$

where $\boldsymbol{d}$ is the total nodal displacement vector. Then, we can obtain the global linear equations

$$\frac{1}{2}\bar{\boldsymbol{K}}\boldsymbol{d} = \bar{\boldsymbol{P}}, \tag{25}$$

where

$$\bar{\boldsymbol{K}} = \sum_{k=1}^{\bar{m}} \bar{\boldsymbol{G}}_k^T \bar{\boldsymbol{K}}_k \bar{\boldsymbol{G}}_k, \tag{26}$$

$$\bar{\boldsymbol{P}} = \sum_{k=1}^{\bar{m}} \bar{\boldsymbol{G}}_k^T \bar{\boldsymbol{P}}_k, \tag{27}$$

are the global stiffness matrix and global force vector, respectively.

In addition, $\bar{\boldsymbol{P}}_k$ is unknown for a given peridynamic element, and thus, it is infeasible to calculate $\bar{\boldsymbol{P}}$ using Eq. (27). However, for the entire structure, $\bar{\boldsymbol{P}}$ represents the external force applied to Ω. Thus, we know that

$$\bar{\boldsymbol{P}} = \boldsymbol{F} = \sum_{i=1}^{m} \boldsymbol{G}_i^T \boldsymbol{F}_i, \tag{28}$$

where

$$\boldsymbol{F}_i = \int_{e_i} \boldsymbol{N}_i^T(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})\mathrm{d}V_{\boldsymbol{x}}. \tag{29}$$

$\boldsymbol{G}_i$ is the transform matrix of the degree of freedom for the nodes of e_i and satisfies

$$\boldsymbol{d}_i = \boldsymbol{G}_i \boldsymbol{d}. \tag{30}$$

Remark 1. PeriFEM is also suitable for dual-horizon peridynamics [39].

Remark 2. In the present work, the Gaussian quadrature method is used, which has been discussed in detail in [31]. For other numerical quadrature methods, refer to [40].

Remark 3. In [30], we compared the accuracy of peridynamics implemented using a mesh-free framework and PeriFEM. There are also mesh-free methods [41] for brittle fracture modeling in addition to mesh-free peridynamics.

4 Numerical Algorithm

4.1 PeriFEM Algorithm

In this subsection, we introduce the PeriFEM algorithm. Considering that the failure progress of structures involves material nonlinearities, the boundary conditions are applied progressively in N incremental steps. The algorithm is shown in Fig. 2, which depicts a detailed flowchart of PeriFEM.

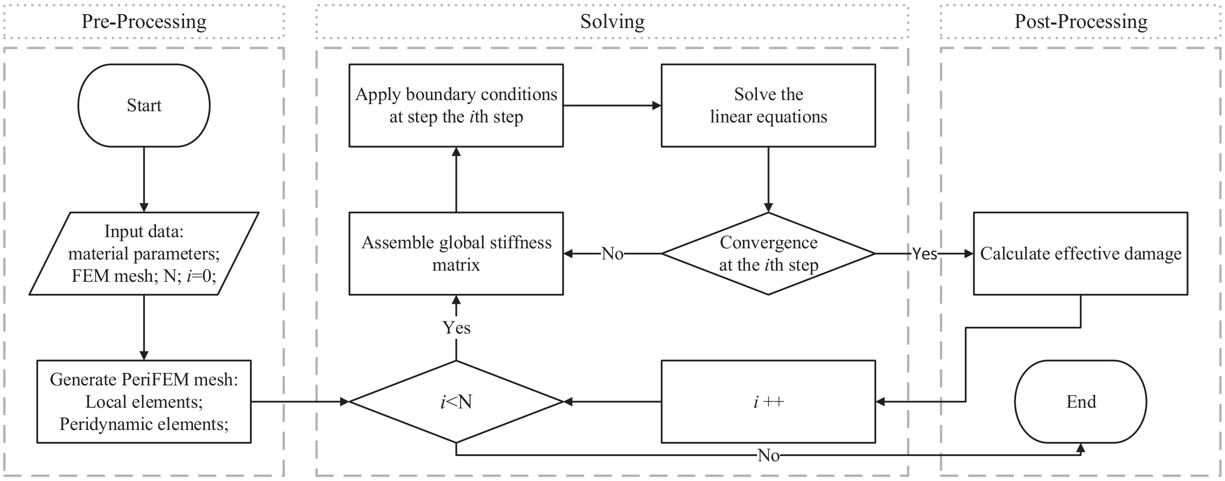


Figure 2: Flowchart of the PeriFEM algorithm

As shown in Fig. 2, the simulation process can be divided into three parts: pre-processing, solving, and post-processing. In the pre-processing stage, some input data is required, such as the material parameters, FEM mesh, and number of incremental steps N. Moreover, the PeriFEM mesh needs to be generated based on the FEM mesh. In the solving stage, the simulation is executed progressively at each incremental step. Each incremental step may contain several iterations, i.e., Eq. (25) may be solved several times until the convergence criterion is satisfied (the convergence criterion will be addressed in the next subsection). Once the results converge, the effective damage is calculated to reveal the cracks. Then, the next incremental step is executed. The simulation ends when all incremental steps are completed.

4.2 Convergence Criterion

This subsection focuses on the convergence criterion at a given incremental step. We introduce two criteria: a bond-based criterion and an equilibrium-based criterion.

Bond-based criterion During the numerical implementation, the definition of bonds is associated with the quadrature points in peridynamic elements, as shown in Fig. 3. For more details on bonds, readers can refer to [31]. This convergence criterion is based on the broken bond information. More specifically, for a given incremental step, if there are no new broken bonds, then after Eq. (25) is solved, we assert that the results converge at this step.

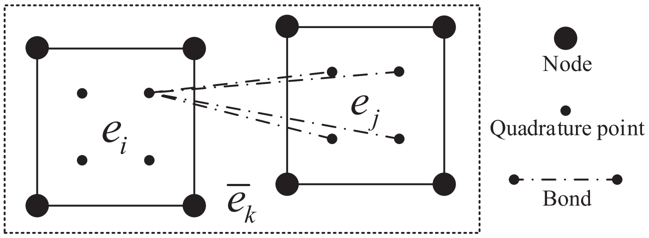


Figure 3: Schematic of bonds in a peridynamic element

Equilibrium-based criterion The equilibrium-based criterion is the default criterion in ABAQUS, which is based on the equilibrium between external and internal forces [42]. For the *j*th iteration in the *i*th incremental step, we can obtain the global nodal displacement vector denoted by $\boldsymbol{d}^{i,j}$ by solving Eq. (25). Then, the status μ of each bond can be updated according to $\boldsymbol{d}^{i,j}$. Based on $\boldsymbol{d}^{i,j}$ and μ, we can calculate the nodal force (i.e., internal force) vector denoted by $\boldsymbol{Q}^{i,j}$, and then define the residual force vector for this iteration as

$$\boldsymbol{R}^{i,j} = \boldsymbol{F}^{i} - \boldsymbol{Q}^{i,j}, \tag{31}$$

where $\boldsymbol{F}^{i}$ is the external force vector at this incremental step. We assert that the results converge at this step if $\left\|\boldsymbol{R}^{i,j}\right\|$ is smaller than a given tolerance.

5 Implementation of PeriFEM in FEM Software

5.1 Implementation of PeriFEM in ABAQUS

In ABAQUS, extracting broken-bond information is not convenient for users; thus, the equilibrium-based convergence criterion (which is the default convergence criterion in ABAQUS) is used. In other words, the incremental steps and iterations in each step are completely controlled by ABAQUS.

Fig. 4 shows the relation between the subroutines and ABAQUS. At the beginning of each incremental step, the calculation model data are read from the input file (.inp), including the material data (PROPS), such as the critical stretch and micro-modulus, the geometry data (COORDS), such as the node coordinates, element nodes, and element numbers, and user-defined element information, such as peridynamic element nodes and element numbers. At each iteration, the element stiffness matrices (AMATRX), right-hand-side vectors (RHS), and state variables (SVARS), which are used to store the broken-bond information during internal computations, are calculated and updated in UEL. At the end of the incremental step, the updated variable information is transmitted to the ABAQUS main program by the UEL subroutine interface. Then, in the UMAT subroutine, the broken-bond and damage information is stored in the state variable (STATEV) through the transfer of global variables. Finally, the results, including the displacement fields and damage fields, can be displayed in the contours of ABAQUS results.

