Engineering with Computers (2023) 39:185–219
https://doi.org/10.1007/s00366-021-01537-x

ORIGINAL ARTICLE

# Implicit implementation of the nonlocal operator method: an open source code

Yongzheng Zhang[1] · Huilong Ren[1]

Received: 12 August 2021 / Accepted: 19 October 2021 / Published online: 8 January 2022


**Abstract**
In this paper, we present an open-source code for the first-order and higher-order nonlocal operator method (NOM) including a detailed description of the implementation. The NOM is based on so-called support, dual-support, nonlocal operators, and an operate energy functional ensuring stability. The nonlocal operator is a generalization of the conventional differential operators. Combined with the method of weighed residuals and variational principles, NOM establishes the residual and tangent stiffness matrix of operate energy functional through some simple matrix without the need of shape functions as in other classical computational methods such as FEM. NOM only requires the definition of the energy drastically simplifying its implementation. The implementation in this paper is focused on linear elastic solids for sake of conciseness through the NOM can handle more complex nonlinear problems. The NOM can be very flexible and efficient to solve partial differential equations (PDEs), it's also quite easy for readers to use the NOM and extend it to solve other complicated physical phenomena described by one or a set of PDEs. Finally, we present some classical benchmark problems including the classical cantilever beam and plate-with-a-hole problem, and we also make an extension of this method to solve complicated problems including phase-field fracture modeling and gradient elasticity material.

**Keywords** Nonlocal operator method · Operator energy functional · Implicit · Dual-support · Variational principle · Taylor series expansion · Stiffness matrix

## 1 Introduction

The nonlocal theory of elasticity [1–5] primarily considers distant action forces between objects. Different from the concept of local theory, an object can interact without physical interaction with a different object. The theory is of great significance to solve many physical problems, such as the law of universal gravitation. Some notable nonlocal numerical methods were proposed according to nonlocal interaction in nonlocal continuum field theories.

Peridynamics (PD) [6–8] is a formulation of continuum mechanics on the basis of the concept of nonlocal integration, PD avoids the singularity of the traditional local differential equations when solving discontinuous problems. One key application of PD is fracture. It shares the same advantages as the Cracking Particles Method (CPM) presented in [9–11]. In contrast to many other discrete crack approaches as presented in [12–20], PD does not require the representation of the discrete crack surface and associated crack tracking algorithms. PD also has been successfully applied to rock fracture and soil damage analysis, such as impact fracture [21, 22], composite material separation [23, 24]and beam and plate structures [25]. However, to eliminate erroneous wave reflection and ghost force among particles, all traditional PD formulas must use the same horizon size. In many applications, to enhance calculating performance, its necessary to use different horizon sizes for the calculation of particles with non-uniform spatial distribution, such as adaptive encryption, multi-scale simulation, and multi-body analysis. In other words, in order to balance the calculation efficiency and calculation accuracy, we hope that PD can be based on the distribution characteristics of the particles, but if the size of the near field is used, it will result in the generation of false stress waves and the problem cannot be solved correctly. To address the aforementioned

✉ Yongzheng Zhang
sdustzyz@163.com

1 Institute of Structural Mechanics, Bauhaus-University Weimar, 99423 Weimar, Germany

issue, dual-horizon PD [26, 27] was developed to improve computing efficiency and to allow for varying horizon sizes. The dual-horizon is the dual term of the horizon when variable horizons are used in the inhomogeneous discretization. It separates the horizon that exerts forces and counter forces between the particles, thereby solving the problem of false stress caused by the horizons. In addition, other nonlocal models mainly include nonlocal linear elasticity [28–30], dynamics of nonlocal fluid [31, 32], electromagnetic nonlocal theory [33, 34], nonlocal damage model [35–37] and nonlocal calculus [38–41].

In most of the current nonlocal numerical methods, the governing equation is generally solved by the explicit time integration method [25, 42]. The explicit algorithm [43–47] is based on dynamic equations, without iteration and has good stability. However, the mass matrix is required to be a diagonal matrix in the explicit solution and the speed advantage can be exerted only when the unit-level calculation is as small as possible. Therefore, the reduced integration method is often used, which is easy to excite the hourglass mode and affect the stress and strain calculation accuracy. To ensure its conditional convergence, the integration step must be less than the minimum value of the free vibration period of all elements. The integration time step is generally 1/1000 to 1/100 of the implicit step, which is suitable for solving instant or short-time loading problems. For the quasi-static (quasi-static) problem, when the total load is fixed, the calculation takes a long time, which seriously affects the calculation efficiency. Compared with the explicit method, implicit method [48–50] has advantages for fast convergence and lower computational cost.

In recent years, several numerical approaches based on peridynamics operators have been proposed, see e.g. the contributions in [51–53]. A new computational method based on nonlocal operators is the NOM first proposed in [54] for electromagnetic problems. The approach has been later on extended to mechanical problems in [55, 56]. The NOM can be considered as a generalization of non-ordinary state-based PD. It has been applied to numerous challenging problems in solid mechanics and can be a viable alternative to FEM or meshless methods. In order to acquire the differential operators, FEM and meshless methods are required to establish the shape functions as well as compute their derivatives, however, NOM can acquire the differential operators easily without the use of shape functions. The tangent stiffness is obtained naturally by simply defining an energy function thus drastically simplifying its implementation. In combination with the weighted residual method and variational principle, the residual and the tangent stiffness matrix can be established by NOM with ease. NOM is enhanced here also with operator energy functional to achieve the linear consistency of the field and avoid instabilities. Although the theoretical framework of the NOM has been proposed, other benefits of this numerical method have not been thoroughly discussed. Furthermore, the details of the derivation of the first-order and higher-order nonlocal differential operators, construction of the first-order and higher-order operator energy functional, the derivation of the residual and tangent stiffness matrix of operate energy functional and the detailed implementation procedure of first-order and higher-order NOM has not been shown before. The purpose of this paper is to describe in detail the method of implicitly implementing for first-order and higher-order NOM, which mainly including the derivation of the first-order and higher-order nonlocal differential operates, the detailed form of first-order and higher-order tangent stiffness matrix for operate energy functional and elastic material constitutions in different conditions. The Mathematica code of first-order and higher-order NOM is presented and explained in detail, and it will be an effective tool for studying complicated physical problems.

The remainder of the paper is outlined as follows: In Section 2, we briefly reviewed the NOM and elaborated on the fundamental concept of support and dual-support. In Section 3, we derived the first and higher-order implicit nonlocal differential operators based on the Taylor series expansion. Then the elastic material constitutions are also derived. Hereafter, to remove the zero-energy mode, the first and higher-order operator energy functional by the nonlocal operator is constructed, combined with the method of weighed residuals and variational principles, residual and tangent stiffness matrix of operate energy functional are established. Finally, we present the detailed steps of the first-order and higher-order NOM implementation process. To demonstrate the capabilities of NOM, four numerical examples are presented in section 4. Finally, we conclude in section 5.

## 2 Nonlocal operator method (NOM)

Consider the initial and present configurations of a solid, as depicted in Fig. 1a, let $\mathbf{x}_i$ be spatial coordinates in the domain $\boldsymbol{\Omega}$; $\boldsymbol{\xi}_{ij} := \mathbf{x}_j - \mathbf{x}_i$ is a spatial vector starts from $\mathbf{x}_i$ to $\mathbf{x}_j$; $\mathbf{u}_i := \mathbf{u}(\mathbf{x}_i, t)$ and $\mathbf{u}_j := \mathbf{u}(\mathbf{x}_j, t)$ are the field value for $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively; $\mathbf{u}_{ij} := \mathbf{u}_j - \mathbf{u}_i$ is the relative field vector for spatial vector $\boldsymbol{\xi}_{ij}$.

Support $\mathcal{S}_i$ of point $\mathbf{x}_i$ is the domain where any spatial point $\mathbf{x}_j$ forms spatial vector $\boldsymbol{\xi}_{ij}(= \mathbf{x}_j - \mathbf{x}_i)$ from $\mathbf{x}_i$ to $\mathbf{x}_j$. The support serves as the basis for the nonlocal operators. It should be noted there is no restriction on the support shapes(such as spherical, cube and so on). Dual-support is defined as a union of the points whose supports include $\mathbf{x}$, indicated by

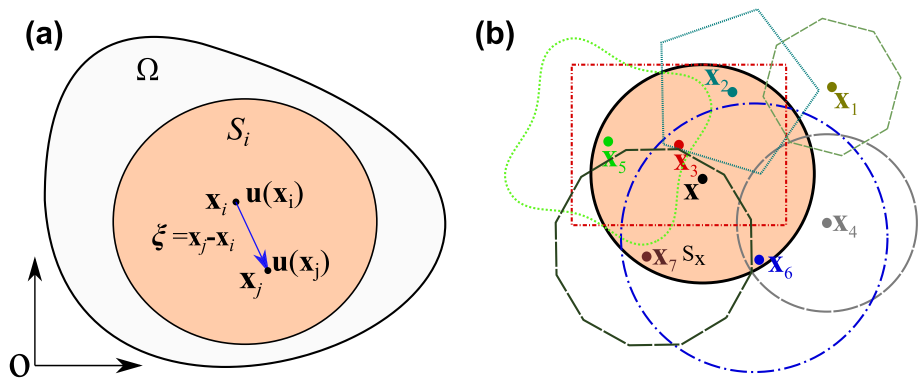


**Fig. 1** **a** The deformed body configuration. **b** Support and dual-support schematic diagram, $\mathcal{S}_\mathbf{x} = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$, $\mathcal{S}'_\mathbf{x} = \{\mathbf{x}_3, \mathbf{x}_6, \mathbf{x}_7\}$

$$\mathcal{S}'_i = \{\mathbf{x}_j | \mathbf{x}_i \in \mathcal{S}_j\}. \tag{1}$$

Point $\mathbf{x}_j$ forms dual-vector $\xi'_{ij}(= \mathbf{x}_i - \mathbf{x}_j = -\xi_{ij})$ in $\mathcal{S}'_i$. In addition, $\xi'_{ij}$ is the spatial vector established in $\mathcal{S}_j$. Figure 1b illustrates the concept of support and dual-support.

In calculus calculations, the NOM replaces the local operator with the fundamental nonlocal operators. By substituting the local differential operator with the corresponding nonlocal operator, the functional defined by the local differential operator can be utilized to establish the residual or tangent stiffness matrix.

The nonlocal gradient operator for a vector field $\mathbf{u}$ and scalar field $u$ at point $\mathbf{x}_i$ in support $\mathcal{S}_i$ are defined as [55]

$$\tilde{\nabla}\mathbf{u}_i(\text{or}\tilde{\nabla} \otimes \mathbf{u}_i) := \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})\mathbf{u}_{ij} \otimes \xi_{ij}\mathrm{d}V_j \cdot \left( \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\mathrm{d}V_j \right)^{-1}, \tag{2}$$

$$\tilde{\nabla}u_i := \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})u_{ij}\xi_{ij}\mathrm{d}V_j \cdot \left( \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\mathrm{d}V_j \right)^{-1}, \tag{3}$$

where $\mathrm{w}(\xi_{ij})$ is the weight function for vector $\xi_{ij}$ in support $\mathcal{S}_i$.

By nodal integration, the nonlocal gradient operator and its variation for a vector field $\mathbf{u}$ in discrete form can be written as

$$\tilde{\nabla}\mathbf{u}_i = \sum_{j\in\mathcal{S}_i} \mathrm{w}(\xi_{ij})\mathbf{u}_{ij} \otimes \xi_{ij}\Delta V_j \cdot \left( \sum_{j\in\mathcal{S}_i} \mathrm{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\Delta V_j \right)^{-1}, \tag{4}$$

$$\tilde{\nabla}\delta\mathbf{u}_i = \sum_{j\in\mathcal{S}_i} \mathrm{w}(\xi_{ij})\delta\mathbf{u}_{ij} \otimes \xi_{ij}\Delta V_j \cdot \left( \sum_{j\in\mathcal{S}_i} \mathrm{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\Delta V_j \right)^{-1}. \tag{5}$$

The operator energy functional for vector field at a point $\mathbf{x}_i$ is defined as

$$\mathcal{F}_i^{hg} = \frac{p^{hg}}{2m_{\mathbf{K}_i}} \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})(\tilde{\nabla}\mathbf{u}_i \cdot \xi_{ij} - \mathbf{u}_{ij}) \cdot (\tilde{\nabla}\mathbf{u}_i \cdot \xi_{ij} - \mathbf{u}_{ij})\mathrm{d}V_j, \tag{6}$$

where $\frac{p^{hg}}{2m_{\mathbf{K}_i}}$ is a coefficient for the operator energy functional, $p^{hg}$ is the penalty coefficient, $m_{\mathbf{K}_i}(= \mathrm{tr}[\mathbf{K}_i])$ is the normalization coefficient, $\mathbf{K}_i$ is a shape tensor and it can be defined as

$$\mathbf{K}_i = \int_{\mathcal{S}_i} \mathrm{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\mathrm{d}V_j \tag{7}$$

## 3 Implementation

### 3.1 Derivation of the first-order implicit nonlocal differential operators

The displacement field at any point in the elastic body can be represented by three displacement components $u$, $v$, $w$ along the rectangular coordinate axis, and it's three dimension vector form is

$$\mathbf{u} = (u, v, w)^T. \tag{8}$$

The first-order Taylor series expansion at point (0,0) for a displacement field $\mathbf{u}$ is given as

$$\mathbf{u}' = \mathbf{u} + \nabla\mathbf{u} \cdot \xi + O(\xi^2), \tag{9}$$

where $\nabla := (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$; $\xi := (x, y, z)^T$ denotes the initial bond vector, $O(\xi^2)$ represents higher-order terms, and for linear field $O(\xi^2) = 0$.

$\left(\mathbf{u}' - \mathbf{u}\right) \otimes \xi$ can be obtained from Eq. 9 and can be shown as

$$\left(\mathbf{u}' - \mathbf{u}\right) \otimes \xi = \nabla\mathbf{u} \cdot \xi \otimes \xi.$$

Integrate $\left(\mathbf{u}' - \mathbf{u}\right) \otimes \xi$ in the domain $\mathcal{S}_i$, one will obtain

$$\int_{\mathcal{S}_i} \mathrm{w}(\xi)\left(\mathbf{u}' - \mathbf{u}\right) \otimes \xi\mathrm{d}V = \nabla\mathbf{u} \int_{\mathcal{S}_i} \mathrm{w}(\xi)\xi \otimes \xi\mathrm{d}V. \tag{10}$$

The gradient operate $\nabla\mathbf{u}$ can be expressed as

$$\nabla \mathbf{u} = \int_{\mathcal{S}_i} \mathrm{w}(\boldsymbol{\xi})(\mathbf{u}' - \mathbf{u}) \otimes \boldsymbol{\xi} \mathrm{d}V \cdot \left[\int_{\mathcal{S}_i} \mathrm{w}(\boldsymbol{\xi})\boldsymbol{\xi} \otimes \boldsymbol{\xi} \mathrm{d}V\right]^{-1}. \tag{11}$$

The discrete form of Eq. 11 at point $\mathbf{x}_i$ can be shown as

$$\nabla \mathbf{u}_i = \sum_{j\in\mathcal{S}_i} \mathrm{w}(\boldsymbol{\xi}_{ij})(\mathbf{u}_j - \mathbf{u}_i) \otimes \boldsymbol{\xi}_{ij}\Delta V_j \cdot \left[\sum_j \mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{\xi}_{ij} \otimes \boldsymbol{\xi}_{ij}\Delta V_j\right]^{-1}. \tag{12}$$

To simplify the equation more conveniently, letting $\mathcal{S}_i = [i, j_1, j_2, j_3, \cdots, j_n]$, $\mathbf{R}_j = \mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{\xi}_{ij}^T V_j \cdot \left[\sum_{j\in\mathcal{S}_i} \mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{\xi}_{ij} \otimes \boldsymbol{\xi}_{ij}\Delta V_j\right]^{-1} = (\xi_{x_j}, \xi_{y_j}, \xi_{z_j})$. The nonlocal gradient operator $\tilde{\nabla}\mathbf{u}$ at point $\mathbf{x}_i$ can be rewritten as

$$\tilde{\nabla}\mathbf{u}_i = \sum_{j\in\mathcal{S}_i} (\mathbf{u}_j - \mathbf{u}_i)\left[\xi_{x_j}, \xi_{y_j}, \xi_{z_j}\right]. \tag{13}$$

According to Eq. 13, the matrix form of nonlocal gradient operator $\tilde{\nabla}\mathbf{u}$ at point $\mathbf{x}_i$ for vector field can be expressed as

$$\tilde{\nabla}\mathbf{u}_i = \begin{bmatrix} \frac{\partial u_i}{\partial x} & \frac{\partial u_i}{\partial y} & \frac{\partial u_i}{\partial z} \\ \frac{\partial v_i}{\partial x} & \frac{\partial v_i}{\partial y} & \frac{\partial v_i}{\partial z} \\ \frac{\partial w_i}{\partial x} & \frac{\partial w_i}{\partial y} & \frac{\partial w_i}{\partial z} \end{bmatrix} = \begin{bmatrix} -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & \xi_{x_{j1}} & \cdots & \xi_{x_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & \xi_{y_{j1}} & \cdots & \xi_{y_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{z_j} & \xi_{z_{j1}} & \cdots & \xi_{z_{jn}} \end{bmatrix} \begin{bmatrix} u_i & v_i & w_i \\ u_{j1} & v_{j1} & w_{j1} \\ \cdots & \cdots & \cdots \\ u_{jn} & v_{jn} & w_{jn} \end{bmatrix}. \tag{14}$$