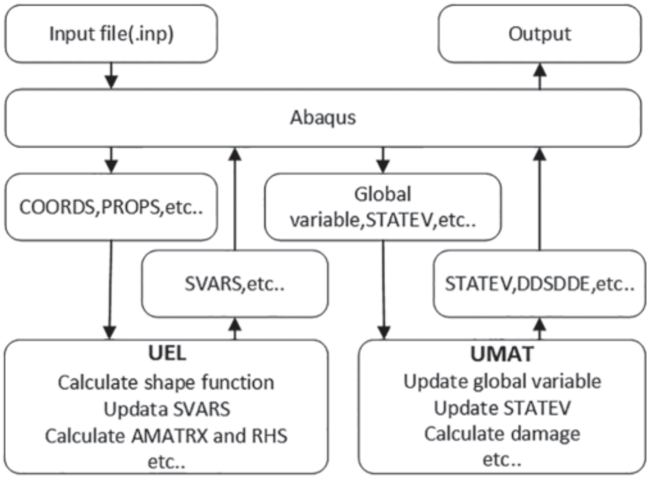


Figure 4: Implementation details of PeriFEM in ABAQUS using UEL and UMAT

In addition, although the stiffness matrix in ABAQUS is obtained entirely based on the peridynamic elements (a user-defined element type), we also need a set of local elements (the native ABAQUS elements, also known as the background elements) for the following reasons: (1) the peridynamic elements are generated from local elements, as detailed in Section 3.1, and (2) the post-processor of ABAQUS does not support user-defined elements. Therefore, to enable the visualization of the simulation results in ABAQUS, native ABAQUS elements are essential. The process of visualization can be summarized as follows. First, the broken-bond information (SVARS) is obtained in UEL. Then, SVARS is passed into UMAT via the global variable (a user-defined variable), based on which the damage information (STATEV) can be obtained according to Eq. (7). Finally, the damage can be visualized through the local elements.

5.2 Implementation of PeriFEM in ANSYS

In ANSYS, APDL allows to extract the broken-bond information from UserElem conveniently, and thus the bond-based convergence criterion can be used. In other words, the incremental steps and iterations in each step are completely controlled by APDL (i.e., controlled by the user).

Fig. 5 shows the relation between the subroutines and ANSYS. At the beginning of each incremental step, the calculation model data are read from the input file (.cdb), including the material data, such as the critical stretch and micro-modulus, the geometry data (ELEM, node), such as node coordinates (node), element nodes, and element numbers (ELEM), and the nodes and element numbers in the user-defined peridynamic elements based on UPFs. At each iteration, the element stiffness matrices (estiff) and element broken-bond information (stored in Rsltvar) based on peridynamic elements are computed and updated in UserElem. At the end of the incremental step, the broken-bond information is transmitted to the ANSYS main program. Finally, the damage is evaluated based on the broken-bond information in the main program and then displayed directly through the GUI interface.

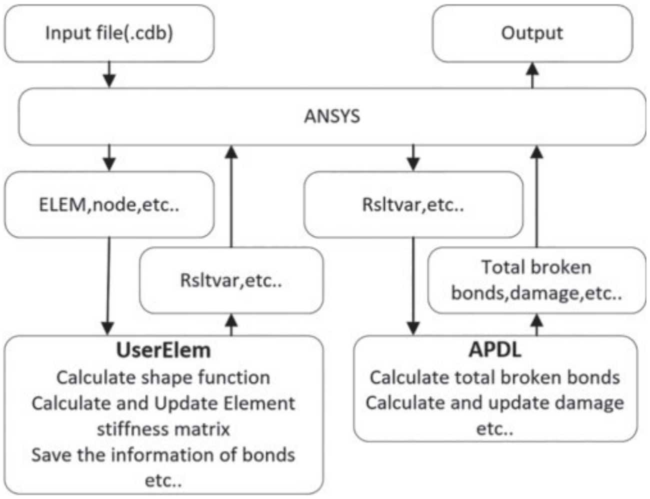

Input file(.cdb)
Output
ANSYS
ELEM,node,etc..
Rsltvar,etc..
Rsltvar,etc..
Total broken bonds,damage,etc..
UserElem
Calculate shape function
Calculate and Update Element stiffness matrix
Save the information of bonds
etc..
APDL
Calculate total broken bonds
Calculate and update damage
etc..

Figure 5: Implementation details of PeriFEM in ANSYS using APDL and UserElem

In addition, we also need both peridynamic elements and local elements in ANSYS, as stated in the previous subsection. The only difference is in how the damage information is obtained. The process of visualization in ANSYS can be summarized as follows. First, the broken-bond information (Rsltvar) is obtained via the UserElem subroutine. Then, RsltVar is extracted from UserElem by APDL, based on which the damage information for each local element can be calculated according to Eq. (7). Finally, the damage contour can be displayed in the *postprocess* module of ANSYS via the local elements.

Remark 4. Even though PeriFEM allows both continuous and discrete local elements [31], only continuous local elements are available in ABAQUS and ANSYS. Thus, only continuous local elements were used in the present study.

6 Numerical Examples

6.1 2D Tests for the Verification and Comparison of ANSYS and ABAQUS

The purpose of the tests presented in this subsection is twofold. On the one hand, we first wish to verify the correct functioning of the code for implementing PeriFEM in ANSYS and ABAQUS. On the other hand, as different convergence criteria were adopted in ANSYS and ABAQUS, the convergence performance, computational cost, and predicted crack patterns need to be compared between ANSYS and ABAQUS. To this end, 2D benchmark examples are carried out using both ANSYS and ABAQUS.

6.1.1 Single-Edge-Notched Plate under Tension

First, we consider a tension test for a single-edge-notched plate. The geometry and boundary conditions are shown in Fig. 6. Young's modulus is set to $E = 200$ GPa, Poisson's ratio is $\nu = 1/3$, and the critical stretch is $s_{crit} = 0.02$. The plate is discretized with quadrilateral local elements and the mesh size is $h = 5$ mm. The horizon is chosen as $\delta = 3h$ and 9860 peridynamic elements are generated. The boundary conditions are divided into 100 incremental steps.

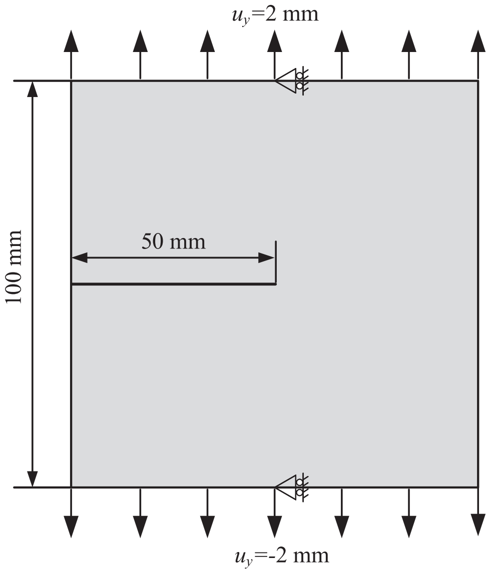
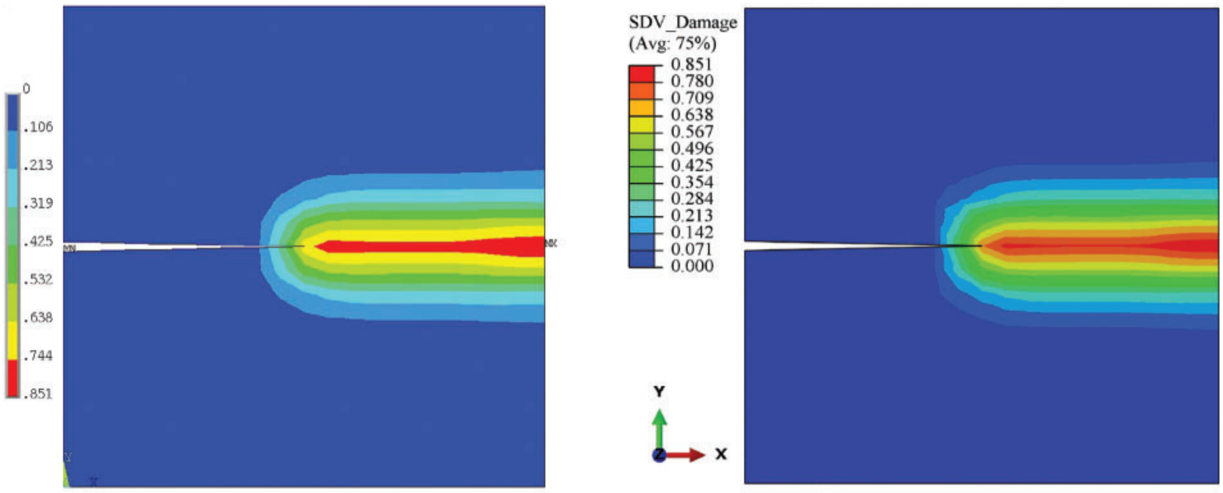


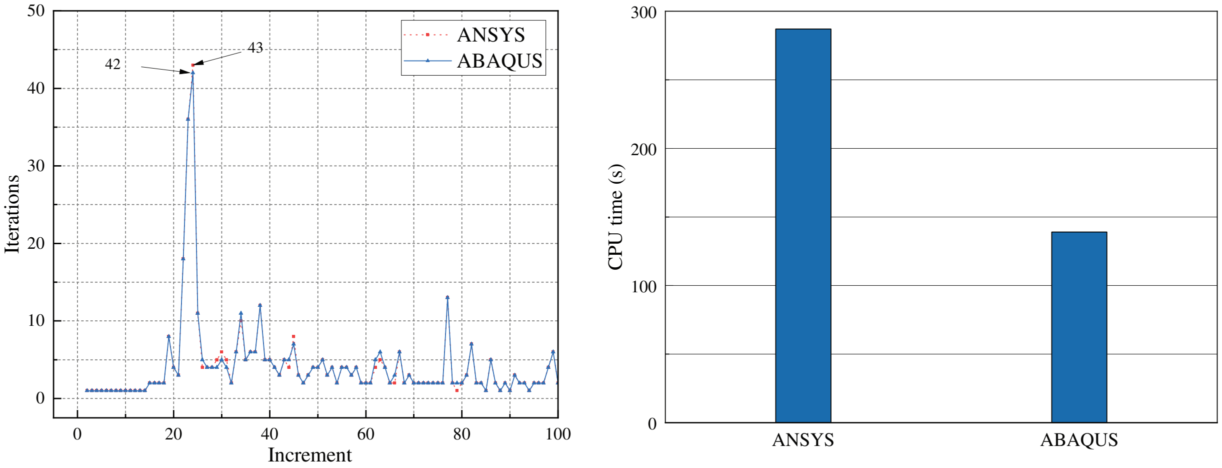
Figure 6: Geometry and boundary conditions of a single-edge-notched plate

It is known that the crack will initiate from the notch tip and propagate horizontally to the right for this test. As shown in Fig. 7, the effective damage contours, which reveal the crack, predicted by ANSYS and ABAQUS are correct. This verifies the correct integration of the PeriFEM code into ANSYS and ABAQUS. Moreover, Fig. 8a shows the iteration numbers in each incremental step of both ANSYS and ABAQUS. It can be seen that the iteration numbers almost coincided with each other in this test, even though different convergence criteria are adopted. Fig. 8b displays the total computational costs of both ANSYS and ABAQUS. In this example, the total number of iterations of ANSYS and ABAQUS is similar, but the overall CPU times are considerably different. Notably, several factors may affect the overall CPU time. For instance, the way in which post-processing is implemented may have had a considerable impact. In ABAQUS, post-processing is implemented through the UMAT subroutine, whereas it is implemented through in-house APDL code in ANSYS.



(a) (b)

Figure 7: Contours of the effective damage for a single-edge-notched plate computed by (a) ANSYS and (b) ABAQUS



(a) (b)

Figure 8: Comparison of the number of iterations in each incremental step and total CPU time between ANSYS and ABAQUS

6.1.2 L-Shaped Panel

We now consider the case of an L-shaped panel. The geometry and boundary conditions are shown in Fig. 9. Young's modulus is set to $E = 25.85$ GPa, Poisson's ratio is $\nu = 1/3$, and the critical stretch is $s_{crit} = 0.0006$. The panel is discretized with quadrilateral local elements, and the mesh size is $h \approx 3.6$ mm. The horizon is chosen as $\delta = 3h$ and 909262 peridynamic elements are generated. The boundary conditions are divided into 140 incremental steps.

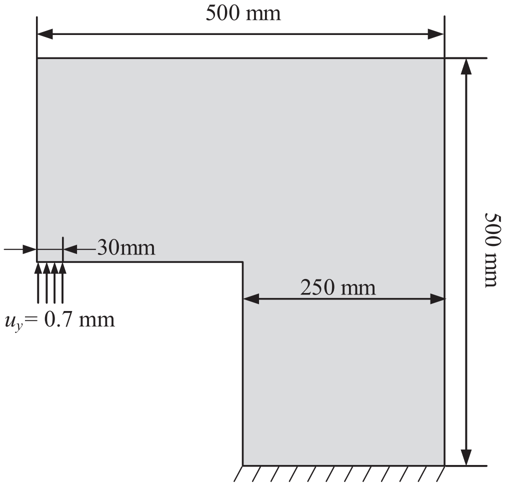
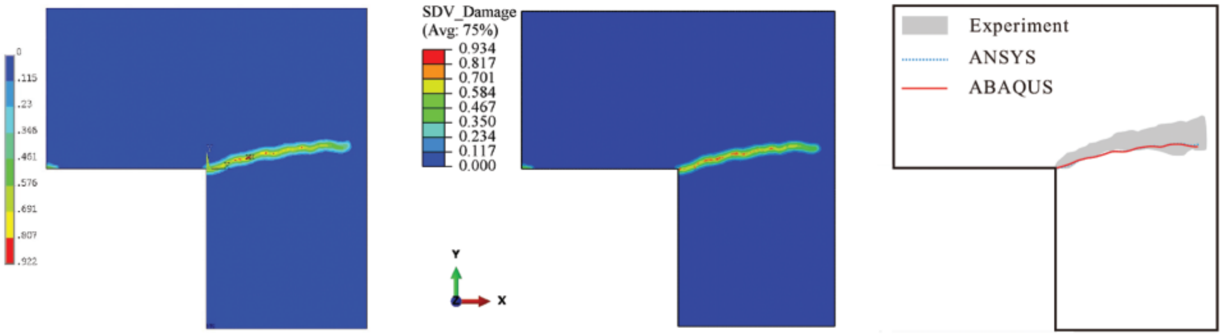


Figure 9: Geometry and boundary conditions of an L-shaped panel

Figs. 10a and 10b display the effective damage zones predicted by ANSYS and ABAQUS, respectively, and Fig. 10c compares the predicted crack paths with the experimental failure zone reported in [43]. The simulated results obtained using both ANSYS and ABAQUS are in good agreement with the experimental data, which further verifies the developed code. Moreover, Fig. 11 shows the number of iterations in each incremental step for ANSYS and ABAQUS. The similarity between the two curves indicates that the simulated results by ANSYS and ABAQUS are in line with each other.



(a) (b) (c)

Figure 10: Contours of the effective damage for an L-shaped panel computed by (a) ANSYS and (b) ABAQUS, as well as (c) a comparison with experimental results

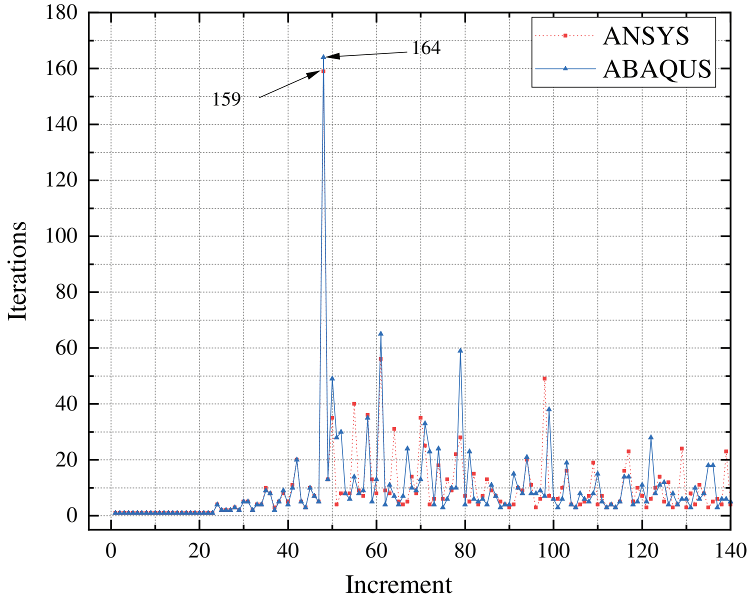


Figure 11: Comparison of the number of iterations in each incremental step between ANSYS and ABAQUS

6.1.3 Double-Edge-Notched Plate under Tension and Shear

In this test, we investigate the mixed-mode fracture of a double-edge-notched plate. The geometry and boundary conditions are shown in Fig. 12. Young's modulus is set to $E = 30$ GPa, Poisson's ratio is $\nu = 1/3$, and the critical stretch is $s_{crit} = 0.02$. The plate is discretized with quadrilateral local elements, and the mesh size is $h \approx 1.25$ mm. The horizon is set as $\delta = 3h$, and 1019120 peridynamic elements are generated. The boundary conditions are divided into 80 incremental steps.