For the convenience of calculations, we transform the matrix form of nonlocal gradient operator $\tilde{\nabla}\mathbf{u}_i$ for vector field into vector form and it can be rewritten as

$$\begin{aligned} \tilde{\nabla}\mathbf{u}_i &= \left[\frac{\partial u_i}{\partial x}, \frac{\partial u_i}{\partial y}, \frac{\partial u_i}{\partial z}, \frac{\partial v_i}{\partial x}, \frac{\partial v_i}{\partial y}, \frac{\partial v_i}{\partial z}, \frac{\partial w_i}{\partial x}, \frac{\partial w_i}{\partial y}, \frac{\partial w_i}{\partial z}\right]^T \\ &= \begin{bmatrix} -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & 0 & 0 & \xi_{x_{j1}} & 0 & 0 & \cdots & \xi_{x_{jn}} & 0 & 0 \\ 0 & -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & 0 & 0 & \xi_{y_{j1}} & 0 & \cdots & 0 & \xi_{y_{jn}} & 0 \\ 0 & 0 & -\sum_{j\in\mathcal{S}_i}\xi_{z_j} & 0 & 0 & \xi_{z_{j1}} & \cdots & 0 & 0 & \xi_{z_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & 0 & 0 & \xi_{x_{j1}} & 0 & 0 & \cdots & \xi_{x_{jn}} & 0 & 0 \\ 0 & -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & 0 & 0 & \xi_{y_{j1}} & 0 & \cdots & 0 & \xi_{y_{jn}} & 0 \\ 0 & 0 & -\sum_{j\in\mathcal{S}_i}\xi_{z_j} & 0 & 0 & \xi_{z_{j1}} & \cdots & 0 & 0 & \xi_{z_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & 0 & 0 & \xi_{x_{j1}} & 0 & 0 & \cdots & \xi_{x_{jn}} & 0 & 0 \\ 0 & -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & 0 & 0 & \xi_{y_{j1}} & 0 & \cdots & 0 & \xi_{y_{jn}} & 0 \\ 0 & 0 & -\sum_{j\in\mathcal{S}_i}\xi_{z_j} & 0 & 0 & \xi_{z_{j1}} & \cdots & 0 & 0 & \xi_{z_{jn}} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \\ u_{j1} \\ v_{j1} \\ w_{j1} \\ \cdots \\ u_{jn} \\ v_{jn} \\ w_{jn} \end{bmatrix} = \mathscr{B}_i \mathscr{U}_i. \end{aligned} \tag{15}$$

Similarly, the nonlocal gradient operator $\tilde{\nabla}u$ at point $\mathbf{x}_i$ for scalar field is written as

$$\tilde{\nabla}u_i = \begin{bmatrix} \frac{\partial u_i}{\partial x} \\ \frac{\partial u_i}{\partial y} \\ \frac{\partial u_i}{\partial z} \end{bmatrix} = \begin{bmatrix} -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & \xi_{x_{j1}} & \cdots & \xi_{x_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & \xi_{y_{j1}} & \cdots & \xi_{y_{jn}} \\ sum_{j\in\mathcal{S}_i}\xi_{z_j} & \xi_{z_{j1}} & \cdots & \xi_{z_{jn}} \end{bmatrix} \begin{bmatrix} u_i \\ u_{j1} \\ \cdots \\ u_{jn} \end{bmatrix} = \mathscr{B}_i \mathscr{U}_i. \tag{16}$$

The 2D form of the first-order nonlocal gradient operators is given in Appendix A.

## 3.2 Derivation of the higher-order implicit nonlocal differential operators

The higher-order NOM is based on higher-order Taylor series expansion of a multi-variable function. Consider a vector field $\mathbf{u}$ at point $\mathbf{x}_j$ ($\mathbf{x}_j \in \mathcal{S}_i$). For convenience, we shorthand $\mathbf{u}(\mathbf{x}_j)$ by $\mathbf{u}_j$, which can be estimated via a Taylor expansion based on $\mathbf{u}_i$ in $\mathbb{r}$ dimensions with the highest order of derivatives as $\mathbb{n}$:

$$\mathbf{u}_j = \mathbf{u}_i + \sum_{(\mathbb{n}_1,\ldots,\mathbb{n}_\mathbb{r})\in\alpha_\mathbb{r}^\mathbb{n}} \frac{\xi_1^{\mathbb{n}_1}\ldots\xi_\mathbb{r}^{\mathbb{n}_\mathbb{r}}}{\mathbb{n}_1!\ldots\mathbb{n}_\mathbb{r}!}\mathbf{u}_{i,\mathbb{n}_1\ldots\mathbb{n}_\mathbb{r}} + O(\xi^{|\alpha|+1}), \tag{17}$$

where

$$\xi_{ij} = (\mathbf{x}_{j1} - \mathbf{x}_{i1}, \ldots, \mathbf{x}_{jd} - \mathbf{x}_{id}) \tag{18}$$

$$\mathbf{u}_{i,\mathbb{n}_1\ldots\mathbb{n}_\mathbb{r}} = \frac{\partial^{\mathbb{n}_1+\ldots+\mathbb{n}_\mathbb{r}}\mathbf{u}_i}{\partial\mathbf{x}_{i1}^{\mathbb{n}_1}\ldots\partial\mathbf{x}_{id}^{\mathbb{n}_\mathbb{r}}} \tag{19}$$

$$|\alpha| = \max(\mathbb{n}_1 + \cdots + \mathbb{n}_\mathbb{r}), \tag{20}$$

where $\mathbb{r}$ represents the dimension and $\mathbb{n}$ denotes the highest order of derivatives. $\alpha_\mathbb{r}^\mathbb{n}$ is a compilation of multi-indexes that have been flattened and it can be written as

$$\alpha_\mathbb{r}^\mathbb{n} = \{(\mathbb{n}_1, \ldots, \mathbb{n}_\mathbb{r}) | 1 \le \sum_{i=1}^{\mathbb{r}} \mathbb{n}_i \le \mathbb{n},\ \mathbb{n}_i \in \mathbb{N}^0, 1 \le i \le \mathbb{r}\}, \tag{21}$$

where $\mathbb{N}^0 = \{0, 1, 2, \ldots\}$.

To eliminate the round-off error in Eq. 17, we here include the characteristic length scale $\mathbb{l}_i$ of support $\mathcal{S}_i$ at $\mathbf{u}_i$, and the modified Taylor series expansion may be rewritten as

$$
\begin{aligned}
\mathbf{u}_j &= \mathbf{u}_i + \sum_{(\mathbb{n}_1,\dots,\mathbb{n}_\mathbb{r})\in\alpha_\mathbb{r}^\mathbb{n}} \frac{\xi_1^{\mathbb{n}_1}\dots\xi_\mathbb{r}^{\mathbb{n}_\mathbb{r}}}{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}\left(\frac{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}{\mathbb{n}_1!\dots\mathbb{n}_\mathbb{r}!}u_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}}\right) + O(\xi^{\mathbb{n}+1}) \\
&= \mathbf{u}_i + \sum_{(\mathbb{n}_1,\dots,\mathbb{n}_\mathbb{r})\in\alpha_\mathbb{r}^\mathbb{n}} \frac{\xi_1^{\mathbb{n}_1}\dots\xi_\mathbb{r}^{\mathbb{n}_\mathbb{r}}}{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}\,\mathbf{u}^{\mathbb{l}}_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}} + O(\xi^{\mathbb{n}+1}).
\end{aligned}
\tag{22}
$$

For any multi-index $(\mathbb{n}_1,\dots,\mathbb{n}_\mathbb{r})\in\alpha_\mathbb{r}^\mathbb{n}$, $\mathbf{u}^{\mathbb{l}}_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}} = \frac{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}{\mathbb{n}_1!\dots\mathbb{n}_\mathbb{r}!}\mathbf{u}_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}}, \forall(\mathbb{n}_1,\dots,\mathbb{n}_\mathbb{r})\in\alpha_\mathbb{r}^\mathbb{n}$. We let $\boldsymbol{p}_j^{\mathbb{l}}$, $\partial_\alpha^{\mathbb{l}}\mathbf{u}_i$ and $\partial_\alpha\mathbf{u}_i$ denotes the flattened polynomials, scaled partial derivatives, partial derivatives, respectively, according to multi-index notation $\alpha_\mathbb{r}^\mathbb{n}$ and they can be shown as

$$
\begin{aligned}
\boldsymbol{p}_j^{\mathbb{l}} &= \left(\frac{\xi_\mathbb{r}}{\mathbb{l}},\dots,\frac{\xi_1^{\mathbb{n}_1}\dots\xi_\mathbb{r}^{\mathbb{n}_\mathbb{r}}}{\mathbb{l}^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}},\dots,\frac{\xi_1^{\mathbb{n}}}{\mathbb{l}^{\mathbb{n}}}\right)^T \\
\partial_\alpha^{\mathbb{l}}\mathbf{u}_i &= \left(\mathbf{u}^{\mathbb{l}}_{i,0\dots1},\dots,\mathbf{u}^{\mathbb{l}}_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}},\dots,\mathbf{u}^{\mathbb{l}}_{i,\mathbb{n}\dots0}\right)^T \\
\partial_\alpha\mathbf{u}_i &= \left(\mathbf{u}_{i,0\dots1},\dots,\mathbf{u}_{i,\mathbb{n}_1\dots\mathbb{n}_\mathbb{r}},\dots,\mathbf{u}_{i,\mathbb{n}\dots0}\right)^T.
\end{aligned}
\tag{23}
$$

The $\mathbb{l}$ in the Eq. 23 allows the terms of the same characteristic scale for length thereby eliminating computational round-off error. The current partial derivatives can be recovered by

$$
\partial_\alpha\mathbf{u}_i = \operatorname{diag}\left[\mathbb{l}_i,\dots,\frac{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}{\mathbb{n}_1!\dots\mathbb{n}_\mathbb{r}!},\dots,\frac{\mathbb{l}_i^{\mathbb{n}}}{\mathbb{n}!}\right]^{-1}\partial_\alpha^{\mathbb{l}}\mathbf{u}_i. \tag{24}
$$

In which diag $[\mathcal{X}_1,\dots,\mathcal{X}_\mathbb{n}]$ signifies a diagonal matrix of $\mathcal{X}_1,\dots,\mathcal{X}_\mathbb{n}$, whose diagonal entries begin from the upper left corner. Hence, the expansion of the Taylor series with $\mathbf{u}_i$ to the left of the Eq. 22 can be rewritten as

$$
\mathbf{u}_{ij} = (\partial_\alpha^{\mathbb{l}}\mathbf{u}_i)^T\boldsymbol{p}_j^{\mathbb{l}}, \forall j\in\mathcal{S}_i, \tag{25}
$$

where $\mathbf{u}_{ij} = \mathbf{u}_j - \mathbf{u}_i$.

Integrate $\mathbf{u}_{ij}$ with weighted coefficient $\mathbb{w}(\xi_{ij})(\boldsymbol{p}_j^{\mathbb{l}})^T$ in support $\mathcal{S}_i$, we obtain

$$
\begin{aligned}
\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\mathbf{u}_{ij}(\boldsymbol{p}_j^{\mathbb{l}})^T\mathrm{d}V_j &= (\partial_\alpha^{\mathbb{l}}\mathbf{u}_i)^T\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\otimes(\boldsymbol{p}_j^{\mathbb{l}})^T\mathrm{d}V_j \\
&= (\partial_\alpha\mathbf{u}_i)^T\,\mathbb{L}_i\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\otimes(\boldsymbol{p}_j^{\mathbb{l}})^T\mathrm{d}V_j,
\end{aligned}
\tag{26}
$$

where

$$
\mathbb{L}_i = \operatorname{diag}\left[\mathbb{l}_i,\dots,\frac{\mathbb{l}_i^{\mathbb{n}_1+\dots+\mathbb{n}_\mathbb{r}}}{\mathbb{n}_1!\dots\mathbb{n}_\mathbb{r}!},\dots,\frac{\mathbb{l}_i^{\mathbb{n}}}{\mathbb{n}!}\right]. \tag{27}
$$

Therefore, the nonlocal operator $\tilde{\partial}_\alpha\mathbf{u}_i$ and its variation can be obtained as

$$
\begin{cases}
\tilde{\partial}_\alpha\mathbf{u}_i = \mathbf{K}_{\alpha i}\cdot\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\mathbf{u}_{ij}\mathrm{d}V_j \\
\tilde{\partial}_\alpha\delta\mathbf{u}_i = \mathbf{K}_{\alpha i}\cdot\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}(\delta\mathbf{u}_j-\delta\mathbf{u}_i)\mathrm{d}V_j
\end{cases}
\tag{28}
$$

where

$$
\mathbf{K}_{\alpha i} = \mathbb{L}_i^{-1}\left(\int_{\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\otimes(\boldsymbol{p}_j^{\mathbb{l}})^T\mathrm{d}V_j\right)^{-1} \tag{29}
$$

In this paper, points of the domain $\Omega$ in $\mathcal{S}_i$ be symbolized as

$$
\mathcal{S}_i = \{j_1,\dots,j_k,\dots,j_{n_i}\} \tag{30}
$$

where $j_1,\dots,j_k,\dots,j_n$ denote the global indices of neighbors for point $\mathbf{x}_i$, and $n$ denotes the quantity of $i$'s neighbors in $\mathcal{S}_i$.

The discrete form of Eq. 28 can be shown as

$$
\tilde{\partial}_\alpha\mathbf{u}_i = \mathbf{K}_{\alpha i}\cdot\sum_{j\in\mathcal{S}_i}\mathbf{u}_{ij}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\Delta V_j = \mathbf{K}_{\alpha i}\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}\Delta\mathbf{u}_i \tag{31}
$$

$$
\tilde{\partial}_\alpha\delta\mathbf{u}_i = \mathbf{K}_{\alpha i}\cdot\sum_{j\in\mathcal{S}_i}\delta\mathbf{u}_{ij}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\Delta V_j = \mathbf{K}_{\alpha i}\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}\delta\Delta\mathbf{u}_i \tag{32}
$$

where

$$
\begin{aligned}
\mathbf{K}_{\alpha i} &= \mathbb{L}_i^{-1}\left(\sum_{j\in\mathcal{S}_i}\mathbb{w}(\xi_{ij})\boldsymbol{p}_j^{\mathbb{l}}\otimes(\boldsymbol{p}_j^{\mathbb{l}})^T\Delta V_j\right)^{-1} \\
\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i} &= \left(\mathbb{w}(\xi_{ij_1})\boldsymbol{p}^{\mathbb{l}}_{j_1}\Delta V_{j_1},\dots,\mathbb{w}(\xi_{ij_{n_i}})\boldsymbol{p}^{\mathbb{l}}_{j_{n_i}}\Delta V_{j_{n_i}}\right) \\
\Delta\mathbf{u}_i &= (\mathbf{u}_{ij_1},\dots,\mathbf{u}_{ij_k},\dots,\mathbf{u}_{ij_n})^T
\end{aligned}
\tag{33}
$$

The nonlocal operator gives all partial derivatives with the highest order up to $\mathbb{n}$. In PDEs, the group of derivatives is a subgroup of the nonlocal operator. Each term in $\tilde{\partial}_\alpha\mathbf{u}_i$ corresponds to a row of $\mathbf{K}_{\alpha i}\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}$ multiplied by $\Delta\mathbf{u}_i$. Equation 31 can be used to substitute the differential operators in PDEs to generate a strong form of algebraic equations. Meanwhile, we can solve the linear (nonlinear) weak formulations using the weighted residual methodology and the variational principle, hence the variation of $\partial_\alpha\mathbf{u}_i$ in Eq. 32 is required in these circumstances. Equation 31 can also be shown more succinctly as

$$
\tilde{\partial}_\alpha\mathbf{u}_i = \mathbf{K}_{\alpha i}\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}\Delta\mathbf{u}_i = \left[-(1,\cdots,1)_n\mathbf{K}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i},\mathbf{K}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}\right]\begin{bmatrix}\mathbf{u}_i\\ \mathbf{u}_{j_1}\\ \mathbf{u}_{j_2}\\ \cdots\\ \mathbf{u}_{j_n}\end{bmatrix} = \mathbf{B}_{\alpha i}\mathbf{U}_i \tag{34}
$$

where $\mathbf{B}_{\alpha i}$ is the nonlocal operator coefficient matrix of point $\mathbf{x}_i$, $(1,\cdots,1)_{n_p}\mathbf{K}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}$ is the column sum of $\mathbf{K}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}$. Using the nodal values in support, the operator matrix generates all partial derivatives of maximal order smaller than $|\alpha| + 1$.

### 3.3 Elastic material constitution

For elastic material, the strain energy functional $\psi$ is a function of the deformation gradient $\boldsymbol{F}$. According to the principles of traditional solid mechanics, the deformation gradient $\boldsymbol{F}$ in 3D form expressed as

$$\boldsymbol{F} = \nabla\mathbf{u} + \mathbf{I}_{3\times3} = \begin{bmatrix} \frac{\partial u}{\partial x}+1 & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}+1 & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z}+1 \end{bmatrix} \tag{35}$$

where $\mathbf{I}_{3\times3}$ denotes the identity tensor.

The first Piola–Kirchhoff stress $\mathbf{P}$ can be derived from the directly derivative of strain energy $\psi(\boldsymbol{F})$ in the context of total Lagrangian formulation, and it can be derived as

$$\mathbf{P} = \frac{\partial\psi(\boldsymbol{F})}{\partial\boldsymbol{F}} = \begin{bmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} & \mathbb{P}_{13} \\ \mathbb{P}_{21} & \mathbb{P}_{22} & \mathbb{P}_{23} \\ \mathbb{P}_{31} & \mathbb{P}_{32} & \mathbb{P}_{33} \end{bmatrix} \tag{36}$$

In addition, the 4th order elastic tensor $\mathbf{D}_4$ can be obtained using derivation of the first Piola–Kirchhoff stress

$$\mathbf{D}_4 = \frac{\partial\mathbf{P}}{\partial\boldsymbol{F}} = \frac{\partial^2\psi(\boldsymbol{F})}{\partial\boldsymbol{F}^T\partial\boldsymbol{F}} \tag{37}$$

To obtain the elasticity matrix, based on the Voigt notation, $\mathbf{D}_4$ can be written as a matrix form $\mathscr{D}_{9\times9}$

$$\mathscr{D}_{9\times9} = \begin{bmatrix} \frac{\partial\mathbb{P}_{11}}{\partial\mathbb{F}_{11}} & \frac{\partial\mathbb{P}_{11}}{\partial\mathbb{F}_{12}} & \cdots & \frac{\partial\mathbb{P}_{11}}{\partial\mathbb{F}_{33}} \\ \frac{\partial\mathbb{P}_{12}}{\partial\mathbb{F}_{11}} & \frac{\partial\mathbb{P}_{12}}{\partial\mathbb{F}_{12}} & \cdots & \frac{\partial\mathbb{P}_{12}}{\partial\mathbb{F}_{33}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\mathbb{P}_{33}}{\partial\mathbb{F}_{11}} & \frac{\partial\mathbb{P}_{33}}{\partial\mathbb{F}_{12}} & \cdots & \frac{\partial\mathbb{P}_{33}}{\partial\mathbb{F}_{33}} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{11}^2} & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{11}\partial\mathbb{F}_{12}} & \cdots & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{11}\partial\mathbb{F}_{33}} \\ \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{12}\partial\mathbb{F}_{11}} & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{22}^2} & \cdots & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{12}\partial\mathbb{F}_{33}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{33}\partial\mathbb{F}_{11}} & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{33}\partial\mathbb{F}_{12}} & \cdots & \frac{\partial^2\psi(\mathbb{F})}{\partial\mathbb{F}_{33}^2} \end{bmatrix}, \tag{38}$$

where $\mathbb{F}(=\mathbb{F}_{11},\mathbb{F}_{12},\cdots,\mathbb{F}_{33}),\mathbb{P}(=\frac{\partial\psi(\mathbb{F})}{\partial\mathbb{F}_{11}},\frac{\partial\psi(\mathbb{F})}{\partial\mathbb{F}_{12}},\cdots,\frac{\partial\psi(\mathbb{F})}{\partial\mathbb{F}_{33}})$ donates the flattened deformation gradient and flattened first Piola–Kirchhoff stress.

In particular, for isotropic linear elastic material, regarding infinitesimal deformations of a continuous linear elastic material with a tiny displacement gradient relative to unity ( i.e. $\nabla\mathbf{u} \ll 1$), any of the strain tensors utilized in finite strain theory (such as the Lagrangian strain tensor) may be geometrically linearized. The non-linear or second-order elements of the finite strain tensor are ignored in this linearization. Hence, we can obtain the Lagrangian strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{F}+\boldsymbol{F}^T) - \mathbf{I}$ and stress tensor $\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\varepsilon}$.

The internal functional energy for 3D linear elastic material can be expressed as

$$\psi(\boldsymbol{\varepsilon})_{3d} = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{D} : \boldsymbol{\varepsilon} = \frac{1}{2}\mathrm{d}\mathscr{U}_{3d}^T\mathscr{D}_{3d}\mathrm{d}\mathscr{U}_{3d} \tag{39}$$

Likewise, the internal energy functional for plane stress and plane strain conditions can be shown as follows

$$\begin{aligned} \psi(\boldsymbol{\varepsilon})_{\text{plane stress}} &= \frac{1}{2}\mathrm{d}\mathscr{U}_{2d}^T\mathscr{D}_{\text{plane stress}}\mathrm{d}\mathscr{U}_{2d} \\ \psi(\boldsymbol{\varepsilon})_{\text{plane strain}} &= \frac{1}{2}\mathrm{d}\mathscr{U}_{2d}^T\mathscr{D}_{\text{plane strain}}\mathrm{d}\mathscr{U}_{2d} \end{aligned} \tag{40}$$

where

$$\mathrm{d}\mathscr{U}_{2d} = \left(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial v}{\partial x},\frac{\partial v}{\partial y}\right)^T \tag{41}$$

$$\mathrm{d}\mathscr{U}_{3d} = \left(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial u}{\partial z},\frac{\partial v}{\partial x},\frac{\partial v}{\partial y},\frac{\partial v}{\partial z},\frac{\partial w}{\partial x},\frac{\partial w}{\partial y},\frac{\partial w}{\partial z}\right)^T \tag{42}$$

For linear elastic material, the elastic matrix $\mathscr{D}$ in plane stress, plane strain and 3D conditions can be expressed as

$$\mathscr{D}_{\text{plane stress}} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & 0 & 0 & \nu \\ 0 & \frac{1-\nu}{2} & \frac{1-\nu}{2} & 0 \\ 0 & \frac{1-\nu}{2} & \frac{1-\nu}{2} & 0 \\ \nu & 0 & 0 & 1 \end{bmatrix}, \tag{43}$$

$$\mathscr{D}_{\text{plane strain}} = \frac{E}{(1-2\nu)(1+\nu)}\begin{bmatrix} 1-\nu & 0 & 0 & \nu \\ 0 & 1/2-\nu & 1/2-\nu & 0 \\ 0 & 1/2-\nu & 1/2-\nu & 0 \\ \nu & 0 & 0 & 1-\nu \end{bmatrix}, \tag{44}$$

$$\mathscr{D}_{3D} = \begin{bmatrix} \lambda+2\mu & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & \lambda \\ 0 & \mu & 0 & \mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & \mu & 0 & \mu & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & \lambda+2\mu & 0 & 0 & 0 & \lambda \\ 0 & 0 & 0 & 0 & 0 & \mu & 0 & \mu & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu & 0 & \mu & 0 \\ \lambda & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & \lambda+2\mu \end{bmatrix}, \tag{45}$$

where $\lambda, \mu$ represent the Lamé constants, which are related to the Young's modulus $E$ and Poisson's ratio $\nu$:

$$\begin{cases} \lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)} \\ \mu = \dfrac{E}{2(1+\nu)} \end{cases}. \tag{46}$$

### 3.4 Construction of the first-order and higher-order operator energy functional

As a particle-based method, when using node integration the first-order and higher-order NOM suffer from a zero-energy mode [57, 58], which results in numerical instability. To eliminate the effect of the zero-energy mode, traditional PD and SPH introduce a penalty term to the force state [51]. Nevertheless, the approach described above is only applicable in the explicit time integration formulation. NOM employs operator energy functional for nonlocal gradients to achieve the linear field of the field and avoid numerical instabilities.

In first-order NOM, the operator energy functional at point $\mathbf{x}_i$ can be construct according to the first-order nonlocal operator $\nabla\otimes\mathbf{u}_i$, it can be expressed in discretization form as

$$\begin{aligned}\mathcal{F}_i^{hg} &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\int_{\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\Big([(u_j-u_i)-\nabla u_i\cdot\boldsymbol{\xi}_{ij}]^2+[(v_j-v_i)-\nabla v_i\cdot\boldsymbol{\xi}_{ij}]^2+\\&[(w_j-w_i)-\nabla w_i\cdot\boldsymbol{\xi}_{ij}]^2\Big)\mathrm{d}V_j\\&=\frac{p^{hg}}{2m_{\mathbf{K}_i}}\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\Big([(u_j-u_i)-\nabla u_i\cdot\boldsymbol{\xi}_{ij}]^2+[(v_j-v_i)-\nabla v_i\cdot\boldsymbol{\xi}_{ij}]^2+\\&[(w_j-w_i)-\nabla w_i\cdot\boldsymbol{\xi}_{ij}]^2\Big)\Delta V_j\\&=\frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\Big((u_j-u_i)^2+(v_j-v_i)^2+(w_j-w_i)^2\Big)\Delta V_j-\nabla\otimes\mathbf{u}_i:\nabla\otimes\mathbf{u}_i\cdot\mathbf{K}_i\right)\end{aligned}\tag{47}$$

where $\nabla u_i=[\frac{\partial u_i}{\partial x},\frac{\partial u_i}{\partial y},\frac{\partial u_i}{\partial z}]^T$, $\nabla\otimes\mathbf{u}_i=\begin{bmatrix}\frac{\partial u_i}{\partial x}&\frac{\partial u_i}{\partial y}&\frac{\partial u_i}{\partial z}\\\frac{\partial v_i}{\partial x}&\frac{\partial v_i}{\partial y}&\frac{\partial v_i}{\partial z}\\\frac{\partial w_i}{\partial x}&\frac{\partial w_i}{\partial y}&\frac{\partial w_i}{\partial z}\end{bmatrix}$, $\mathbf{K}_i=\begin{bmatrix}k_{11}&k_{12}&k_{13}\\k_{12}&k_{22}&k_{23}\\k_{13}&k_{23}&k_{33}\end{bmatrix}$.

It should be noted that the shape tensor $\mathbf{K}_i$ is involved in $\nabla\otimes\mathbf{u}_i:\nabla\otimes\mathbf{u}_i\cdot\mathbf{K}_i$ and the operator energy functional is valid in any dimension.

For convenience, we let $\Theta=\frac{1}{2}\nabla\otimes\mathbf{u}_i:\nabla\otimes\mathbf{u}_i\cdot\mathbf{K}_i$, $\Upsilon_{3d}=\frac{1}{2}[(u_j-u_i)^2+(v_j-v_i)^2+(w_j-w_i)^2]$.

To facilitate numerical implementation, we transform Eq. 47 into a detailed form and the point $\mathbf{x}_i$ displacement vector and first-order differential of displacement vector in 3D can be shown as

$$\begin{aligned}\Theta_{3d}=&\frac{1}{2}\begin{bmatrix}\frac{\partial u_i}{\partial x}&\frac{\partial u_i}{\partial y}&\frac{\partial u_i}{\partial z}\\\frac{\partial v_i}{\partial x}&\frac{\partial v_i}{\partial y}&\frac{\partial v_i}{\partial z}\\\frac{\partial w_i}{\partial x}&\frac{\partial w_i}{\partial y}&\frac{\partial w_i}{\partial z}\end{bmatrix}\left(\begin{bmatrix}\frac{\partial u_i}{\partial x}&\frac{\partial u_i}{\partial y}&\frac{\partial u_i}{\partial z}\\\frac{\partial v_i}{\partial x}&\frac{\partial v_i}{\partial y}&\frac{\partial v_i}{\partial z}\\\frac{\partial w_i}{\partial x}&\frac{\partial w_i}{\partial y}&\frac{\partial w_i}{\partial z}\end{bmatrix}\begin{bmatrix}k_{11}&k_{12}&k_{13}\\k_{21}&k_{22}&k_{23}\\k_{31}&k_{32}&k_{33}\end{bmatrix}\right)\\=&\frac{1}{2}\Big(\frac{\partial u_i}{\partial x}[k_{11}\frac{\partial u_i}{\partial x}+k_{12}\frac{\partial u_i}{\partial y}+k_{13}\frac{\partial u_i}{\partial z}]+\frac{\partial u_i}{\partial y}[k_{12}\frac{\partial u_i}{\partial x}+k_{22}\frac{\partial u_i}{\partial y}+k_{23}\frac{\partial u_i}{\partial z}]+\\&\frac{\partial u_i}{\partial z}[k_{13}\frac{\partial u_i}{\partial x}+k_{23}\frac{\partial u_i}{\partial y}+k_{33}\frac{\partial v_i}{\partial z}]+\frac{\partial v_i}{\partial x}[k_{11}\frac{\partial v_i}{\partial x}+k_{12}\frac{\partial v_i}{\partial y}+k_{13}\frac{\partial v_i}{\partial z}]+\\&\frac{\partial v_i}{\partial y}[k_{12}\frac{\partial v_i}{\partial x}+k_{22}\frac{\partial v_i}{\partial y}+k_{23}\frac{\partial v_i}{\partial z}]+\frac{\partial v_i}{\partial z}[k_{13}\frac{\partial v_i}{\partial x}+k_{23}\frac{\partial v_i}{\partial y}+k_{33}\frac{\partial v_i}{\partial z}]+\\&\frac{\partial w_i}{\partial x}[k_{11}\frac{\partial w_i}{\partial x}+k_{12}\frac{\partial w_i}{\partial y}+k_{13}\frac{\partial w_i}{\partial z}]+\frac{\partial w_i}{\partial y}[k_{12}\frac{\partial w_i}{\partial x}+k_{22}\frac{\partial w_i}{\partial y}+k_{23}\cdot\frac{\partial w_i}{\partial z}]+\\&\frac{\partial w_i}{\partial z}[k_{13}\frac{\partial w_i}{\partial x}+k_{23}\frac{\partial w_i}{\partial y}+k_{33}\frac{\partial w_i}{\partial z}]\Big)\end{aligned}\tag{48}$$

$$\mathscr{U}_i = (u_i, v_i, w_i, u_{j1}, v_{j1}, w_{j1} \cdots u_{jn}, v_{jn}, w_{jn})^T$$
$$\mathrm{d}\mathscr{U}_i = (\frac{\partial u_i}{\partial x}, \frac{\partial u_i}{\partial y}, \frac{\partial u_i}{\partial z}, \frac{\partial v_i}{\partial x}, \frac{\partial v_i}{\partial y}, \frac{\partial v_i}{\partial z}, \frac{\partial w_i}{\partial x}, \frac{\partial w_i}{\partial y}, \frac{\partial w_i}{\partial z})^T \tag{49}$$

In this paper, the special variation $\bar{\delta}\mathcal{F}$ , $\bar{\delta}^2\mathcal{F}$ and $\delta\mathcal{F}$ , $\delta^2\mathcal{F}$ are defined as

$$\bar{\delta}\mathcal{F} := \partial_{\mathrm{d}\mathscr{U}}\mathcal{F}, \bar{\delta}^2\mathcal{F} := \partial_{\mathrm{d}\mathscr{U}\mathrm{d}\mathscr{U}}\mathcal{F} \tag{50}$$

$$\delta\mathcal{F} := \partial_{\mathscr{U}}\mathcal{F}, \delta^2\mathcal{F} := \partial_{\mathscr{U}\mathscr{U}}\mathcal{F} \tag{51}$$

Hence, the following relationships can be derived as

$$\begin{aligned}\delta\Upsilon_{3d} &= \partial_{\mathscr{U}_{3d}}\Upsilon_{3d} = \partial_{\mathscr{U}_{3d}}\{\frac{1}{2}[(u_j-u_i)^2+(v_j-v_i)^2+(w_j-w_i)^2]\}\\ &= \left[(u_i-u_j),(v_i-v_j),(w_i-w_j),(-u_i+u_j),(-v_i+v_j),(-w_i+w_j)\right]\end{aligned} \tag{54}$$

$$\begin{aligned}\delta^2\Upsilon_{3d} &= \partial_{\mathscr{U}_{3d}\mathscr{U}_{3d}}\Upsilon_{3d}\\ &= \partial_{\mathscr{U}_{3d}}\{(u_i-u_j),(v_i-v_j),(w_i-w_j),(-u_i+u_j),\\ &\quad(-v_i+v_j),(-w_i+w_j)\}\\ &= \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0\\ 0 & 1 & 0 & 0 & -1 & 0\\ 0 & 0 & 1 & 0 & 0 & -1\\ -1 & 0 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0 & 0 & 1\end{bmatrix}\end{aligned} \tag{55}$$

$$\begin{aligned}\bar{\delta}\Theta_{3d} =&\partial_{\mathrm{d}\mathscr{U}_{3d}}\Theta_{3d}\\ =&\begin{bmatrix} [k_{11}\frac{\partial u_i}{\partial x}+k_{12}\frac{\partial u_i}{\partial y}+k_{13}\frac{\partial u_i}{\partial z}] & [k_{12}\frac{\partial u_i}{\partial x}+k_{22}\frac{\partial u_i}{\partial y}+k_{23}\frac{\partial u_i}{\partial z}] & [k_{13}\frac{\partial u_i}{\partial x}+k_{23}\frac{\partial u_i}{\partial y}+k_{33}\frac{\partial v_i}{\partial z}]\\ [k_{11}\frac{\partial v_i}{\partial x}+k_{12}\frac{\partial v_i}{\partial y}+k_{13}\frac{\partial v_i}{\partial z}] & [k_{12}\frac{\partial v_i}{\partial x}+k_{22}\frac{\partial v_i}{\partial y}+k_{23}\frac{\partial v_i}{\partial z}] & [k_{13}\frac{\partial v_i}{\partial x}+k_{23}\frac{\partial v_i}{\partial y}+k_{33}\frac{\partial v_i}{\partial z}]\\ [k_{11}\frac{\partial w_i}{\partial x}+k_{12}\frac{\partial w_i}{\partial y}+k_{13}\frac{\partial w_i}{\partial z}] & [k_{12}\frac{\partial w_i}{\partial x}+k_{22}\frac{\partial w_i}{\partial y}+k_{23}\frac{\partial w_i}{\partial z}] & [k_{13}\frac{\partial w_i}{\partial x}+k_{23}\frac{\partial w_i}{\partial y}+k_{33}\frac{\partial w_i}{\partial z}]\end{bmatrix}\end{aligned} \tag{52}$$

$$\begin{aligned}\bar{\delta}^2\Theta_{3d} =&\partial_{\mathrm{d}\mathscr{U}_{3d}\mathrm{d}\mathscr{U}_{3d}}\Theta_{3d}\\ =&\partial_{\mathrm{d}\mathscr{U}_{3d}}\bar{\delta}\Theta_{3d}\\ =&\begin{bmatrix} k_{11} & k_{12} & k_{13} & 0 & 0 & 0 & 0 & 0 & 0\\ k_{12} & k_{22} & k_{23} & 0 & 0 & 0 & 0 & 0 & 0\\ k_{13} & k_{23} & k_{33} & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & k_{11} & k_{12} & k_{13} & 0 & 0 & 0\\ 0 & 0 & 0 & k_{12} & k_{22} & k_{23} & 0 & 0 & 0\\ 0 & 0 & 0 & k_{13} & k_{23} & k_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & k_{11} & k_{12} & k_{13}\\ 0 & 0 & 0 & 0 & 0 & 0 & k_{12} & k_{22} & k_{23}\\ 0 & 0 & 0 & 0 & 0 & 0 & k_{13} & k_{23} & k_{33}\end{bmatrix}\end{aligned} \tag{53}$$