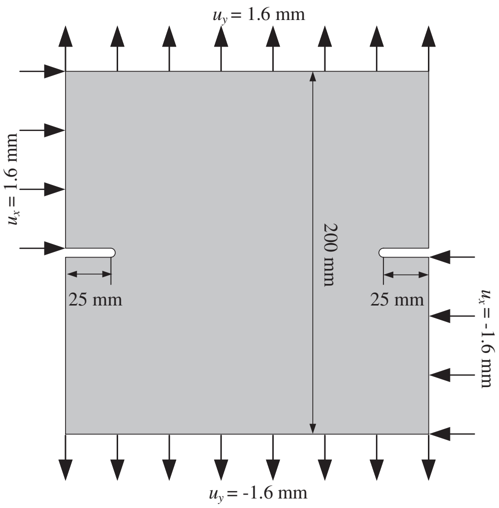


Figure 12: Geometry and boundary condition of a double-edge-notched plate

The predicted effective damage contours and a comparison with the experimental results reported in [44] are shown in Fig. 13. It can be seen that the predicted crack paths are in good agreement with the experimental results. However, one may note that the simulated results obtained with ABAQUS presented in the figure are only up to the 20th incremental step, but not for the 80th incremental step. This is because from the 21st increment onward, ABAQUS encounters a convergence issue, as shown in Fig. 14. Figs. 14a and 14b display the number of broken bonds and the largest residual force during the iteration, respectively. Although the number of broken bonds no longer increases, the largest residual force still does not satisfy the convergence condition and falls into a regular oscillation, which means that the iteration will never converge.

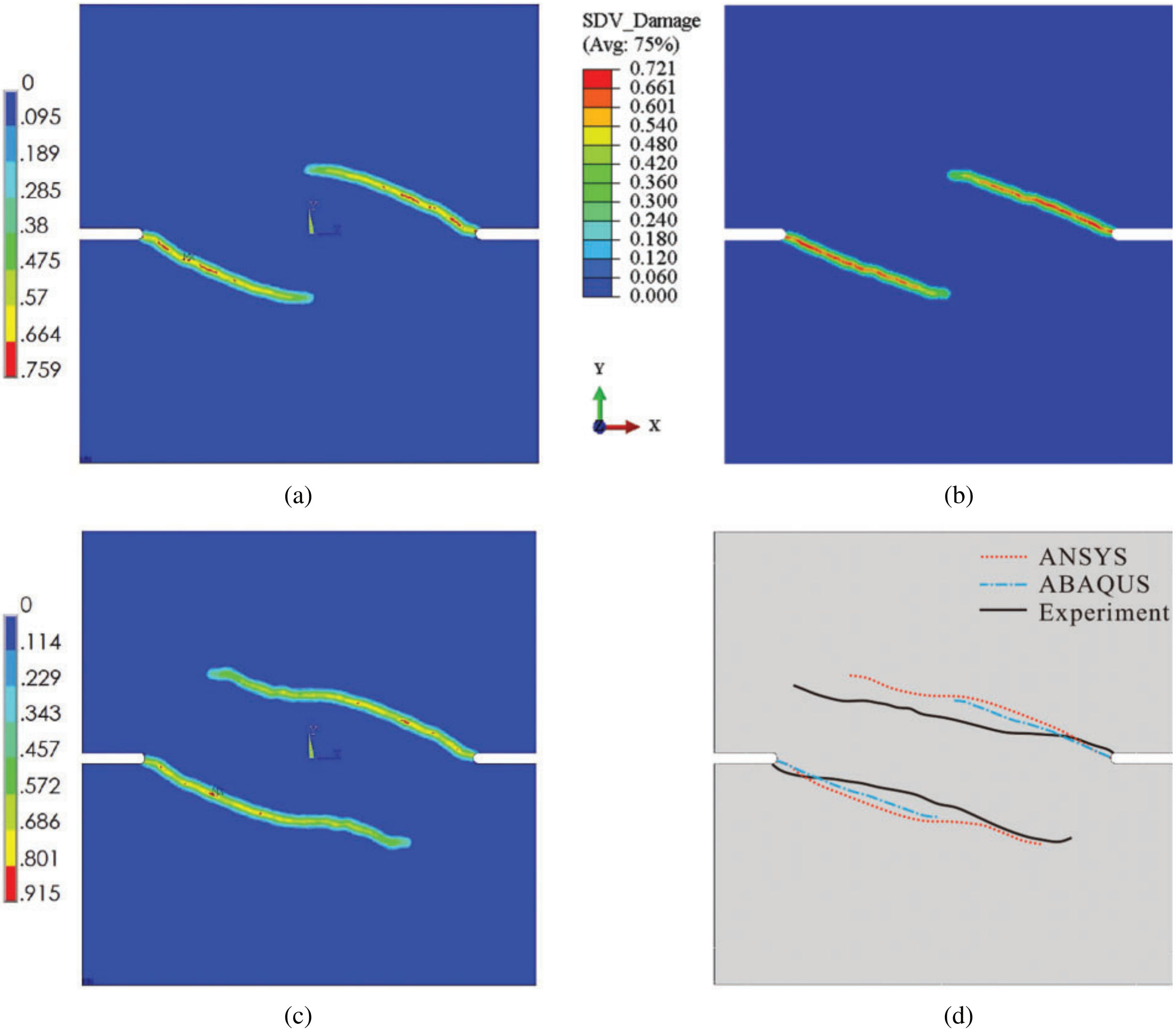
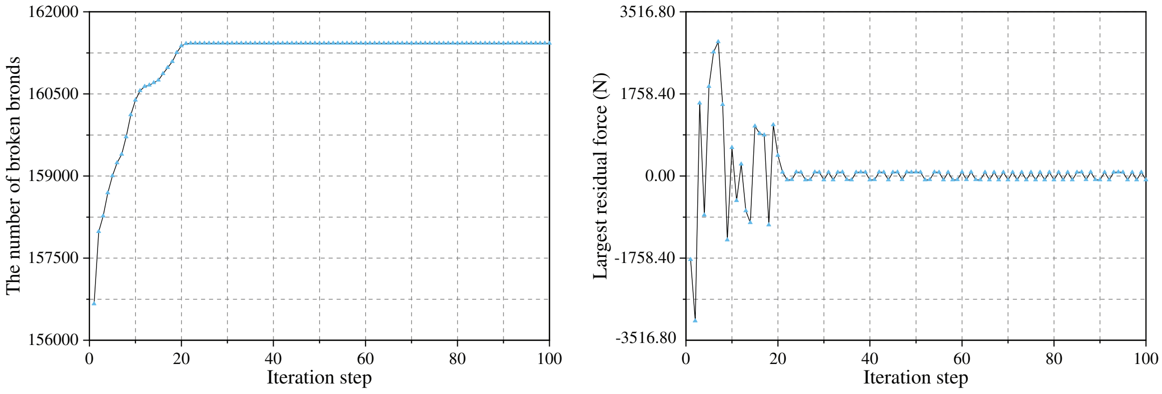


Figure 13: Contours of the effective damage for a double-edged plate obtained with (a) ANSYS at 20th incremental step, (b) ABAQUS at 20th incremental step, and (c) ANSYS at 80th incremental step, as well as (d) a comparison with experimental results



(a) (b)

Figure 14: Iteration information of ABAQUS at the 21st incremental step: (a) number of broken bonds and (b) largest residual force

Remark 5. The above non-convergence phenomenon is not a problem of the code but may stem from the constitutive model and the specimen. It was reported in [45] that the discontinuity of the constitutive model can result in convergence issues.

6.2 3D Tests

We now investigate 3D examples using ANSYS and ABAQUS to verify the ability of the proposed method to deal with complex problems. In all tests, hexahedral local elements are used.

6.2.1 Three-Point Bending Test on a Slanted Notched Beam

This example shows a three-point bending test of a beam with a slanted notch. More specifically, the vertical notch had an inclination of $\theta = 45^\circ$ with respect to the longitudinal mid-plane section of the beam. We consider the same geometry and boundary conditions as those reported in [46], which are presented in Fig. 15.

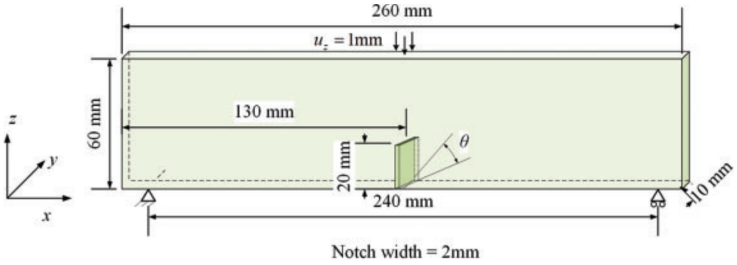


Figure 15: Geometry and boundary condition for the slanted notched beam

Young's modulus is set to $E = 2.8$ GPa, Poisson's ratio is $\nu = 1/4$, and the critical stretch is $s_{crit} = 0.00235$. The beam is discretized with hexahedral local elements and the mesh size is $h \approx 2.0$ mm. The horizon is set as $\delta = 3h$, and 1618634 peridynamic elements are generated. The prescribed displacement is equally divided into 100 incremental steps.