According to Eqs. 53 and 55, Eq. 47 can be rewritten as

$$\begin{aligned}\mathcal{F}_i^{hg} &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\sum_{j\in\mathcal{S}_i}\mathbb{w}(\boldsymbol{\xi}_{ij})\left((u_j-u_i)^2+(v_j-v_i)^2+(w_j-w_i)^2\right)\Delta V_j - \nabla\otimes\mathbf{u}_i : \nabla\otimes\mathbf{u}_i\cdot\mathbf{K}_i\right)\\ &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\mathscr{U}_i^T\begin{bmatrix}\sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix}\mathscr{U}_i - \mathrm{d}\mathscr{U}_i^T\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\mathrm{d}\mathscr{U}_i\right)\\ &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\mathscr{U}_i^T\begin{bmatrix}\sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix}\mathscr{U}_i - \mathscr{U}_i^T\mathscr{B}_i^T\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\mathscr{B}_i\mathscr{U}_i\right)\\ &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\mathscr{U}_i^T\left(\begin{bmatrix}\sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - \mathscr{B}_i^T\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\mathscr{B}_i\right)\mathscr{U}_i\end{aligned} \tag{56}$$

where $\mathbf{I}_j = \mathrm{w}(\boldsymbol{\xi}_{ij})\Delta V_j(1,1,1)\otimes(1,1,1)^T$.

$$\mathscr{K}_i^{hg} = \partial_{\mathscr{U}_i\mathscr{U}_i}\mathcal{F}_i^{hg} = \frac{p^{hg}}{m_{\mathbf{K}_i}}\left(\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - \mathscr{B}_i^T\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\mathscr{B}_i\right) \tag{57}$$

The global tangent stiffness matrix of operate energy functional, internal residual and tangent stiffness matrix of physical energy functional in support $\mathcal{S}_i$ can be obtained by

$$\overset{hg}{\mathscr{K}_i} = \sum_{j\in\mathcal{S}_i}\partial_{\mathscr{U}_i\mathscr{U}_i}\mathcal{F}_i^{hg}\Delta V_j = \sum_{j\in\mathcal{S}_i}\mathscr{K}_i^{hg}\Delta V_j \tag{58}$$

$$\mathscr{R}_i = \sum_{j\in\mathcal{S}_i}\partial_{\mathscr{U}_i}\psi(\boldsymbol{\varepsilon})_i\Delta V_j = \sum_{j\in\mathcal{S}_i}\mathscr{B}_i^T\cdot\mathscr{D}\cdot\Delta\mathscr{U}_i\Delta V_j \tag{59}$$

$$\mathscr{K}_i = \sum_{j\in\mathcal{S}_i}\partial_{\mathscr{U}_i}\mathscr{R}_i\Delta V_j = \sum_{j\in\mathcal{S}_i}\mathscr{B}_i^T\cdot\mathscr{D}\cdot\mathscr{B}_i\Delta V_j \tag{60}$$

The derivation of the first-order hourglass tangent stiffness matrix and the global tangent stiffness matrix in 2D form is given in Appendix B.

The operator energy functional for higher-order NOM at point $\mathbf{x}_i$ can be construct according to the higher-order nonlocal operator $\partial_\alpha^{\mathbb{1}}\mathbf{u}_i$, it can be expressed as

$$\begin{aligned}
\mathcal{F}_{\alpha i}^{hg} &= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\big(\mathbf{u}_{ij} - (\boldsymbol{p}_j^{\mathbb{1}})^T\tilde{\partial}_\alpha^{\mathbb{1}}\mathbf{u}_i\big)^2\Delta V_j\\
&= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\big(\mathbf{u}_{ij}^2 + \partial_\alpha^{\mathbb{1}}\mathbf{u}_i^T\boldsymbol{p}_j^{\mathbb{1}}\otimes(\boldsymbol{p}_j^{\mathbb{1}})^T\partial_\alpha^{\mathbb{1}}\mathbf{u}_i\\
&\quad - 2\mathbf{u}_{ij}(\boldsymbol{p}_j^{\mathbb{1}})^T\partial_\alpha^{\mathbb{1}}\mathbf{u}_i\big)\Delta V_j\\
&= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\Big(\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\mathbf{u}_{ij}^2\Delta V_j + \partial_\alpha^{\mathbb{1}}\mathbf{u}_i^T\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{p}_j^{\mathbb{1}}\\
&\quad \otimes(\boldsymbol{p}_j^{\mathbb{1}})^T\Delta V_j\ \partial_\alpha^{\mathbb{1}}\mathbf{u}_i - 2\Delta\mathbf{u}_i^T\boldsymbol{p}_{\mathrm{w}i}^T\partial_\alpha^{\mathbb{1}}\mathbf{u}_i\Big)\\
&= \frac{p^{hg}}{2m_{\mathbf{K}_i}}\Big(\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\mathbf{u}_{ij}^2\Delta V_j + \tilde{\partial}_\alpha\mathbf{u}_i^T\mathbb{L}_i\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{p}_j^{\mathbb{1}}\\
&\quad \otimes(\boldsymbol{p}_j^{\mathbb{1}})^T\Delta V_j\ \mathbb{L}_i\partial_\alpha^{\mathbb{1}}\mathbf{u}_i - 2\Delta\mathbf{u}_i^T(\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}})^T\mathbb{L}_i\tilde{\partial}_\alpha\mathbf{u}_i\Big)
\end{aligned} \tag{62}$$

By submitting the Eq. 34 into the below Eq. 62 and it can be rewritten as

$$\begin{aligned}
\mathcal{F}_{\alpha i}^{hg} =& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\Big(\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\mathbf{u}_{ij}^2\Delta V_j - \Delta\mathbf{u}_i^T(\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}})^T\Big(\sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})\boldsymbol{p}_j^{\mathbb{1}}\otimes(\boldsymbol{p}_j^{\mathbb{1}})^T\Delta V_j\Big)^{-1}\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}}\Delta\mathbf{u}_i\Big)\\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\Big(\Delta\mathbf{u}_i^T\ \mathrm{diag}\left[\mathrm{w}(\boldsymbol{\xi}_{j_1})\Delta V_{j_1},\ldots,\mathrm{w}(\boldsymbol{\xi}_{j_{n_i}})\Delta V_{j_{n_i}}\right]\Delta\mathbf{u}_i - \Delta\mathbf{u}_i^T(\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}})^T\mathbf{K}_{\alpha i}\mathbb{L}_i\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}}\Delta\mathbf{u}_i\Big)\\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\Delta\mathbf{u}_i^T\left(\begin{bmatrix}\mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - (\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}})^T\mathbf{K}_{\alpha i}\mathbb{L}_i\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}}\right)\Delta\mathbf{u}_i\\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}(\mathbf{K}_{\alpha i}^{-1}\mathbf{p}_{\mathrm{w}i}^{\mathbb{1}}\mathbf{B}_{\alpha i}\mathbf{U}_i)^T\left(\begin{bmatrix}\mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - (\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}})^T\mathbf{K}_i\mathbb{L}_i\boldsymbol{p}_{\mathrm{w}i}^{\mathbb{1}}\right)\mathbf{K}_{\alpha i}^{-1}\mathbf{p}_{\mathrm{w}i}^{\mathbb{1}}\mathbf{B}_{\alpha i}\mathbf{U}_i
\end{aligned} \tag{63}$$

where $\mathbf{I}_j = \mathrm{w}(\boldsymbol{\xi}_{ij})\Delta V_j(1,1,1)\otimes(1,1,1)^T$.

Finally, the summation of 3D form for the first-order global tangent stiffness matrix and hourglass tangent stiffness matrix in support $\mathcal{S}_i$ can be obtained

$$\begin{aligned}
\mathbb{K}_i =& \mathscr{K}_i + \overset{hg}{\mathscr{K}_i}\\
=& \sum_{j\in\mathcal{S}_i}\Delta V_j\left(\mathscr{B}_i^T\cdot\mathscr{D}\cdot\mathscr{B}_i + \frac{p^{hg}}{m_{\mathbf{K}_i}}\left(\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - \mathscr{B}_i^T\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\mathscr{B}_i\right)\right)\\
=& \sum_{j\in\mathcal{S}_i}\Delta V_j\left(\mathscr{B}_i^T\left(\mathscr{D} - \frac{p^{hg}}{m_{\mathbf{K}_i}}\begin{bmatrix}\mathbf{K}_i & 0 & 0\\ 0 & \mathbf{K}_i & 0\\ 0 & 0 & \mathbf{K}_i\end{bmatrix}\right)\mathscr{B}_i + \frac{p^{hg}}{m_{\mathbf{K}_i}}\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n}\\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \vdots & \mathbf{0} & \ddots & \mathbf{0}\\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix}\right)
\end{aligned} \tag{61}$$

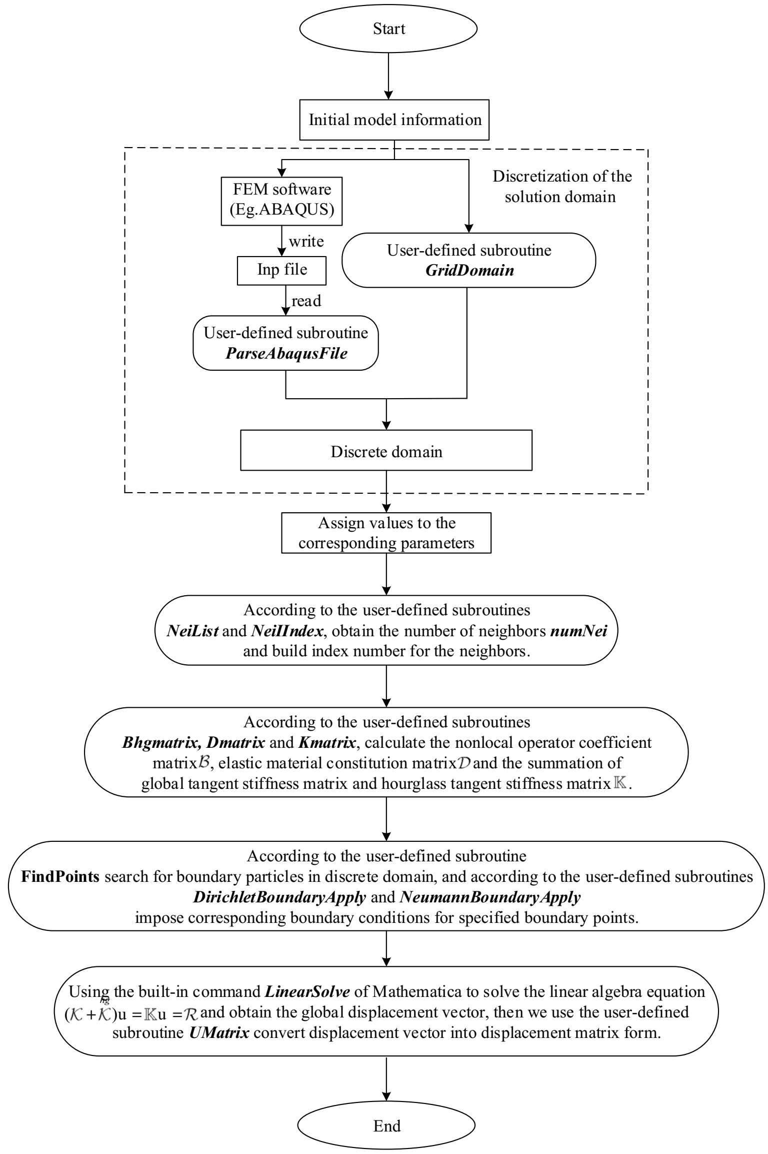


**Fig. 2** NOM implementation procedure

The first and second derivative $\mathcal{F}^{hg}_{\alpha i}$ yield the higher-order residual and the tangent stiffness matrix of operator energy functional and can be expressed as

$$\mathscr{R}^{hg}_{\alpha i} = \frac{\partial \mathcal{F}^{hg}_{\alpha i}}{\partial \mathbf{U}_i} = \frac{p^{hg}}{m_{\mathbf{K}_i}} (\mathbf{K}^{-1}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}\mathbf{B}_{\alpha i})^T \left( \begin{bmatrix} \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - (\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i})^T \mathbf{K}_{\alpha i}\mathbb{L}_i \boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}\right)\mathbf{K}^{-1}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}\mathbf{B}_{\alpha i}\mathbf{U}_i \tag{64}$$

$$\mathscr{K}^{hg}_{\alpha i} = \frac{\partial \mathscr{R}^{hg}_{\alpha i}}{\partial \mathbf{U}_i^T} = \frac{\partial^2 \mathcal{F}^{hg}_{\alpha i}}{\partial \mathbf{U}_i \partial \mathbf{U}_i^T} = \frac{p^{hg}}{m_{\mathbf{K}_i}} (\mathbf{K}^{-1}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}\mathbf{B}_{\alpha i})^T \left( \begin{bmatrix} \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n}\end{bmatrix} - (\boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i})^T \mathbf{K}_{\alpha i}\mathbb{L}_i \boldsymbol{p}^{\mathbb{l}}_{\mathbb{w}i}\right)\mathbf{K}^{-1}_{\alpha i}\mathbf{p}^{\mathbb{l}}_{\mathbb{w}i}\mathbf{B}_{\alpha i} \tag{65}$$

The higher-order global tangent stiffness matrix of operator energy functional in support $\mathcal{S}_i$ can be computed by

$$\overset{hg}{\mathscr{K}}_{\alpha i} = \sum_{j\in\mathcal{S}_i} \partial_{\mathbf{U}_i\mathbf{U}_i}\mathcal{F}^{hg}_{\alpha i}\Delta V_j = \sum_{j\in\mathcal{S}_i} \mathscr{K}^{hg}_{\alpha i}\Delta V_j \tag{66}$$

Finally, the summation of the higher-order global tangent stiffness matrix and hourglass tangent stiffness matrix in support $\mathcal{S}_i$ can be obtained

$$\mathbb{K}_{\alpha i} = \mathscr{K}_{\alpha i} + \overset{hg}{\mathscr{K}}_{\alpha i}$$
$$= \sum_{j \in S_i} \left( \mathbf{B}_{\alpha i}{}^{T} \cdot \mathscr{D} \cdot \mathbf{B}_{\alpha i} + \frac{p^{hg}}{m_{\mathbf{K}_i}} (\mathbf{K}_{\alpha i}^{-1} \mathbf{p}_{\mathbb{w}i}^{\mathbb{l}} \mathbf{B}_{\alpha i})^T \left( \begin{bmatrix} \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix} - (\boldsymbol{p}_{\mathbb{w}i}^{\mathbb{l}})^T \mathbf{K}_{\alpha i} \mathbb{L}_i \boldsymbol{p}_{\mathbb{w}i}^{\mathbb{l}} \right) \mathbf{K}_{\alpha i}^{-1} \mathbf{p}_{\mathbb{w}i}^{\mathbb{l}} \mathbf{B}_{\alpha i} \right) \Delta V_j \tag{67}$$

### 3.5 Numerical implementation with an open-source code

The numerical implementation of the first-order and higher-order NOM are summarized as the following steps, and a flow chart of the NOM implementation procedure is depicted in Fig. 2.

**Step 1. Discretization of the solution domain.**

Consider the solution domain to be a discrete domain consisting of discrete points of varying sizes and shapes that are linked to one another. There are two methods to achieve the discretization of the solution domain. For the first method, a user-defined subroutine ***GridDomain*** is customized, which can discretize the solution domain evenly. This method is mainly used for the discretization of the rule solution domain. The second method can be achieved as follows: initially, the model is created using finite element software (such as ABAQUS), then the model is divided into grids of different sizes according to the characteristics of the model and export the model information into a ***inp*** format file. The model mesh and node information can be read through a user-defined subroutine ***ParseAbaqusFile***, therefore the discrete solution domain with different densities can be achieved.

The Mathematica code for discretization of the solution domain is shown below. In the user-defined subroutine ***GridDomain***, where ***xmin***, ***xmax*** are the minimum and maximum values of solution domain and ***dx*** is the spacing between points.

```
GridDomain[xmin_, xmax_, dx_] := Block[{ndim = Length[xmin], nx = Ceiling[(xmax - xmin) / dx], dx2, res},
   dx2 = (xmax - xmin) / nx;
   If[ndim == 2, res = Table[{i, j}, {i, 0, nx[[1]]}, {j, 0, nx[[2]]}]];
   If[ndim == 3, res = Table[{i, j, k}, {i, 0, nx[[1]]}, {j, 0, nx[[2]]}, {k, 0, nx[[3]]}]];
   res = Flatten[res];
   res = ArrayReshape[res, {Length[res] / ndim, ndim}];
   Do[res[[All, i]] *= dx2[[i]], {i, ndim}]; Do[res[[i]] += xmin, {i, Length[res]}]; res];
ParseAbaqusFile[file_String] :=
  Module[{i, j, k, node, element, setList, oneSet, starList, starLine, tlineStart, tlineEnd, tline,
    strings, String2List, i1, i2, len, pair, ei, j1},
   If[FileExtension[file] ≠ "inp", Print["Error,the file should be Abaqus keyword file .inp"]; Return[];];
   strings = ReadList[file, String];
   If[strings == $Failed, Print["Error,file not found"];
    Return[]];
   i = Dimensions[strings][[1]];
   node = {};
   element = {};
   setList = {};
   starList = {};
   Do[If[StringStartsQ[strings[[j]], "*"] && ! StringStartsQ[strings[[j]], {"**", "$"}], AppendTo[starList, j]],
    {j, i}];
   String2List[str_String] := Module[{sl, s2}, s2 = StringReplace[str, {"E", "e"} → "*^"];
     sl = StringSplit[s2, {",", " "}];
     DeleteCases[ToExpression[sl], Null]];
   Do[starLine = ToLowerCase[strings[[starList[[j]]]]];
    tlineStart = starList[[j]] + 1; tlineEnd = starList[[j + 1]] - 1;
    Which[StringStartsQ[starLine, "*node"], Do[tline = strings[[k]];
      If[! StringStartsQ[tline, "*"], AppendTo[node, String2List[tline]]];, {k, tlineStart, tlineEnd}]
      (**), StringStartsQ[starLine, "*element"], Do[tline = strings[[k]];
      If[! StringStartsQ[tline, "*"], AppendTo[element, String2List[tline]]];, {k, tlineStart, tlineEnd}]
      (**), StringStartsQ[starLine, "*set"], oneSet = {};
     Do[tline = strings[[k]];
      If[! StringStartsQ[tline, "*"], AppendTo[oneSet, String2List[tline]]];, {k, tlineStart, tlineEnd}];
     AppendTo[setList, Flatten[oneSet]] (**)];, {j, 1, Length[starList] - 1}];
   len = Length[node]; i1 = node[[1, 1]]; i2 = node[[len, 1]];
   If[i1 == 1 && i2 == len, Return[{node[[All, 2 ;; -1]], element[[All, 2 ;; -1]], setList}]];
   pair = ConstantArray[0, i2];
   Do[pair[[node[[i, 1]]]] = i, {i, len}];
   len = Length[element];
   Do[ei = element[[i]];
    Do[j1 = ei[[j]];
     ei[[j]] = pair[[j1]];, {j, 2, Length[ei]}];
    element[[i]] = ei;, {i, len}];
   {node[[All, 2 ;; -1]], element[[All, 2 ;; -1]], setList}];
```

**Step 2. Definition of the problem and solution domain.** Determine the overall attributes and geometric boundary conditions of the solution domain based on the real case, and assign values to the corresponding parameters.

**Step 3. Specify and search for the number of neighbors for each point in support and build index numbers for the specified neighbors.**

In this step, a user-defined subroutine ***NeiList*** is customized by the built-in command ***Nearest*** of Mathematica to find the required number of neighbors, and the process of building index number for the specified neighbor points can be achieved by the user-defined subroutine ***NeiIIndex***. The Mathematica code for finding the specified area points and specified number neighbor points in the solution domain is shown below, where ***coord*** is the coordinate of the points, ***numNei*** represents the number of specified neighbors and ***ndim*** denotes the dimension of coordinate.

```
NeiList = Nearest[coord → Automatic, coord, numNei + 1];
NeiIIndex[NeiList_, ndim_] := Module[{n2, n3}, n2 = ConstantArray[0, ndim Length[NeiList]];
   n3 = ndim (NeiList - 1);
   Do[n2[[i ;; -1 ;; ndim]] = n3 + i, {i, ndim}];
   n2];
```

**Step 4. Establish the nonlocal operator coefficient matrix, hourglass tangent stiffness matrix, and the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix in first-order and higher-order form.**

① First-order nonlocal operator method.

As shown in below Mathematica code, three user-defined subroutines ***BHgmatrix***, ***Dmatrix*** and ***Kmatrix*** are customized to calculate the nonlocal operator coefficient matrix $\mathscr{B}$, hourglass tangent stiffness matrix $\mathscr{K}_{hg}$, elastic material constitution matrix $\mathscr{D}$ and the summation of hourglass tangent stiffness matrix and tangent stiffness matrix $\mathbb{K}$. In the user-defined subroutine ***BHgmatrix***, initially the basic forms $\mathscr{B}_i$, $\mathscr{K}^{hg}_i$ and shape tensor $\mathbf{K}_i$ at point $\mathbf{x}_i$ are constructed, which are named as ***bmatrix***, ***hgmatrix*** and ***kmatrix***, respectively. Subsequently by Eq. 7, shape tensor can be calculated; by the equation $\mathbf{I}_j = \mathbb{w}(\xi_{ij})V_j(1,1,1) \otimes (1,1,1)^T$ and Eq. 56, the matrix $\mathbf{I}_j$ can be calculated and the $\mathscr{K}^{hg}_i$ matrix can be assembled. Similarly, by equation $\mathbf{R}_j = \mathbb{w}(\xi_{ij})\xi_{ij}^T V_j \cdot \left[\sum_{j\in\mathcal{S}_i} \mathbb{w}(\xi_{ij})\xi_{ij} \otimes \xi_{ij}\Delta V_j\right]^{-1}$ and Eq. 15, the intermediate variables $\mathbf{R}_j$ can be obtained and the $\mathscr{B}_i$ matrix is assembled. In the user-defined subroutine ***Dmatrix***, by Eqs. 43, 44 and 45, the linear elastic solid elastic matrix $\mathscr{D}$ for plane stress, plane strain and 3D conditions can be obtained. Where type=1,2,3 represent the three constitutive models mentioned above. In the user-defined subroutine ***Kmatrix***, by Eq. 67, the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix $\mathbb{K}$ can be established. In the below Mathematica code, where ***WeiF*** represents the weight function, ***voli*** represents the volume of each point, ***Es, mu*** represents the Young's Modulus and Poisson's ratio, respectively.

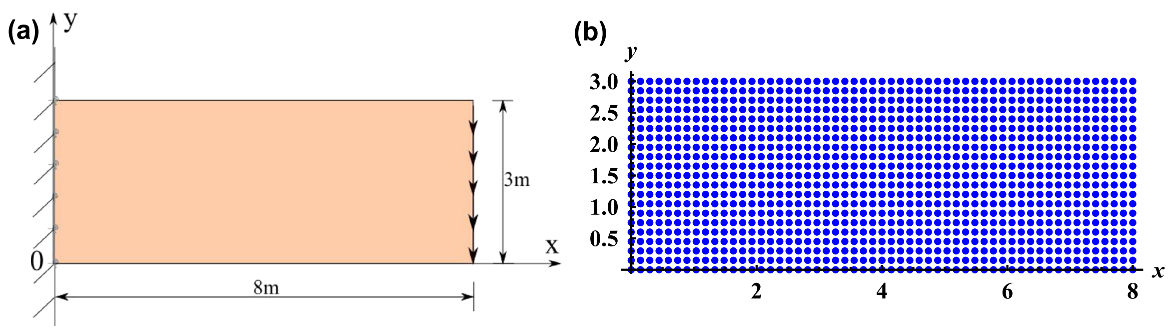


**Fig. 3** **a** Set up of the cantilever beam; **b** the discretization of the cantilever beam($\Delta x$=$\mathcal{H}$/20)

```
BHgmatrix[coord_List, WeiF_, voli_List] := Block[{num = Length[coord],
    ndim = Length[coord[[1]]], bmatrix, kmatrix, hgmatrix, invk, xi, vw, ri, Rj, i1, i2, I1, I2},
   bmatrix = ConstantArray[0., {ndim ndim, ndim num}];
   hgmatrix = ConstantArray[0., {ndim num, ndim num}];
   kmatrix = ConstantArray[0., {ndim, ndim}];
   I2 = ConstantArray[0., {ndim ndim, ndim ndim}];
   Do[xi = coord[[j]] - coord[[1]];
    ri = Sqrt[xi.xi];
    vw = voli[[j]] × WeiF[ri];
    kmatrix += vw TensorProduct[xi, xi];
    i1 = ndim (j - 1) + 1; i2 = ndim j;
    I1 = vw IdentityMatrix[ndim];
    hgmatrix[[1 ;; ndim, 1 ;; ndim]] += I1; hgmatrix[[1 ;; ndim, i1 ;; i2]] = -I1;
    hgmatrix[[i1 ;; i2, 1 ;; ndim]] = -I1;
    hgmatrix[[i1 ;; i2, i1 ;; i2]] = I1, {j, 2, num}];
   invk = Inverse[kmatrix];
   Do[xi = coord[[j]] - coord[[1]]; ri = Sqrt[xi.xi]; Rj = voli[[j]] × WeiF[ri] invk.xi;
    Do[bmatrix[[ndim (k - 1) + 1 ;; k ndim, k]] -= Rj;
     bmatrix[[ndim (k - 1) + 1 ;; k ndim, ndim (j - 1) + k]] = Rj, {k, ndim}];
    , {j, 2, num}];
   Do[I2[[j ndim + 1 ;; (j + 1) ndim, j ndim + 1 ;; (j + 1) ndim]] = kmatrix, {j, 0, ndim - 1}];
   hgmatrix -= bmatrix^T.I2.bmatrix;
   hgmatrix /= Tr[kmatrix];
   {bmatrix, hgmatrix}
  ];
Dmatrix[Es_, mu_, type_: 3] :=
  Block[{DMatrix, ndim, L1, L2},
   If[type == 1,
    Return[Es / (1. - mu^2) {{1., 0., 0., mu}, {0., (1. - mu) / 2, (1. - mu) / 2, 0.}, {0., (1. - mu) / 2, (
       {mu, 0., 0., 1.}}]];
   If[type == 2,
    Return[Es / (1. - 2 mu) / (1. + mu) {{1. - mu, 0., 0., mu}, {0., 0.5 - mu, 0.5 - mu, 0.}, {0., 0.5 - mu
       {mu, 0., 0., 1. - mu}}]];
   L1 = Es mu / ((1. + mu) (1. - 2 mu)); L2 = Es / 2. / (1. + mu);
   {{L1 + 2 L2, 0., 0., 0., L1, 0., 0., 0., L1}, {0., L2, 0., L2, 0., 0., 0., 0., 0.}, {0., 0., L2, 0., 0
    {0., L2, 0., L2, 0., 0., 0., 0., 0.}, {L1, 0., 0., 0., L1 + 2 L2, 0., 0., 0., L1}, {0., 0., 0., 0., 0
    {0., 0., L2, 0., 0., 0., L2, 0., 0.}, {0., 0., 0., 0., 0., L2, 0., L2, 0.}, {L1, 0., 0., 0., L1, 0.,
  ];
Kmatrix[voli_, bmatrix_, hgmatrix_, Es_, mu_, etype_: 3, pen_: 0.2] :=
  voli (pen Es hgmatrix + bmatrix^T.Dmatrix[Es, mu, etype].bmatrix);
```

**Table 1** The statistical results of L2-norm and error for $u_{x max}$ and $u_{y max}$ at different discretizations

| Npoint | $\Delta x$ | L2-norm | $\frac{u_{x_{max}}^{NOM}}{u_{x_{max}}^{exact}} - 1$ | $\frac{u_{y_{max}}^{NOM}}{u_{y_{max}}^{exact}} - 1$ |
|---|---|---|---|---|
| 308 | $\mathcal{H}/10$ | 0.0118 | -0.2738 | -0.2679 |
| 1155 | $\mathcal{H}/20$ | 0.0096 | -0.0731 | -0.0779 |
| 2511 | $\mathcal{H}/30$ | 0.0082 | -0.0289 | -0.0308 |
| 4428 | $\mathcal{H}/40$ | 0.0073 | -0.0209 | -0.0245 |
| 6885 | $\mathcal{H}/50$ | 0.0068 | 0.0150 | 0.0128 |

② Higher-order nonlocal operator method.

The main codes for higher-order NOM are presented in Appendix C, which includes constructing the global higher-order nonlocal operator coefficient matrix $\mathbf{B}_\alpha$, global hourglass tangent stiffness matrix $\overset{hg}{\mathscr{K}}_\alpha$ and the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix $\mathbb{K}_\alpha$. Initially, a user-defined subroutine ***MultiIndex*** is customized to construct multi-index notation in

**Step 5. Search boundary points in discrete domain and impose corresponding boundary conditions based on actual problems.**

A user-defined subroutine ***FindPoints*** is customize for searching boundary points. When using user-defined subroutine ***FindPoints***, it's need to call another customized subroutine ***LessThancoord*** at the same time. Subsequently, to employ Dirichlet and Neumann boundary conditions to the boundary points, two user-defined subroutines ***DirichletBoundaryApply*** and ***NeumannBoundaryApply*** are customized. Where the penalty method is used in the user-defined subroutine ***DirichletBoundaryApply***. The related Mathematica codes are shown as below, where ***Ksp*** represents the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix $\mathbb{K}$, ***Rsp*** represents the global internal residual, ***pd*** represents the index of the points of application of specified displacement, ***pc*** represents the penalty coefficient and ***pf*** represents the index of the points of application of specified force.

```
FindPoints[coord_List, xmin_List, xmax_List, show_: False] :=
  Module[{ilist = {}, y, r, ndim = Length[coord[[1]]], len = Length[coord]},
   Do[If[LessThancoord[coord[[i]], xmax] && LessThancoord[xmin, coord[[i]]], AppendTo[ilist, i]], {i, len}];
   If[show, If[ Length[coord[[1]]] == 2, Print[ListPlot[coord[[ilist]], DataRange → Automatic]]];
    If[ Length[coord[[1]]] == 3, Print[ListPointPlot3D[coord[[ilist]], DataRange → Full]]];
   ilist];
LessThancoord[coord1_List, coord2_List] := Module[{p = True}, Do[If[coord1[[i]] > coord2[[i]], p = False;
     Break[]], {i, Length[coord]}];
   Return[p]];
DirichletBoundaryApply[Ksp_, Rsp_, pd_, pc_, scale_: 1.0] := Module[{pi, tk, tk2}, tk = Diagonal[Ksp] // Normal;
   tk2 = tk;
   tk2[[pd[[1]]]] *= (pc + 1.);
   tk2 -= tk;
   Ksp += SparseDiag[tk2];
   Do[pi = pd⟦1, i⟧; Rsp⟦pi⟧ = tk2[[pi]] pd⟦2, i⟧ scale, {i, Length[pd⟦1⟧]}];];
SparseDiag[a_List] := SparseArray[{{i_, i_} :→ a[[i]]}, Length[a]];
NeumannBoundaryApply[Rsp_, pf_, scale_: 1.0] := Module[{}, Rsp⟦pf⟦1⟧⟧ += pf⟦2⟧ scale;];
```

Eq. 21. where ***d*** denotes the dimension, ***sum*** denotes the maximal order of derivatives. Then, the customized user-defined subroutine ***GFD0CoeffHalf*** can obtain the factors of multi-index, polynomials and partial derivatives in designated dimensions with maximal higher-order derivatives in Eq. 23. For constructing the elastic material constitution matrix, according to Eqs. 36–38, the user-defined subroutine ***FuncRK*** is compiled to obtain elastic matrix $\mathscr{D}$. For higher-order global hourglass tangent stiffness matrix $\overset{hg}{\mathscr{K}}_\alpha$ and the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix $\mathbb{K}_\alpha$, according to Eqs. 33, 34 and 67, two main user-defined subroutines ***NOMPwKhg*** and ***NOMRK*** are customized to achieve the establishment of corresponding matrix.

**Step 6. Solve the global displacement in the discrete domain.**

According to step 1 to step 5, the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix $\mathbb{K}$ and global residual vector $\mathscr{R}$ can be calculated. And the global displacement matrix can be calculated by solving the following linear algebra Eq. 68.

$$(\mathscr{K} + \overset{hg}{\mathscr{K}})\mathbf{u} = \mathbb{K}\mathbf{u} = \mathscr{R} \tag{68}$$

we employ the built-in command ***LinearSolve*** of Mathematica to solve the global displacement vector, then we use the user-defined subroutine ***UMatrix*** convert displacement vector into displacement matrix form. The related Mathematica codes are shown below.

```
u = LinearSolve[Ksp, Rsp];
um = UMatrix[u, ndim];
UMatrix[u_List, udim_Integer] := Module[{m}, If[udim == 1, m = u;
    Return[m]];
   m = ConstantArray[0., {Length[u] / udim, udim}];
   Do[m[[i]] = u[[udim (i - 1) + 1 ;; udim i]], {i, Length[m]}]; m];
```

## 4 Numerical examples

Four numerical examples in 2D or 3D are presented to validate the first-order and higher-order NOM in this section. The numerical results are compared with the analytical solution or that by FEM software to verify the feasibility.