The ABAQUS software is used for this test. The predicted effective damage contours are shown in Fig. 16. Initially, a mode III fracture is predominant. Therefore, it can be seen that the crack face first propagates along the direction of the slanted notch, which stems from the stress concentration around the notch caused by the asymmetrical bending of the beam. Then, as the crack propagated, the mid-plane section acts as an attractor of the crack surface owing to the coincidence between the loading and itself. Thus, a mode I fracture gradually became predominant. It can be seen that the crack surface twists gradually in space as the prescribed displacement increased. Finally, it aligns with the mid-plane section. We can conclude that the torsion of the crack surface was successfully captured in the simulation, although this torsion was not very obvious. This is due to a limitation in computational resources; the mesh is not sufficiently fine. Nonetheless, the overall results show that the proposed technique can be successfully applied to 3D problems.

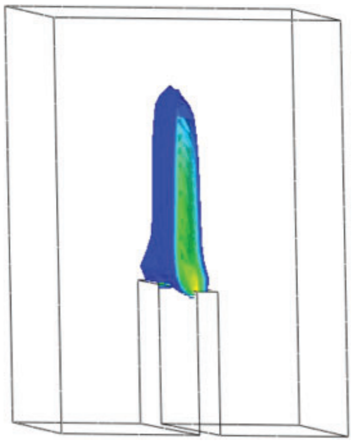
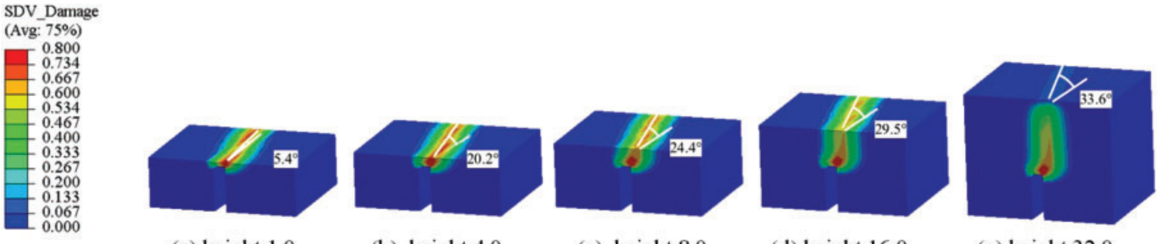


Figure 16: Crack pattern and damage profile for the slanted notched beam

To better display the results, prospective views of the damage profile at various heights above the notch are shown in Fig. 17. We used two reference (white) lines; one for marking the main damage zone and the other for marking the direction of the slanted notch to better show the rotation angle of the main damage zone. The difference between the revolving angles at different heights reveals the rotation of the damage field. The damage starts from the slanted notch, and then twists spatially and develops along the vertical direction. First, the damage field revolving angle increased rapidly. Then, it increases slowly. Finally, the top view of the damage profile is nearly along the mid-plane section of the geometric model.



(a) height 1.0 (b) height 4.0 (c) height 8.0 (d) height 16.0 (e) height 32.0

Figure 17: Predicted damage profiles at different heights above the notch

6.2.2 Prismatic Skew-Notched Beam under Torsion

Finally, a prismatic skew-notched concrete beam under torsion is simulated using ANSYS. This example was also tested by Brokenshire [47] and reported in detail in [48]. The concrete specimen used in those experiments included two steel frames at both ends. However, due to Saint Venant's principle, these steel frames can be simplified as displacement constraints and have minimal influence on the stress state near the notch. The geometry and boundary conditions of such a simplified model are shown in Fig. 18.

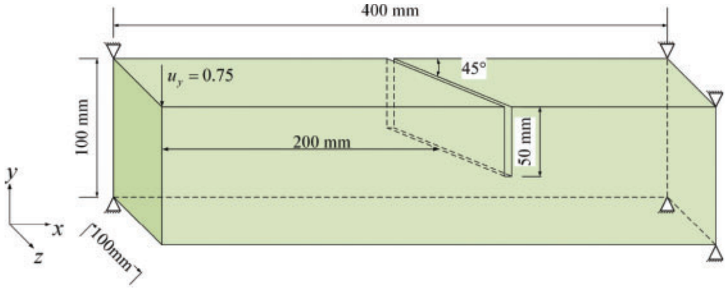


Figure 18: Geometry and boundary conditions of the prismatic skew-notched beam

Young's modulus is set to $E = 35$ GPa, Poisson's ratio is $\nu = 1/4$, and the critical stretch is $s_{crit} = 0.0006316$. The beam is discretized with hexahedral local elements and the mesh size is $h \approx 5$ mm. The horizon is set as $\delta = 3h$ and 3886626 peridynamic elements are generated. The prescribed displacement is equally divided into 50 incremental steps.

Fig. 19 shows the displacement contour and three different views of the damage iso-surface at the last incremental step. Fig. 19a shows the displacement contour of the beam, and Figs. 19b–19d show top, slanted, and side views of the damage isosurface, respectively. A skew-symmetrical crack surface with a complex twisting pattern can be clearly observed from these three different views. Moreover, distinct antisymmetric crack kinking can be observed along the straight crack front of the inclined initial crack. This is caused by the combination of the mode I and mode III loading conditions. Meanwhile, from Fig. 19c, we can observe that crack kinking becomes more pronounced further away from the notch. This is because when the crack initiates, mode I fracture is predominant. However, as the crack propagates, mode III fracture plays a more important role, which in turn causes crack kinking to become more pronounced.

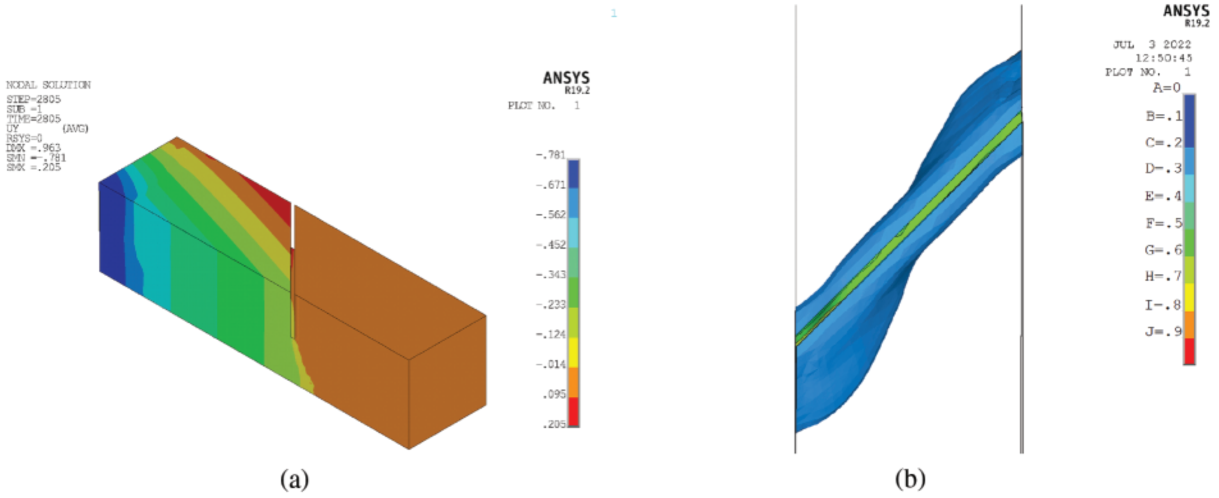
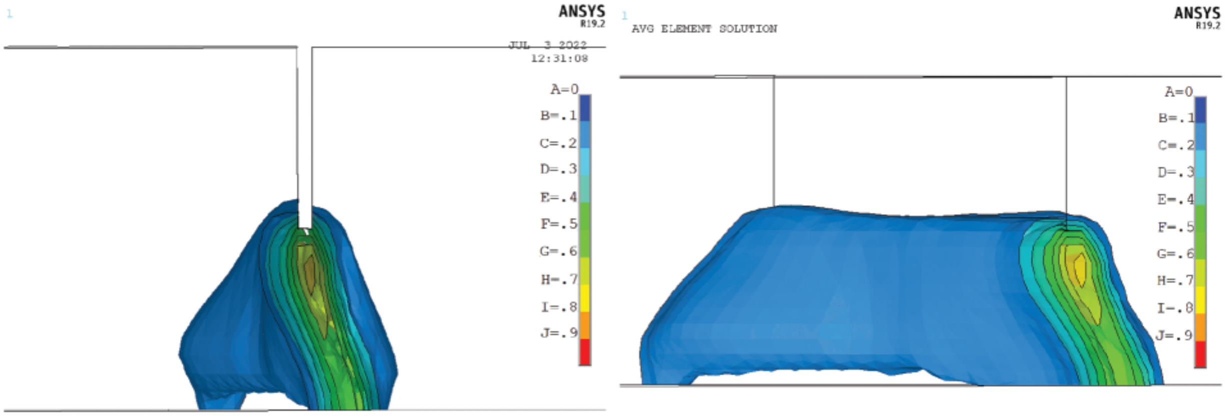