### 4.1 A cantilever beam under shear load

In this section, a 2D cantilever beam loaded at the end with shear load is considered. Figure 3a depicts geometry and boundary conditions of the cantilever beam. The beam with $\mathcal{H} = 3$m in height, $\mathcal{L} = 8$m in length. The cantilever beam parameters are: $E = 6 \times 10^3$MPa, $\nu = 0.33$. The shear load is parabolic distributed. The beam is discretized into 308,1155,2511,4428,6885 points respectively, which corresponding to $\Delta x \in \{\mathcal{H}/10, \mathcal{H}/20, \mathcal{H}/30, \mathcal{H}/40, \mathcal{H}/50\}$. Where $\Delta x$ denotes the spacing of the points, the discretization of $\Delta x = 20$ is shown in Fig. 3b. Plane stress conditions are considered in this section. The analytical solution refers to literature [59, 60].

$$u_x = \frac{\mathcal{P}(y - \mathcal{H}/2)}{6E\mathcal{I}}\left[(6\mathcal{L} - 3x)x + (2 + \nu)((y - \mathcal{H}/2)^2 - \frac{\mathcal{H}^2}{4})\right] \tag{69}$$

$$u_y = -\frac{\mathcal{P}}{6E\mathcal{I}}\left[3\nu(y - \mathcal{H}/2)^2(\mathcal{L} - x) + (4 + 5\nu)\frac{\mathcal{H}^2 x}{4} + (3\mathcal{L} - x)x^2\right] \tag{70}$$

$$\begin{aligned}\sigma_{xx}(x, y) = \frac{\mathcal{P}(\mathcal{L} - x)(y - \mathcal{H}/2)}{\mathcal{I}}, \sigma_{yy}(x, y) = 0, \tau_{xy}(x, y)\\ = -\frac{\mathcal{P}}{2\mathcal{I}}\left(\frac{\mathcal{H}^2}{4} - (y - \mathcal{H}/2)^2\right),\end{aligned} \tag{71}$$

where $x \in [0, \mathcal{L}], y \in [0, \mathcal{H}]$, $\mathcal{P}$=-5 kN, $\mathcal{I} = \frac{\mathcal{H}^3}{12}$. The discretized cantilever beam on displacement boundary are constrained using the accurate displacements according to Eqs. 69 and 70, as well as the force boundary according to Eq. 71.

To obtain the displacement of each point, we need to solve the Eq. 68, which can be shown in detail as Eq. 72. In this work, we use the penalty approach to apply Dirichlet boundary conditions. This method can be achieved through the following steps:

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & \cdots & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & \cdots & \cdots & k_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{i1} & k_{i2} & \cdots & k_{ii} & \cdots & k_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{n1} & k_{n2} & \cdots & \cdots & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_i \\ \vdots \\ R_n \end{bmatrix} \tag{72}$$

When the displacement of point i $u_i = \overline{u_i}$, we modify the i-th equation as follows, multiply it's diagonal element $K_{ii}$ by a penalty factor $\eta$(In computations, $\eta$ is set to $10^{10}$.) and replace $R_i$ with $\eta k_{ii}\overline{u_i}$ to obtain:

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & \cdots & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & \cdots & \cdots & k_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{i1} & k_{i2} & \cdots & \eta k_{ii} & \cdots & k_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{n1} & k_{n2} & \cdots & \cdots & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ \eta k_{ii}\overline{u_i} \\ \vdots \\ R_n \end{bmatrix} \tag{73}$$

The modified i-th equation can be expressed as:

$$k_{i1}u_1 + k_{i2}u_2 + \cdots + \eta k_{ii}u_i + \cdots + k_{in}u_n = \eta k_{ii}\overline{u_i} \tag{74}$$

Since $\eta k_{ii} \gg k_{ij}(i \neq j)$, the $\eta k_{ii}u_i$ term at the left end of the equation is much larger than the other terms, so it can be approximated $\eta k_{ii}u_i \approx \eta k_{ii}\overline{u_i}$. Then we can obtain $u_i \approx \overline{u_i}$. According to the Eq. 73, we can obtain the displacement of each point. It should be noted that this method is suitable for any given displacement of points, as the order of the formula and the displacement order of the points remain unchanged during the solution process.

The number of target point's neighbors and the radius of support size in NOM can be very flexible and unlimited. However, a large number of neighbors is expensive in the k-nearest neighbor's search. Based on our numerical experience, eight points closest to the target point are selected to construct the target point's support domain. The difference between the numerical result and analytical solution is measured by the L2-norm, which is calculated by

$$\|\mathbf{u}\|_{L2} = \sqrt{\frac{\sum_j(\mathbf{u}_j - \mathbf{u}_j^{exact}) \cdot (\mathbf{u}_j - \mathbf{u}_j^{exact})\Delta V_j}{\sum_j \mathbf{u}_j^{exact} \cdot \mathbf{u}_j^{exact}\Delta V_j}}. \tag{75}$$

Table.1 shows the statistical results of the L2-norm for displacement at various discretizations, Fig. 4 depicts the displacement for discretization at $\Delta x = \mathcal{H}/50$. The displacement cloud diagram of the cantilever beam for discretization at $\Delta x = \mathcal{H}/50$ using first-order NOM numerical results and analytical results can be seen in Fig. 5. As shown in Table 1, Figs. 4 and 5, good agreements can be seen between the first-order NOM numerical results and analytical results. It shows very close variations in L2-norm and the displacement errors in different directions.

### 4.2 Problem of an infinite plate with a hole in tension

Consider a 2D infinite plate with a circular hole, as shown in Fig. 6. The plate's length is $\mathcal{L}$ and there is a small circular hole with a radius of $\rho = \mathbb{a}$ in the plate. Because the plate's thickness is substantially smaller than its length, it can be considered a plane stress problem. The force boundary condition of $\mathcal{P} = 1$ MPa. Based on the symmetry of the structure and load, one-quarter of the model is used for analysis.

When the plate's width is substantially more than the radius of the small hole, we can obtain the stresses analytical solution [61] in polar coordinate system according to classical elasticity theory, and it can be expressed as

$$\begin{aligned}
\sigma_\rho &= \frac{\mathcal{P}}{2}\left(1-\frac{\mathbb{a}^2}{\rho^2}\right) + \frac{\mathcal{P}}{2}\cos 2\theta\left(1-\frac{\mathbb{a}^2}{\rho^2}\right)\left(1-3\frac{\mathbb{a}^2}{\rho^2}\right),\\
\sigma_\theta &= \frac{\mathcal{P}}{2}\left(1+\frac{\mathbb{a}^2}{\rho^2}\right) - \frac{\mathcal{P}}{2}\cos 2\theta\left(1+3\frac{\mathbb{a}^4}{\rho^4}\right)\\
\tau_{\rho\theta} &= \tau_{\theta\rho} = -\frac{\mathcal{P}}{2}\sin 2\theta\left(1-\frac{\mathbb{a}^2}{\rho^2}\right)\left(1+3\frac{\mathbb{a}^2}{\rho^2}\right).
\end{aligned} \tag{76}$$

The highest hoop normal stress is achieved at the hole's edge, as illustrated in Fig. 6. When $\rho = \mathbb{a}$, $\theta = \frac{\pi}{2}(\frac{3\pi}{2})$, $\sigma_\theta$ attains the maximum value of three times the uniformly distributed stress $\mathcal{P}$, $(\sigma_\theta)_{\max} = 3\mathcal{P}$, and $\sigma_\theta$ sharply approaches to $\mathcal{P}$ as it moves away to the edge. We characterize the stress concentration phenomena using the stress concentration factor $K$ in this case. We solve stress concentration problems by the first-order NOM, plane stress condition is considered. The numerical results were compared with ABAQUS standard and analytical solutions. The parameters for the plate include $E = 3 \times 10^4$MPa, $\nu = 0.3$, and the radius of the hole is $\rho = 1$ m. To facilitate the application of force boundary conditions, we converted the Eq. 76 from polar coordinate system to Cartesian coordinate system, it can be shown as

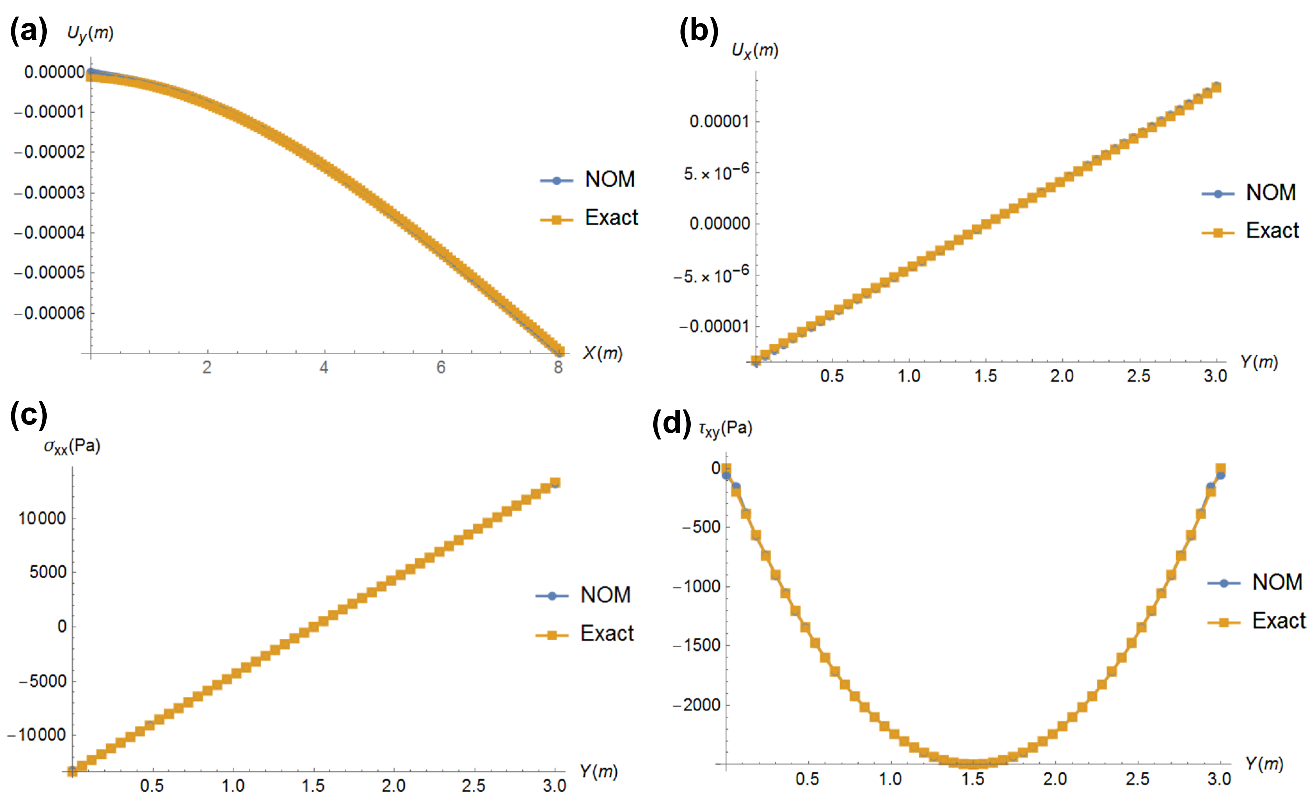


**Fig. 4** The displacement and stress for discretization at $\Delta x = \mathcal{H}/50$. **a** Displacement of points y=$\mathcal{H}$/2 in the *x* direction; **b** displacement of points x=$\mathcal{L}$/2 in the *y* direction; **c** stress of points x=$\mathcal{L}$/2 in the *y* direction; **d** stress of points x=$\mathcal{L}$/2 in the *y* direction

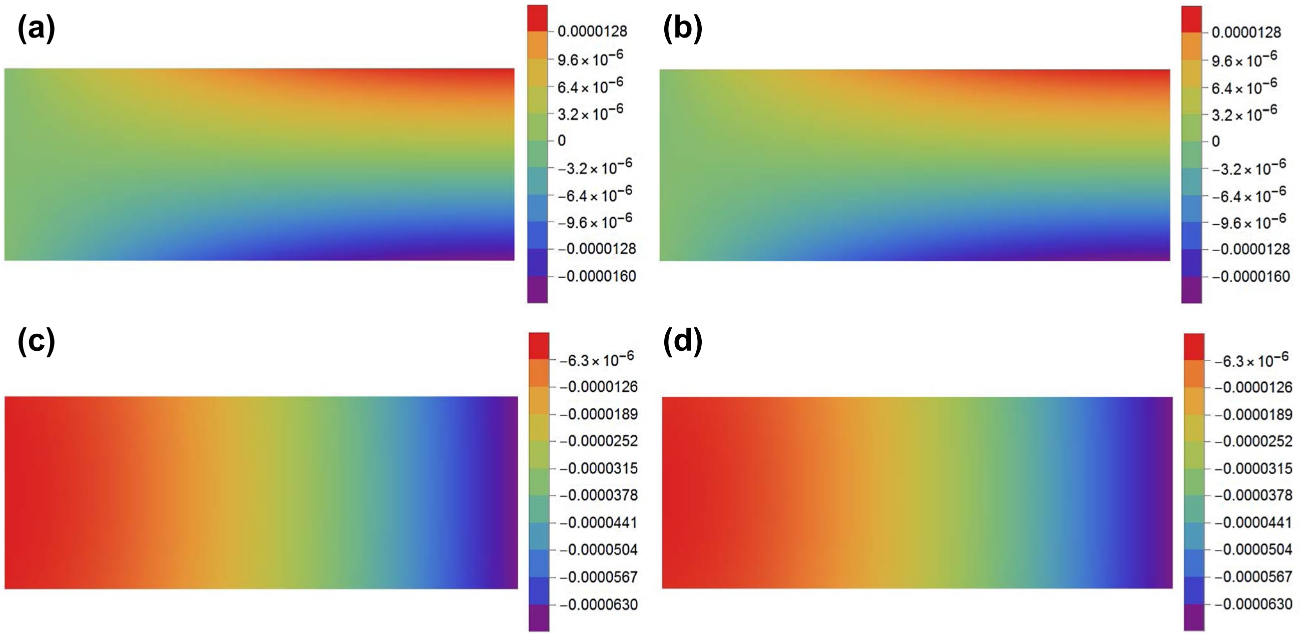


**Fig. 5** The displacement cloud diagram of the cantilever beam for discretization at $\Delta x = \mathcal{H}/50$. **a** Displacement in y direction by first-order NOM numerical results; **b** displacement in y direction analytical results; **c** displacement in x direction by first-order NOM numerical results; **d** displacement in x direction analytical results

$$
\begin{aligned}
\sigma_{xx}(\rho,\theta) &= \mathcal{P} - \mathcal{P}\frac{\mathbb{a}^2}{\rho^2}\left(\frac{3\cos 2\theta}{2} + \cos 4\theta\right) + \mathcal{P}\frac{3\mathbb{a}^4}{2\rho^4}\cos 4\theta \\
\sigma_{yy}(\rho,\theta) &= -\mathcal{P}\frac{\mathbb{a}^2}{\rho^2}\left(\frac{\cos 2\theta}{2} - \cos 4\theta\right) - \mathcal{P}\frac{3\mathbb{a}^4}{2\rho^4}\cos 4\theta \\
\tau_{xy}(\rho,\theta) &= -\mathcal{P}\frac{\mathbb{a}^2}{\rho^2}\left(\frac{\sin 2\theta}{2} + \sin 4\theta\right) + \mathcal{P}\frac{3\mathbb{a}^4}{2\rho^4}\sin 4\theta
\end{aligned}
\tag{77}
$$

To validate the feasibility of the first-order NOM, we compare the numerical simulation results with the analytical solutions. The analytical solution of the displacement under plane stress conditions can be expressed as

$$
\begin{aligned}
u_x(\rho,\theta) &= \frac{\mathcal{P}\mathbb{a}}{8\mu}\left(\frac{\rho}{\mathbb{a}}(\kappa+1)\cos\theta + \frac{2\mathbb{a}}{\rho}((1+\kappa)\cos\theta + \cos 3\theta) - \frac{2\mathbb{a}^3}{\rho^3}\cos 3\theta\right) \\
u_y(\rho,\theta) &= \frac{\mathcal{P}\mathbb{a}}{8\mu}\left(\frac{\rho}{\mathbb{a}}(\kappa-3)\sin\theta + \frac{2\mathbb{a}}{\rho}((1-\kappa)\sin\theta + \sin 3\theta) - \frac{2\mathbb{a}^3}{\rho^3}\sin 3\theta\right)
\end{aligned}
\tag{78}
$$

where $\mu = \frac{E}{2(1+\nu)}$, and $\kappa = \frac{3-\nu}{1+\nu}$.

For the displacement boundary conditions, we set $u_x = 0, u_y = 0$ at left and bottom boundaries by penalty method, meanwhile, for the force boundary conditions, we applied the traction force at right and top boundaries computed by Eq. 77.