Figure 19: (Continued)

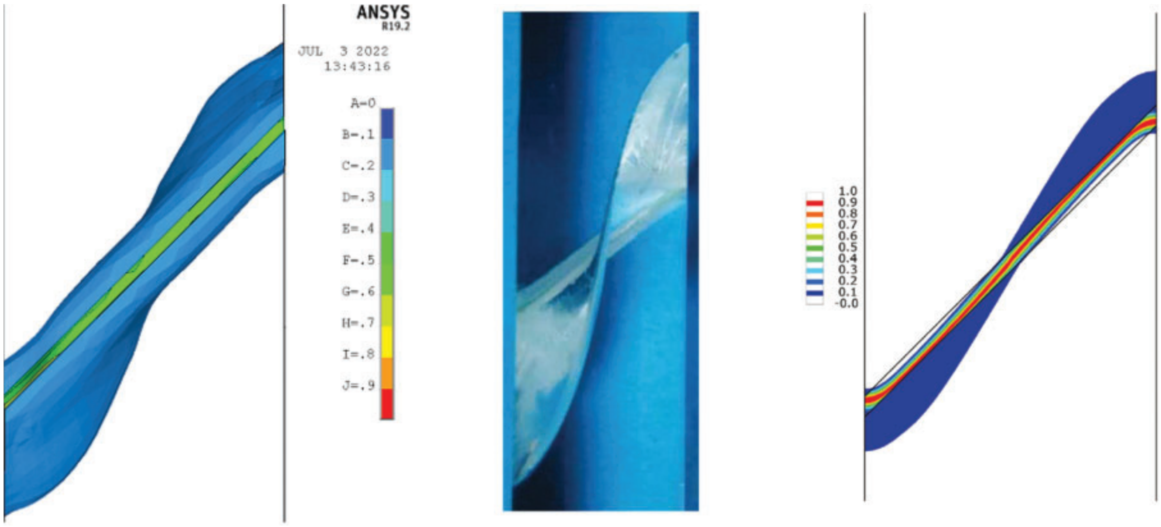


(c)

(d)

Figure 19: (a) Displacement contour and (b) top view, (c) slanted view, and (d) side view of the damage isosurface of the concrete beam at the last incremental step

Fig. 20 shows three top views of the damage isosurface obtained using three different methods. Fig. 20a shows the results obtained by ANSYS using PeriFEM, Fig. 20b shows the results obtained from an experiment on PMMA [49], and Fig. 20c shows the results obtained by ABAQUS using the phase-field method [46]. The crack surface shown in Fig. 20a is similar to those in Figs. 20b and 20c. Moreover, the above results are consistent with those obtained from an isotropic damage model in the context of stabilized mixed finite elements [50], which demonstrates the feasibility of using PeriFEM in ANSYS to simulate 3D problems. Note that the maximum vertical displacement in ANSYS is 0.75 mm, which is half the 1.5 mm value used in the experimental test and ABAQUS simulation. According to the load–displacement curve in [46], the beam completely fractures when the vertical displacement is 0.75 mm.



(a)

(b)

(d)

Figure 20: Top view of the damage surface obtained using different methods: (a) ANSYS with PeriFEM; (b) PMMA test in [49]; (c) ABAQUS and the phase-field method [46]

7 Conclusions

In this study, we proposed a unified implementation strategy for peridynamics in commercial FEM software by leveraging available APIs and using the peridynamics-based finite element method. As a demonstration of this strategy, PeriFEM was implemented in ANSYS using UPFs and ABAQUS using UEL. Various 2D and 3D numerical tests demonstrated the effectiveness of the proposed strategy for brittle fracture modeling. Based on the proposed implementation details and the simulation results obtained in this study, we can conclude the following:

- It is convenient to integrate PeriFEM into commercial FEM software through its API via the proposed unified implementation strategy.
- The brittle fracture of materials under complex loads can be simulated by the proposed strategy using commercial FEM software.

Therefore, the present work facilitates the application of peridynamics in engineering practice. Future work will focus on implementing coupled local/non-local models in software to improve computational efficiency and reduce boundary effects.

Funding Statement: The authors gratefully acknowledge the financial support received from the National Natural Science Foundation of China (12272082, 11872016) and the National Key Laboratory of Shock Wave and Detonation Physics (JCKYS2021212003).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Hey, A. J., Tansley, S., Tolle, K. M. (2009). *The fourth paradigm: Data-intensive scientific discovery*, vol. 1. WA: Microsoft Research Redmond.
2. Logan, D. L. (2016). *A first course in the finite element method*. Boston MA, USA: Cengage Learning.
3. Belytschko, T., Liu, W. K., Moran, B., Elkhodary, K. (2014). *Nonlinear finite elements for continua and structures*. West Sussex, UK: John Wiley & Sons.
4. Fang, X. J., Jin, F. (2007). Extended finite element method based on ABAQUS. *Engineering Mechanics, 24(7),* 6–10.
5. Li, W., Nguyen-Thanh, N., Zhou, K. (2020). Phase-field modeling of brittle fracture in a 3D polycrystalline material via an adaptive isogeometric-meshfree approach. *International Journal for Numerical Methods in Engineering, 121(22),* 5042–5065. https://doi.org/10.1002/nme.6509
6. Nguyen-Thanh, N., Li, W., Huang, J., Zhou, K. (2020). Adaptive higher-order phase-field modeling of anisotropic brittle fracture in 3D polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering, 372(2),* 113434. https://doi.org/10.1016/j.cma.2020.113434
7. Lindgaard, E., Bak, B., Glud, J., Sjølund, J., Christensen, E. (2017). A user programmed cohesive zone finite element for ANSYS Mechanical. *Engineering Fracture Mechanics, 180(2),* 229–239. https://doi.org/10.1016/j.engfracmech.2017.05.026
8. Msekh, M. A., Sargado, J. M., Jamshidian, M., Areias, P. M., Rabczuk, T. (2015). Abaqus implementation of phase-field model for brittle fracture. *Computational Materials Science, 96(8),* 472–484. https://doi.org/10.1016/j.commatsci.2014.05.071
9. Molnár, G., Gravouil, A. (2017). 2D and 3D Abaqus implementation of a robust staggered phase-field solution for modeling brittle fracture. *Finite Elements in Analysis and Design, 130(582–593),* 27–38. https://doi.org/10.1016/j.finel.2017.03.002

10. Wu, J. Y., Huang, Y. (2020). Comprehensive implementations of phase-field damage models in Abaqus. *Theoretical and Applied Fracture Mechanics, 106(4),* 102440. https://doi.org/10.1016/j.tafmec.2019.102440
11. Silling, S. A. (2000). Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids, 48(1),* 175–209. https://doi.org/10.1016/S0022-5096(99)00029-0
12. Silling, S. A., Epton, M., Weckner, O., Xu, J., Askari, E. (2007). Peridynamic states and constitutive modeling. *Journal of Elasticity, 88(2),* 151–184. https://doi.org/10.1007/s10659-007-9125-1
13. Li, J., Li, S., Lai, X., Liu, L. (2022). Peridynamic stress is the static first Piola–Kirchhoff virial stress. *International Journal of Solids and Structures, 241(1–2),* 111478. https://doi.org/10.1016/j.ijsolstr.2022.111478
14. Zhang, N., Gu, Q., Huang, S., Xue, X., Li, S. (2021). A practical bond-based peridynamic modeling of reinforced concrete structures. *Engineering Structures, 244(1),* 112748. https://doi.org/10.1016/j.engstruct.2021.112748
15. Chen, Z., Chu, X. (2021). Peridynamic modeling and simulation of fracture process in fiber-reinforced concrete. *Computer Modeling in Engineering & Sciences, 127(1),* 241–272. https://doi.org/10.32604/cmes.2021.015120
16. Han, J., Li, S., Yu, H., Li, J., Zhang, A. M. (2022). On nonlocal cohesive continuum mechanics and cohesive peridynamic modeling (CPDM) of inelastic fracture. *Journal of the Mechanics and Physics of Solids, 164,* 104894. https://doi.org/10.1016/j.jmps.2022.104894
17. Chen, X., Gunzburger, M. (2011). Continuous and discontinuous finite element methods for a peridynamics model of mechanics. *Computer Methods in Applied Mechanics and Engineering, 200(9–12),* 1237–1250. https://doi.org/10.1016/j.cma.2010.10.014
18. Azdoud, Y., Han, F., Lubineau, G. (2014). The morphing method as a flexible tool for adaptive local/non-local simulation of static fracture. *Computational Mechanics, 54(3),* 711–722. https://doi.org/10.1007/s00466-014-1023-3
19. Ren, B., Wu, C., Askari, E. (2017). A 3D discontinuous Galerkin finite element method with the bond-based peridynamics model for dynamic brittle failure analysis. *International Journal of Impact Engineering, 99(5),* 14–25. https://doi.org/10.1016/j.ijimpeng.2016.09.003
20. Silling, S. A. (2003). Dynamic fracture modeling with a meshfree peridynamic code. In: *Computational fluid and solid mechanics 2003*, pp. 641–644. Massachusetts, USA: Elsevier.
21. Silling, S. A., Askari, E. (2005). A meshfree method based on the peridynamic model of solid mechanics. *Computers & Structures, 83(17–18),* 1526–1535. https://doi.org/10.1016/j.compstruc.2004.11.026
22. Seleson, P., Littlewood, D. J. (2016). Convergence studies in meshfree peridynamic simulations. *Computers & Mathematics with Applications, 71(11),* 2432–2448. https://doi.org/10.1016/j.camwa.2015.12.021
23. Diyaroglu, C., Madenci, E., Phan, N. (2019). Peridynamic homogenization of microstructures with orthotropic constituents in a finite element framework. *Composite Structures, 227,* 111334. https://doi.org/10.1016/j.compstruct.2019.111334
24. Huang, X., Bie, Z., Wang, L., Jin, Y., Liu, X. et al. (2019). Finite element method of bond-based peridynamics and its ABAQUS implementation. *Engineering Fracture Mechanics, 206(1),* 408–426. https://doi.org/10.1016/j.engfracmech.2018.11.048
25. Bie, Y., Liu, Z., Yang, H., Cui, X. (2020). Abaqus implementation of dual peridynamics for brittle fracture. *Computer Methods in Applied Mechanics and Engineering, 372,* 113398. https://doi.org/10.1016/j.cma.2020.113398
26. Anicode, S. V. K., Madenci, E. (2022). Bond-and state-based peridynamic analysis in a commercial finite element framework with native elements. *Computer Methods in Applied Mechanics and Engineering, 398(12),* 115208. https://doi.org/10.1016/j.cma.2022.115208
27. Macek, R. W., Silling, S. A. (2007). Peridynamics via finite element analysis. *Finite Elements in Analysis and Design, 43(15),* 1169–1178. https://doi.org/10.1016/j.finel.2007.08.012

28. Ren, B., Wu, C., Seleson, P., Zeng, D., Nishi, M. et al. (2022). An FEM-based peridynamic model for failure analysis of unidirectional fiber-reinforced laminates. *Journal of Peridynamics and Nonlocal Modeling, 4(1),* 139–158. https://doi.org/10.1007/s42102-021-00063-0
29. https://www.lstc-cmmg.org/peri-dynamics
30. Han, F., Li, Z. (2022). A peridynamics-based finite element method (PeriFEM) for quasi-static fracture analysis. *Acta Mechanica Solida Sinica, 35(3),* 446–460. https://doi.org/10.1007/s10338-021-00307-y
31. Li, Z., Han, F. (2023). The peridynamics-based finite element method (PeriFEM) with adaptive continuous/discrete element implementation for fracture simulation. *Engineering Analysis with Boundary Elements, 146(5),* 56–65. https://doi.org/10.1016/j.enganabound.2022.09.033
32. Han, F., Li, Z., Zhang, J., Liu, Z., Yao, C. et al. (2022). ABAQUS and ANSYS implementations of peridynamics-based finite element method (PeriFEM) for brittle fracture. https://www.researchsquare.com/article/rs-1888499/v1.pdf
33. Madenci, E., Oterkus, E. (2014). *Peridynamic theory and its applications*. New York, NY: Springer.
34. Bobaru, F., Yang, M., Alves, L. F., Silling, S. A., Askari, E. et al. (2009). Convergence, adaptive refinement, and scaling in 1D peridynamics. *International Journal for Numerical Methods in Engineering, 77(6),* 852–877. https://doi.org/10.1002/nme.2439
35. Lubineau, G., Azdoud, Y., Han, F., Rey, C., Askari, A. (2012). A morphing strategy to couple nonlocal to local continuum mechanics. *Journal of the Mechanics and Physics of Solids, 60(6),* 1088–1102. https://doi.org/10.1016/j.jmps.2012.02.009
36. Wang, Y., Han, F., Lubineau, G. (2021). Strength-induced peridynamic modeling and simulation of fractures in brittle materials. *Computer Methods in Applied Mechanics and Engineering, 374(1),* 113558. https://doi.org/10.1016/j.cma.2020.113558
37. Yu, H., Li, S. (2020). On energy release rates in peridynamics. *Journal of the Mechanics and Physics of Solids, 142(3),* 104024. https://doi.org/10.1016/j.jmps.2020.104024
38. Han, F., Lubineau, G., Azdoud, Y., Askari, A. (2016). A morphing approach to couple state-based peridynamics with classical continuum mechanics. *Computer Methods in Applied Mechanics and Engineering, 301,* 336–358. https://doi.org/10.1016/j.cma.2015.12.024
39. Ren, H., Zhuang, X., Cai, Y., Rabczuk, T. (2016). Dual-horizon peridynamics. *International Journal for Numerical Methods in Engineering, 108(12),* 1451–1476. https://doi.org/10.1002/nme.5257
40. Pasetto, M., Shen, Z., D'Elia, M., Tian, X., Trask, N. et al. (2022). Efficient optimization-based quadrature for variational discretization of nonlocal problems. *Computer Methods in Applied Mechanics and Engineering, 396(6),* 115104. https://doi.org/10.1016/j.cma.2022.115104
41. Morales, R. C., Baek, J., Sharp, D., Aderounmu, A., Wei, H. et al. (2022). Mode-II fracture response of pmma under dynamic loading conditions. *Journal of Dynamic Behavior of Materials, 8(1),* 104–121. https://doi.org/10.1007/s40870-021-00320-9
42. Dassault Systëmes (2012). *Abaqus analysis user's manual*. http://130.149.89.49:2080/v6.12/books/usb/default.htm
43. Winkler, B. J. (2001). *Traglastuntersuchungen von unbewehrten und bewehrten Betonstrukturen auf der Grundlage eines objektiven Werkstoffgesetzes für Beton (Ph.D. Thesis)*. Universität Innsbruck.
44. Nooru-Mohamed, M., Schlangen, E., van Mier, J. G. (1993). Experimental and numerical study on the behavior of concrete subjected to biaxial tension and shear. *Advanced Cement Based Materials, 1(1),* 22–37. https://doi.org/10.1016/1065-7355(93)90005-9
45. Ni, T., Zaccariotto, M., Zhu, Q. Z., Galvanetto, U. (2019). Static solution of crack propagation problems in peridynamics. *Computer Methods in Applied Mechanics and Engineering, 346(3),* 126–151. https://doi.org/10.1016/j.cma.2018.11.028

46. Wu, J. Y., Huang, Y., Zhou, H., Nguyen, V. P. (2021). Three-dimensional phase-field modeling of mode I+II/III failure in solids. *Computer Methods in Applied Mechanics and Engineering, 373(48),* 113537. https://doi.org/10.1016/j.cma.2020.113537
47. Brokenshire, D. (1996). *A study of torsion fracture test (Ph.D. Thesis)*. Cardiff University.
48. Jefferson, A. D., Barr, B., Bennett, T., Hee, S. (2004). Three dimensional finite element simulations of fracture tests using the craft concrete model. *Computers and Concrete, 1(3),* 261–284. https://doi.org/10.12989/cac.2004.1.3.261
49. Buchholz F. G., Just V., Richard, H. (2005). Computational simulation and experimental findings of three-dimensional fatigue crack growth in a single-edge notched specimen under torsion loading. *Fatigue & Fracture of Engineering Materials & Structures, 28(1–2),* 127–134. https://doi.org/10.1111/j.1460-2695.2005.00864.x
50. Benedetti, L., Cervera, M., Chiumenti, M. (2017). 3D numerical modelling of twisting cracks under bending and torsion of skew notched beams. *Engineering Fracture Mechanics, 176(3),* 235–256. https://doi.org/10.1016/j.engfracmech.2017.03.025
51. Azdoud, Y., Han, F., Lubineau, G. (2013). A morphing framework to couple non-local and local anisotropic continua. *International Journal of Solids and Structures, 50(9),* 1332–1341. https://doi.org/10.1016/j.ijsolstr.2013.01.016

Appendix

Appendix A. Calculation of $\tilde{c}(\theta,\varphi)$ for anisotropic materials

In 2013, Azdoud et al. [51] derived an expression for $c(\boldsymbol{\xi})$ using the real spherical harmonic expansion technique, where $\tilde{c}(\theta,\varphi)$ in Eq. (4) was defined as

$$\tilde{c}(\theta,\varphi) = a_{00} + \sum_{k=1}^{+\infty}\left[\sum_{m=0}^{k} P_k^m(\cos(\theta))\,(a_{km}\cos(m\varphi) + b_{km}\sin(m\varphi))\right]. \tag{32}$$