To obtain a good discrete result, initially, we build the model and meshes in ABAQUS, then export it as an "model.inp" file, which includes the element and point coordinate information. We read the exported "model.inp" file in the environment of Mathematica. We can calculate the coordinates of each point in the model, the area of each element, and assign parameters such as area and force to the relevant points of the element according to the core principle. So as to realize the discreteness of the unit in the environment of Mathematica. It should be noted that only the nodes are used and no interaction between elements. The Mathematica software reads the "model.inp" model ($\mathcal{L}/\mathbb{a} = 5$) and discretization of the model as shown in Fig. 7. In this study, we use the built-in command ***Nearest*** of Mathematica to find the required number of neighbor points, and eight neighbors are selected.

We fix the diameter of the hole to 2m and change the $\mathcal{L}$. Four cases of relative size of plate width and hole radius with $\mathcal{L}/\mathbb{a}$=5,7,9,11 and four cases with total 525,2050,4575,8100 nodes are investigated. The plate is discretized according to ABAQUS mesh elements and CPS3 elements are adopted in ABAQUS [62] to calculate the reference results. At first we test the maximum value of stress in the x-direction $(\sigma_{xx})_{max}$, according to the formulation

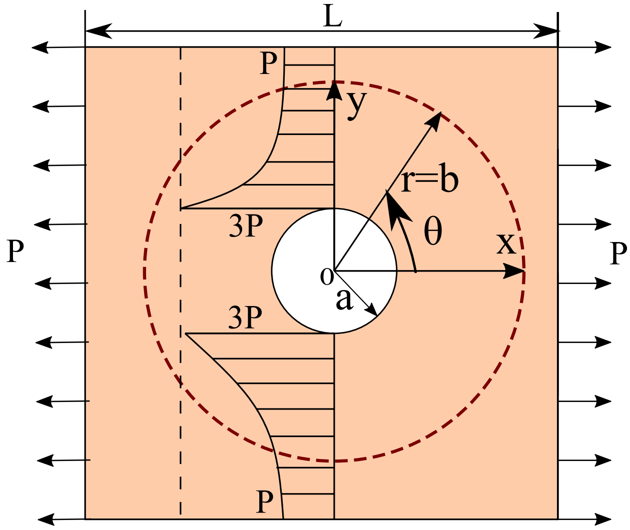


**Fig. 6** Setup of infinite plate with a circular hole

$K = \frac{(\sigma_{xx})_{max}}{\mathcal{P}}$, we can obtain the corresponding $K$ and compared it to the analytic solutions. According to the first-order NOM, ABAQUS standard, and analytical solution, the statistical results of the stress concentration factor $K$, as shown in Table 2.

The displacement and stress distribution of the model($\mathcal{L}/\mathbb{a} = 5$) on polar coordinate system $\rho = 2$ at the case of with total 4575 and 8100 nodes using first-order NOM are compared to the analytic solutions, as seen in Figs. 8 and 9. The displacement's L2 norm computed by Eq. 75 and stress field at four different discretization cases are shown in Table 3. These characteristics agree well with the theoretical analysis of this problem mentioned above, and it proved that the first-order NOM established in this paper has good applicability.

### 4.3 Crack propagation problem by NOM with phase-field modeling

Phase field fracture modeling [63–68] shows great advantages for modeling complex crack propagation, and its based on the Γ-convergence of the approximations of free discontinuity problems [69]. In this section, we use NOM to establish implicit phase field modeling and a phase field $\phi(\boldsymbol{x}, t) \in [0, 1]$ is introduced to approximate the fracture surface, Γ. Where $\phi(\boldsymbol{x}, t) = 1$ denotes the complete damage of the material and $\phi(\boldsymbol{x}, t) = 0$ denotes the material is undamaged.

According to literature [70], the density of the crack surface per unit volume $\gamma(\phi)$ and the surface energy in the solid due to the formation of crack can be shown as

$$\gamma(\phi, \nabla\phi) = \frac{\phi^2}{2l_0} + \frac{l_0}{2}\nabla\phi \cdot \nabla\phi, \tag{79}$$

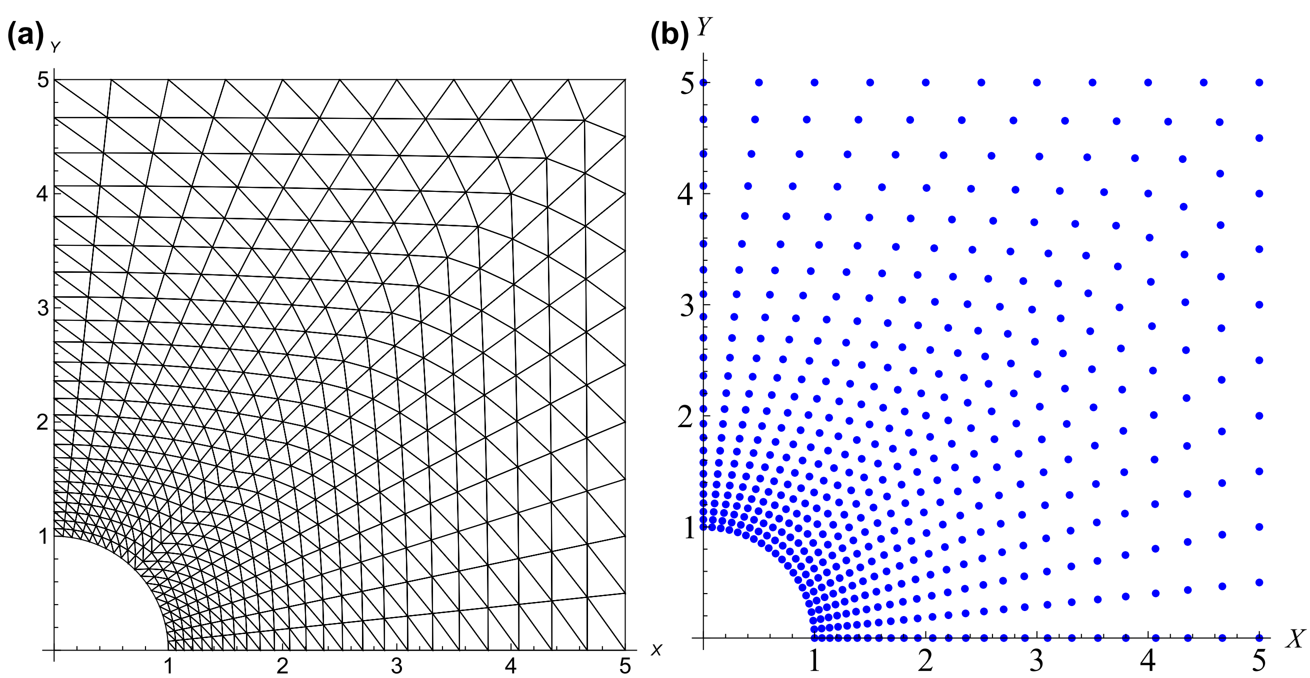


**Fig. 7** **a** The “model.inp” model shown in mathematica; **b** discretization of the model

**Table 2** The statistical results of the stress concentration factor $K$ at different $\mathcal{L}/a$ ratios, where 525 and 2050 denotes the total number of points in the model

| $\mathcal{L}/\mathrm{a}$ | Analytical solution | ABAQUS(525) | NOM (525) | ABAQUS(2050) | NOM(2050) |
|---|---|---|---|---|---|
| 5 | 3.000 | 3.330 | 3.361 | 3.350 | 3.371 |
| 7 | 3.000 | 3.164 | 3.176 | 3.178 | 3.203 |
| 9 | 3.000 | 3.100 | 3.109 | 3.115 | 3.126 |
| 11 | 3.000 | 3.064 | 3.078 | 3.087 | 3.096 |

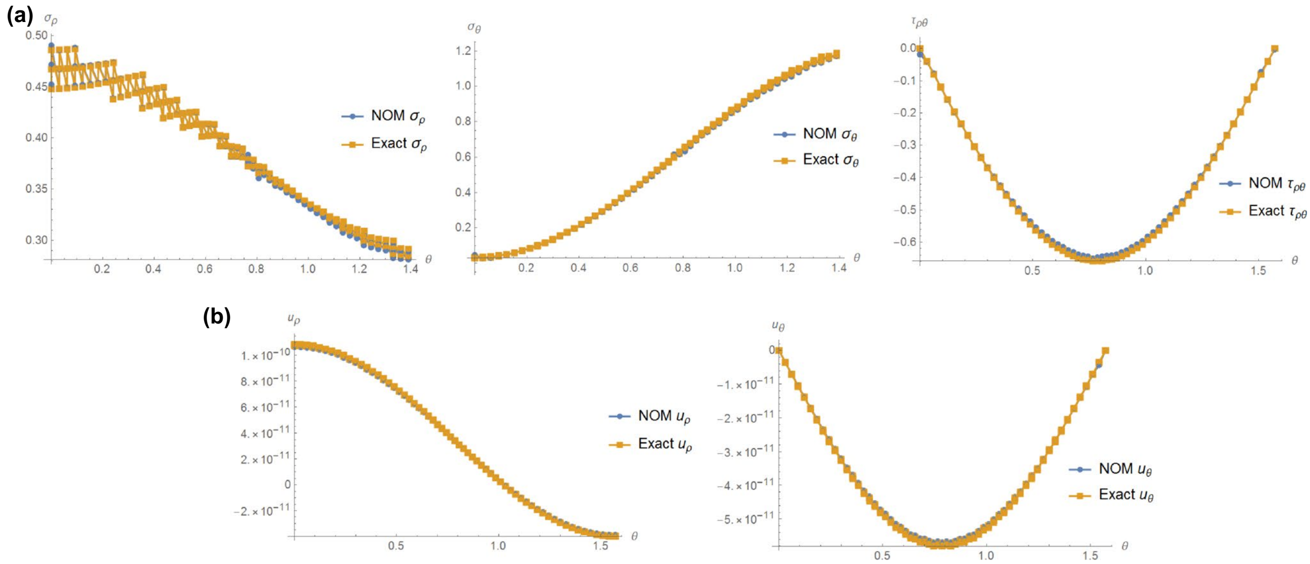


**Fig. 8** Analytical and numerical results of the stress and displacement. **a** The results of the stress (Nnodes = 4575); **b** the results of the displacement (Nnodes = 4575)

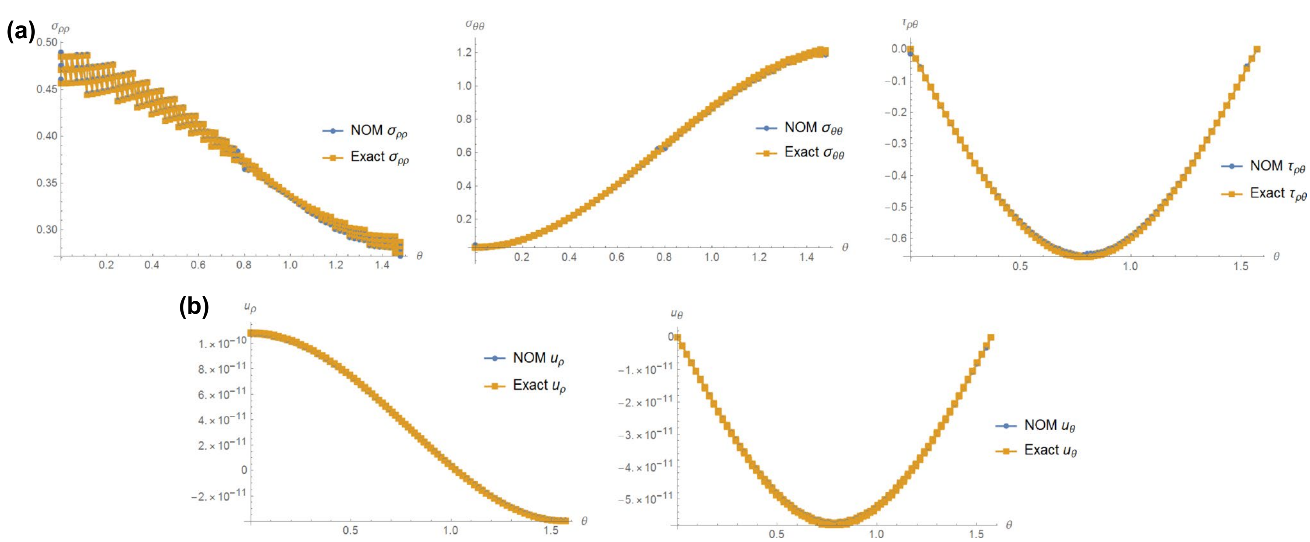


**Fig. 9** Analytical and numerical results of the stress and displacement. **a** The results of the stress (Nnodes=8100); **b** the results of the displacement (Nnodes=8100)

**Table 3** The L2 norm of the displacement and stress at four different discretization cases

| Nnodes | L2 norm of displacement | L2 norm of stress |
|---|---|---|
| 525 | 0.0581 | 0.0908 |
| 2050 | 0.0276 | 0.0478 |
| 4575 | 0.0163 | 0.0428 |
| 8100 | 0.0118 | 0.0376 |

$$\int_\Gamma G_c \mathrm{d}A \approx \int_\Omega G_c\left(\frac{\phi^2}{2l_0} + \frac{l_0}{2}\nabla\phi\cdot\nabla\phi\right)\mathrm{d}V, \tag{80}$$

where $G_c$ represent critical energy release rate, $l_0$ is a length scale factor that governs the phase field's transition area, hence it can reflect the width of the crack, When $l_0 \to 0$, the $\Gamma$-limit recovers the sharp discontinuous interface of the crack.

To guarantee that the crack is only driven by tensile load in phase field simulation, the elastic energy must be decomposed into tensile and compressive components [71]. The strain tensor $\boldsymbol{\varepsilon}$ can be decomposed as follows by using Eigen-decomposition

$$\boldsymbol{\varepsilon}_+ = \sum_{a=1}^{d}\langle\varepsilon_a\rangle_+ \mathbf{n}_a\otimes\mathbf{n}_a, \boldsymbol{\varepsilon}_- = \sum_{a=1}^{d}\langle\varepsilon_a\rangle_- \mathbf{n}_a\otimes\mathbf{n}_a, \tag{81}$$

where the tensile and compressive parts of strain tensors are $\boldsymbol{\varepsilon}_+$ and $\boldsymbol{\varepsilon}_-$, respectively; $\langle x\rangle_\pm := (x \pm |x|)/2$; the principal strain is $\varepsilon_a$, and the principal strain direction is $\mathbf{n}_a$.

The elastic energy density $\psi_\varepsilon(\boldsymbol{\varepsilon})$ and Cauchy stress tensor $\boldsymbol{\sigma}$ for the tensile and compressive parts are represented by the decomposed strain tensor as following

$$\begin{aligned}
&\psi_e(\boldsymbol{\varepsilon},\phi) = [(1-\phi)^2+\kappa_0]\psi_e^+(\boldsymbol{\varepsilon}) + \psi_e^-(\boldsymbol{\varepsilon})\\
&\psi_e^+(\boldsymbol{\varepsilon}) = \frac{\lambda}{2}\langle tr(\boldsymbol{\varepsilon})\rangle_+^2 + \mu\langle tr(\boldsymbol{\varepsilon}_+)^2\rangle\\
&\psi_e^-(\boldsymbol{\varepsilon},\phi) = \frac{\lambda}{2}\langle tr(\boldsymbol{\varepsilon})\rangle_-^2 + \mu\langle tr(\boldsymbol{\varepsilon}_-)^2\rangle
\end{aligned} \tag{82}$$

$$\begin{aligned}
\boldsymbol{\sigma} = \frac{\partial\psi_e(\boldsymbol{\varepsilon},\phi)}{\partial\boldsymbol{\varepsilon}} &= [(1-\phi)^2+\kappa_0]\frac{\partial\psi_e^+(\boldsymbol{\varepsilon})}{\partial\boldsymbol{\varepsilon}} + \frac{\partial\psi_e^-(\boldsymbol{\varepsilon})}{\partial\boldsymbol{\varepsilon}}\\
&= [(1-\phi)^2+\kappa_0][\lambda\langle tr(\boldsymbol{\varepsilon})\rangle_+\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}_+] + \lambda\langle tr(\boldsymbol{\varepsilon})\rangle_-\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}_-,
\end{aligned} \tag{83}$$

where $\kappa_0$ ($\kappa_0 > 0$ and $\kappa_0 \ll 1$) is a tiny positive factor that prevents the positive component of the elastic energy density from degrading completely, $\lambda, \mu$ are the Lamé parameters. $\boldsymbol{I}$ donates identity tensor.

The overall potential functional is calculated as the sum of the phase field approximations for the fracture energy, elastic energy, body energy, and potential energy caused by the surface load, it can be shown as

$$\begin{aligned}
\Pi(\boldsymbol{u},\phi) = &\underbrace{\int_\Omega \frac{G_c}{2}\left(\frac{\phi^2}{l_0} + l_0\nabla\phi\cdot\nabla\phi\right)\mathrm{d}\Omega + \int_\Omega [(1-\phi)^2+\kappa_0]\psi_e^+(\boldsymbol{\varepsilon}) + \psi_e^-(\boldsymbol{\varepsilon})\mathrm{d}\Omega}_{W_{int}}\\
&\underbrace{-\int_\Omega \mathbf{b}\cdot\boldsymbol{u}\,\mathrm{d}\Omega - \int_{\partial\Omega_f}\boldsymbol{f}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma}_{W_{ext}}
\end{aligned} \tag{84}$$

where $\mathbf{b}$ denotes the body force, $\boldsymbol{u}$ donates the displacement, and $\boldsymbol{f}$ donates the surface loads at the boundary, $W_{\mathrm{int}}$ donates internal energy, $W_{\mathrm{ext}}$ donates external work.