Here, θ and φ supplemented with $\|\boldsymbol{\xi}\|$ comprise the spherical coordinates of bond $\boldsymbol{\xi}$, as shown in Fig. 21. a_{00}, a_{km}, and b_{km} are real parameters and P_k^m are Legendre functions. They showed that, for the *orthotropic model*, it holds that

$$\begin{aligned}\tilde{c}(\theta,\varphi) &= a_{00} + a_{20}P_2^0(\cos(\theta)) + a_{22}\cos(2\phi)P_2^2(\cos(\theta))\\ &\quad + a_{40}P_4^0(\cos(\theta)) + a_{42}\cos(2\phi)P_4^2(\cos(\theta)) + a_{44}\cos(4\phi)P_4^4(\cos(\theta)),\end{aligned} \tag{33}$$

whereas for the *transverse isotropic model* (assuming that $\boldsymbol{e}_3$ is the reference direction of an axisymmetric material), it holds that

$$\tilde{c}(\theta,\varphi) = \tilde{c}(\theta) = a_0 + a_2P_2^0(\cos\theta) + a_4P_4^0(\cos\theta). \tag{34}$$

In addition, they gave an expression of the real parameters in Eqs. (33) and (34) for $\hat{c}(\|\boldsymbol{\xi}\|) = 1$. However, they did not explain how to calculate these real parameters; thus, it is still difficult to determine the real parameters for $\hat{c}(\|\boldsymbol{\xi}\|) \neq 1$.

The parameters in Eqs. (33) and (34) can be determined according to the deformation energy equivalence; the key point is to define an equivalent stiffness tensor. Based on the assumption of uniform strain fields, Lubineau et al. derived an effective stiffness tensor in [35] as

$$\boldsymbol{D} = \int_{H_\delta(\boldsymbol{x})} \frac{c(\boldsymbol{\xi})}{2}\boldsymbol{\xi}\otimes\boldsymbol{\xi}\otimes\boldsymbol{\xi}\otimes\boldsymbol{\xi}\,\mathrm{d}V_{x'}. \tag{35}$$

Let us denote the classical stiffness tensor by $\boldsymbol{E}$ and set

$$\boldsymbol{D} = \boldsymbol{E}. \tag{36}$$

Then, we can determine the parameters.

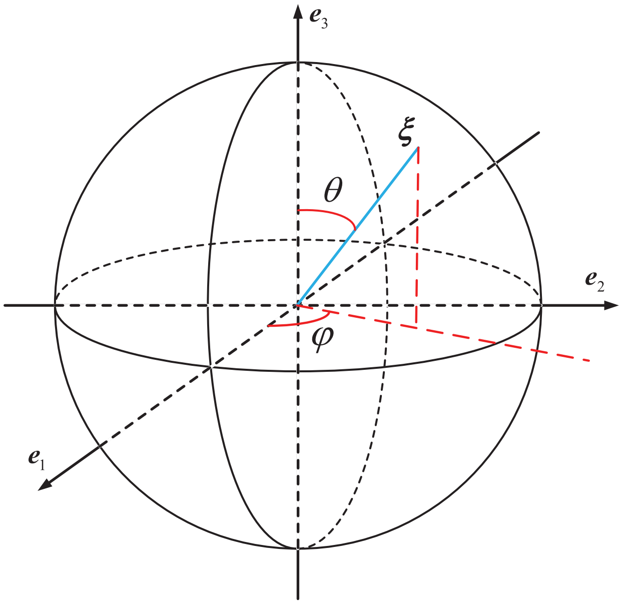


Figure 21: Spherical coordinates for bond $\boldsymbol{\xi}$

Appendix B. Orthotropic model

In the classical orthotropic model, there are nine independent variables in $\boldsymbol{E}$, which are

$$E_{1111}, E_{2222}, E_{3333}, E_{1122}, E_{1133}, E_{2233}, E_{1212}, E_{1313}, E_{2323}. \tag{37}$$

In the peridynamic orthotropic model, there are only six independent variables in $\boldsymbol{D}$ because of the centrosymmetry of the long-range force. These variables are

$$D_{1111}, D_{2222}, D_{3333}, D_{1122} = D_{1212}, D_{1133} = D_{1313}, D_{2233} = D_{2323}. \tag{38}$$

Therefore, Eq. (36) implies that

$$D_{1111} = E_{1111}, D_{2222} = E_{2222}, D_{3333} = E_{3333}, D_{1122} = E_{1122}, D_{1133} = E_{1133}, D_{2233} = E_{2233}. \tag{39}$$

Substituting Eq. (33) into Eq. (35), we have

$$\boldsymbol{D} = a_{00}\boldsymbol{D}^{00} + a_{20}\boldsymbol{D}^{20} + a_{22}\boldsymbol{D}^{22} + a_{40}\boldsymbol{D}^{40} + a_{42}\boldsymbol{D}^{42} + a_{44}\boldsymbol{D}^{44}, \tag{40}$$

where

$$\boldsymbol{D}^{ij} = \int_{H_\delta(\boldsymbol{x})} \frac{c^1(\|\boldsymbol{\xi}\|)cos(j\varphi)P_i^j(cos(\theta))}{2}\boldsymbol{\xi}\otimes\boldsymbol{\xi}\otimes\boldsymbol{\xi}\otimes\boldsymbol{\xi}\,\mathrm{d}V_{x'}, \quad i,j = 0,2,4 \text{ and } j \le i. \tag{41}$$

Then, Eq. (39) implies that

$$\begin{bmatrix} D^{00}_{1111} & D^{20}_{1111} & D^{22}_{1111} & D^{40}_{1111} & D^{42}_{1111} & D^{44}_{1111} \\ D^{00}_{2222} & D^{20}_{2222} & D^{22}_{2222} & D^{40}_{2222} & D^{42}_{2222} & D^{44}_{2222} \\ D^{00}_{3333} & D^{20}_{3333} & D^{22}_{3333} & D^{40}_{3333} & D^{42}_{3333} & D^{44}_{3333} \\ D^{00}_{1122} & D^{20}_{1122} & D^{22}_{1122} & D^{40}_{1122} & D^{42}_{1122} & D^{44}_{1122} \\ D^{00}_{1133} & D^{20}_{1133} & D^{22}_{1133} & D^{40}_{1133} & D^{42}_{1133} & D^{44}_{1133} \\ D^{00}_{2233} & D^{20}_{2233} & D^{22}_{2233} & D^{40}_{2233} & D^{42}_{2233} & D^{44}_{2233} \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{20} \\ a_{22} \\ a_{40} \\ a_{42} \\ a_{44} \end{bmatrix} = \begin{bmatrix} E_{1111} \\ E_{2222} \\ E_{3333} \\ E_{1122} \\ E_{1133} \\ E_{2233} \end{bmatrix}. \tag{42}$$

By solving the above equations, the parameters in Eq. (33) can be obtained.

Appendix C. Transverse isotropic model

In the classical transverse isotropic model, there are five independent variables in $\boldsymbol{E}$ (assuming that $\boldsymbol{e}_3$ is the reference direction of an axisymmetric material). These variables are

$$E_{1111}, E_{3333}, E_{1122}, E_{1133}, E_{1313}. \tag{43}$$

In the peridynamic transverse isotropic model, there are only three independent variables in $\boldsymbol{D}$ because of the centrosymmetry of the long-range force. These variables are

$$D_{1111}, D_{3333}, D_{1133} = D_{1313}, \tag{44}$$

and D_{1122} depends on the other coefficients. Therefore, Eq. (36) implies that

$$D_{1111} = E_{1111}, D_{3333} = E_{3333}, D_{1133} = E_{1133}. \tag{45}$$

Substituting Eq. (34) into Eq. (35), we have

$$\boldsymbol{D} = a_0\boldsymbol{D}^0 + a_2\boldsymbol{D}^2 + a_4\boldsymbol{D}^4, \tag{46}$$

where

$$\boldsymbol{D}^i = \int_{H_\delta(\boldsymbol{x})} \frac{c^1(\|\boldsymbol{\xi}\|)P^0_i(cos(\theta))}{2} \boldsymbol{\xi} \otimes \boldsymbol{\xi} \otimes \boldsymbol{\xi} \otimes \boldsymbol{\xi} \, \mathrm{d}V_{x'}, \quad i = 0, 2, 4. \tag{47}$$

Then, Eq. (45) implies that

$$\begin{bmatrix} D^0_{1111} & D^2_{1111} D^4_{1111} \\ D^0_{3333} & D^2_{3333} D^4_{3333} \\ D^0_{1133} & D^2_{1133} D^4_{1133} \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \\ a_4 \end{bmatrix} = \begin{bmatrix} E_{1111} \\ E_{3333} \\ E_{1133} \end{bmatrix}. \tag{48}$$

By solving the above equations, the parameters in Eq. (34) can be obtained.