The Lagrange energy functional can be expressed as

$$L = \frac{1}{2}\int_\Omega \rho\dot{\boldsymbol{u}}\cdot\dot{\boldsymbol{u}}\,\mathrm{d}\Omega - \Pi(\boldsymbol{u},\phi) \tag{85}$$

According to Hamilton's principle, the first variation of the Lagrange energy functional $L$ should be 0, namely $\delta L = 0$, hence we can obtain the following phase-field governing equations and Neumann boundary conditions

$$\begin{aligned}
[(1-\phi)^2+\kappa_0]\nabla\cdot\boldsymbol{\sigma} + \mathbf{b} &= \rho\ddot{\boldsymbol{u}} && \text{in} \quad \Omega\\
[2l_0\psi_e^+ + G_c]\phi - G_c l_0^2\nabla^2\phi &= 2l_0\psi_e^+ && \text{in} \quad \Omega\\
\boldsymbol{\sigma}\cdot\boldsymbol{n}^* &= \boldsymbol{f} && \text{on} \quad \partial\Omega_f\\
\nabla\phi\cdot\boldsymbol{n}^* &= 0 && \text{on} \quad \partial\Omega_f
\end{aligned} \tag{86}$$

where $\boldsymbol{n}^*$ is the outward-pointing normal vector for boundary $\partial\Omega_f$.

To maintain the phase field that increases monotonically [70], the local history field of strain $H(\boldsymbol{x},t)$ is introduced and it can be shown as

$$H(\boldsymbol{x},t) = \max_{s\in[0,t]}\psi_e^+(\boldsymbol{\varepsilon}(\boldsymbol{x},s)) \tag{87}$$

we use $H(\boldsymbol{x},t)$ replace the $\psi_e^+$ in the Eq. 86, then we can obtain

$$\begin{aligned}
[(1-\phi)^2+\kappa_0]\nabla\cdot\boldsymbol{\sigma} + \mathbf{b} &= \rho\ddot{\boldsymbol{u}},\\
[2l_0 H + G_c]\phi - G_c l_0^2\nabla^2\phi &= 2l_0 H,\\
\boldsymbol{\sigma}\cdot\boldsymbol{n}^* &= \boldsymbol{f},\\
\nabla\phi\cdot\boldsymbol{n}^* &= 0
\end{aligned} \tag{88}$$

Since the implicit NOM is based on the governing equations weak form, and the weak forms of Eq. 88 are formulated as follows

$$\int_\Omega G_c(\frac{1}{l_0}\phi\delta\phi + l_0\nabla\phi\cdot\nabla\delta\phi)\mathrm{d}\Omega + \int_\Omega -2(1-\phi)\delta\phi H\mathrm{d}\Omega = 0$$
$$\int_\Omega -\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\mathrm{d}\Omega + \int_\Omega \mathbf{b}\cdot\delta\boldsymbol{u}\mathrm{d}\Omega + \int_{\partial\Omega_f}\boldsymbol{f}\cdot\delta\boldsymbol{u}\mathrm{d}\Gamma = 0 \tag{89}$$

The continuous support $\mathcal{S}_i$ is expressed in NOM by

$$N_i = \{j_1, .., j_k, .., j_n\} \tag{90}$$

where $j_1, .., j_k, .., j_n$ denote the global indices of point $i$'s neighbors.

The nonlocal operator for displacement $\tilde{\nabla}\otimes\boldsymbol{u}_i$ and phase field $\tilde{\nabla}\phi_i$ in discrete form can be rewritten as

$$\tilde{\nabla}\otimes\boldsymbol{u}_i \simeq \sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})(\boldsymbol{u}_j-\boldsymbol{u}_i)\otimes(\mathbf{K}_i^{-1}\boldsymbol{\xi}_{ij})\Delta V_j = \mathscr{B}_i^{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}_i \tag{91}$$

$$\tilde{\nabla}\phi_i \simeq \sum_{j\in\mathcal{S}_i}\mathrm{w}(\boldsymbol{\xi}_{ij})(\phi_j-\phi_i)(\mathbf{K}_i^{-1}\boldsymbol{\xi}_{ij})\Delta V_j = \mathscr{B}_i^{\phi}\cdot\tilde{\phi}_i \tag{92}$$

where $\simeq$ denotes discretization, $\tilde{\boldsymbol{u}}_i(=\mathbf{u}_i, \mathbf{u}_{j_1}, .., \mathbf{u}_{j_k}, .., \mathbf{u}_{j_n})$ is all the variations of the displacement in support $\mathcal{S}_i$, $\tilde{\phi}_i(=\phi_i, \phi_{j_1}, .., \phi_{j_k}, .., \phi_{j_n})$ represents all variations of the phase field in support $\mathcal{S}_i$. $\Delta V_j$ is the volume of neighbor point $j$, $\mathscr{B}_i^{\boldsymbol{u}}$ and $\mathscr{B}_i^{\phi}$ are the the coefficient matrices, as shown in Eqs. 15 and 16.

The research was based on a library of linear elastic material. According to the above expressions, Eq. 89 can be shown as

$$\sum_{j\in\mathcal{S}_i}\big(G_c l_0[\mathscr{B}_i^{\phi}]^T\cdot\mathscr{B}_i^{\phi}\cdot\tilde{\phi}_i + [G_c\frac{1}{l_0}\tilde{\phi}_i - 2(N-\tilde{\phi}_i)H]\big)\Delta V_j = 0$$
$$\sum_{j\in\mathcal{S}_i} -[\mathscr{B}_i^{\boldsymbol{u}}]^T\mathbf{D}_{ijkl}\{\varepsilon_i\}\Delta V_j + \sum_{j\in\mathcal{S}_i}\mathbf{b}_i N\Delta V_j + \int_{\partial\Omega_f}\boldsymbol{f}_i N\Delta S_j = 0 \tag{93}$$

where $\boldsymbol{N} = [1, 0, \cdots, 0, 0]^T$ is a $(n_i+1)$-dimensional column vector.

We can obtain the point $\mathbf{x}_i$ residual of the overall systems

$$\begin{aligned}\boldsymbol{R}_i^{\boldsymbol{u}} &= \boldsymbol{F}_i^{\boldsymbol{u},ext} - \boldsymbol{F}_i^{\boldsymbol{u},int}\\ &= \underbrace{\sum_{j\in\mathcal{S}_i}\boldsymbol{f}_i\Delta S_j + \sum_{j\in\mathcal{S}_i}\mathbf{b}_i\Delta V_j}_{\boldsymbol{F}_i^{\boldsymbol{u},ext}} - \sum_{j\in\mathcal{S}_i}[\mathscr{B}_i^{\boldsymbol{u}}]^T\mathbf{D}_{ijkl}\{\boldsymbol{\varepsilon}_i\}\Delta V_j\\ R_i^{\phi} &= -F_i^{\phi,int}\\ &= \sum_{j\in\mathcal{S}_i}\big(G_c l_0[\mathscr{B}_i^{\phi}]^T\cdot\mathscr{B}_i^{\phi}\cdot\tilde{\phi}_i + [G_c\frac{1}{l_0}\tilde{\phi}_i - 2(\boldsymbol{N}-\tilde{\phi}_i)H]\big)\Delta V_j\end{aligned} \tag{94}$$

where $\boldsymbol{F}_i^{\boldsymbol{u},\mathrm{int}}$, $\boldsymbol{F}_i^{\boldsymbol{u},\mathrm{ext}}$ are internal force, external force, respectively, which correspond to the displacement field.

Whereas $F_i^{\phi,\mathrm{int}}$ is the internal force corresponding to the phase field. The corresponding tangent stiffness matrices can be obtained based on the internal forces:

$$\mathscr{K}_i^{\boldsymbol{u}} = \frac{\partial\boldsymbol{F}_i^{\boldsymbol{u},int}}{\partial\tilde{\boldsymbol{u}}_i} = \sum_{j\in\mathcal{S}_i}[\mathscr{B}_i^{\boldsymbol{u}}]^T\mathbf{D}_{ijkl}\mathscr{B}_i^{\boldsymbol{u}}\Delta V_j \tag{95}$$

$$\mathscr{K}_i^{\phi} = \frac{\partial F_i^{\phi,int}}{\partial\tilde{\phi}_i} = \sum_{j\in\mathcal{S}_i}\big([\mathscr{B}_i^{\phi}]^T G_c l_0\mathscr{B}_i^{\phi} + (\frac{G_c}{l_0}+2H)\boldsymbol{N}\otimes\boldsymbol{N}^T\big)\Delta V_j \tag{96}$$

where $\mathbf{D}_{ijkl}$ is a elasticity matrix, and it can be obtained using Eigen-decomposition algorithm for fourth-order isotropic tensor [70]

$$\begin{aligned}\mathbf{D}_{ijkl} &= \frac{\partial\{\boldsymbol{\sigma}_{ij}\}}{\partial\{\boldsymbol{\varepsilon}_{kl}\}} = \frac{\partial}{\partial\{\boldsymbol{\varepsilon}_{kl}\}}\big([(1-\phi)^2+\kappa_0]\boldsymbol{\sigma}_{ij}^+ + \boldsymbol{\sigma}_{ij}^-\big)\\ &= [(1-\phi)^2+\kappa_0]\mathbf{D}_{ijkl}^+ + \mathbf{D}_{ijkl}^-\end{aligned} \tag{97}$$

The stress based on the Eigen-decomposition method can be expressed as

$$\begin{aligned}\boldsymbol{\sigma}^{\pm} = \lambda\langle\varepsilon_1+\varepsilon_2+\varepsilon_3\rangle_{\pm}I_{3\times3} + 2\mu(\langle\varepsilon_1\rangle_{\pm}\boldsymbol{n}_1\otimes\boldsymbol{n}_1\\ + \langle\varepsilon_2\rangle_{\pm}\boldsymbol{n}_2\otimes\boldsymbol{n}_2 + \langle\varepsilon_3\rangle_{\pm}\boldsymbol{n}_3\otimes\boldsymbol{n}_3)\end{aligned} \tag{98}$$

where $\lambda$, $\mu$ are the Lamé constants, $I_{3\times3}$ is the identity matrix, $\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3$ are Eigenvectors for principal strains $\varepsilon_1, \varepsilon_2, \varepsilon_3$ of $\boldsymbol{\varepsilon}(=\sum_{i=1}^3\varepsilon_i\boldsymbol{n}_i\otimes\boldsymbol{n}_i)$ (Fig. 10).

Then Eq. 110 can be shown as

$$\mathscr{K}_i^{\boldsymbol{u}} = \sum_{j\in\mathcal{S}_i}[\mathscr{B}_i^{\boldsymbol{u}}]^T\big([(1-\phi)^2+\kappa_0]\mathbf{D}_{ijkl}^+ + \mathbf{D}_{ijkl}^-\big)\mathscr{B}_i^{\boldsymbol{u}}\Delta V_j \tag{99}$$

However, Eq. 89 suffers from the zero-energy mode, the penalty force from the nonlocal operator functional should be added, then the summation of global tangent stiffness matrix and hourglass tangent stiffness matrix in displacement and phase field as shown in following

$$\mathbb{K}_i^{\boldsymbol{u}} = \sum_{j\in\mathcal{S}_i}\Big([\mathscr{B}_i^{\boldsymbol{u}}]^T\big([(1-\phi)^2+\kappa_0]\mathbf{D}_{ijkl}^+ + \mathbf{D}_{ijkl}^-\big)\mathscr{B}_i^{\boldsymbol{u}} + \overset{hg}{\mathscr{K}_i^{\boldsymbol{u}}}\Big)\Delta V_j \tag{100}$$

$$\mathbb{K}_i^{\phi} = \sum_{j\in\mathcal{S}_i}\Big([\mathscr{B}_i^{\phi}]^T G_c l_0\mathscr{B}_i^{\phi} + (\frac{G_c}{l_0}+2H)\boldsymbol{N}\otimes\boldsymbol{N}^T + \overset{hg}{\mathscr{K}_i^{\phi}}\Big)\Delta V_j \tag{101}$$

In this case, we consider a single notched square plate subjected to static tension loading. Figure 12 depicts the geometrical and boundary conditions of the plate. We consider the plate parameters as follows: $E = 2.1\times10^5$ MPa , $\nu = 0.3$ and the energy release rate $g_c = 2.7\times10^{-3}$ kN/mm. The

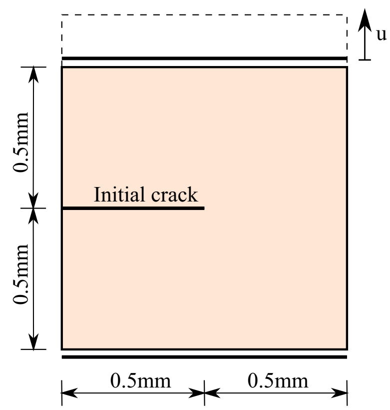


**Fig. 10** Geometry and boundary conditions

plane strain condition is considered in this test. We deleted the points closest to the initial crack to produce the initial crack and each point has eight neighbors to construct their own support domain. The single edge notched square plate is discretized into 40401 points. The phase-field length scale is set to $l_0 = 0.015$ mm. Due to the maximal order of partial derivatives in Eq. 89 is 1, the first-order NOM is employed. The penalty method is used to enforce the Dirichlet boundary conditions for phase and displacement fields. In this case, to prevent the singularity during computation, $\kappa_0 = 1 \times 10^{-6}$ is chosen. Due to the mutation property of crack propagation, in the staggered scheme, a small vertical displacement increment $\Delta u = 1 \times 10^{-5}$ mm is applied to the upper boundary of the plate and the bottom boundary is fixed all the time.

Figure 11 depicts the crack patterns in horizontal direction at various displacements for $l_0 = 0.015$. Figure 12 depicts the reaction force-displacement curves. As expected in the literature [72], when adopted staggered scheme, the rate of the crack evolves seems to be delayed in comparison to a completely monolithic scheme. In comparison with the results by Miehe [70, 72], the load curves obtained by the implicit nonlocal model align well with the reference result with the increase of the vertical displacement $\Delta u$. The crack tip still has a small bond force caused by the support and dual-support which leads to slight differences appear at post localization where FEM exhibits a sharper decay compared to the NOM.

### 4.4 3D gradient elasticity cantilever beam under shear load

To illustrate the feasibility of higher-order NOM, we expand it to handle the gradient elasticity beam issue. The isotropic elasticity gradient material's energy functional [73, 74] can be represented as

$$\mathbb{W} = \int_\Omega \frac{1}{2}\bar{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon} + \frac{\ell^2}{2}\nabla\bar{\boldsymbol{\sigma}} \,\vdots\, \nabla\boldsymbol{\varepsilon}\, \mathrm{d}\Omega - \int_\Omega \boldsymbol{b}\cdot\boldsymbol{u}\mathrm{d}\Omega - \int_{\partial\Omega} \mathbb{P}\cdot\boldsymbol{u}\mathrm{d}S - \int_{\partial\Omega} \mathbb{R}\cdot(\boldsymbol{n}^*\cdot\nabla\boldsymbol{u})\mathrm{d}S \tag{102}$$

where $\bar{\boldsymbol{\sigma}}$ represent the Cauchy-like stress tensor. $\ell$ represent the gradient material factor, $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + \boldsymbol{u}\nabla)$ represent the infinitesimal strain tensor. $\boldsymbol{b}$ represent body force density, whereas $\mathbb{P}$ and $\mathbb{R}$ represent the traction force and double traction force act on on $\partial\Omega$. $\boldsymbol{n}^*$ represents the outward-pointing normal vector for the boundary surface.

Based on the strain gradient elasticity theory [75], the variation of the strain energy and the external work can be written as

$$\begin{aligned}\delta\mathbb{W}^{int} &= \int_\Omega \left(\bar{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}(\delta\boldsymbol{u}) + \boldsymbol{\mu} \,\vdots\, \nabla\boldsymbol{\varepsilon}(\delta\boldsymbol{u})\right)\mathrm{d}\Omega \\ &= \int_\Omega \left(\bar{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}(\delta\boldsymbol{u}) + \frac{\ell^2}{2}\nabla\bar{\boldsymbol{\sigma}} \,\vdots\, \nabla\boldsymbol{\varepsilon}(\delta\boldsymbol{u})\right)\mathrm{d}\Omega\end{aligned} \tag{103}$$

$$\delta\mathbb{W}^{ext} = \int_\Omega \boldsymbol{b}\cdot\delta\boldsymbol{u}\mathrm{d}\Omega + \int_{\partial\Omega} \mathbb{P}\cdot\delta\boldsymbol{u}\mathrm{d}S + \int_{\partial\Omega} \mathbb{R}\cdot(\boldsymbol{n}^*\cdot\nabla\delta\boldsymbol{u})\mathrm{d}S \tag{104}$$

where $\boldsymbol{\mu}$ denotes the third rank double stress tensor.

In Eq. 103, the Cauchy-like quantities $\bar{\boldsymbol{\sigma}}$, work-conjugate to $\boldsymbol{\varepsilon}$, and the double stress $\boldsymbol{\mu}$, work-conjugate to $\nabla\boldsymbol{\varepsilon}$, are derived from the partial derivatives of the strain energy density with respect to $\boldsymbol{\varepsilon}$ and $\nabla\boldsymbol{\varepsilon}$ [74], respectively. Accordingly, we can obtain $\bar{\boldsymbol{\sigma}}$ and $\boldsymbol{\mu}$ as shown in follows

$$\bar{\boldsymbol{\sigma}} = \frac{\partial\mathbb{W}^{int}}{\partial\boldsymbol{\varepsilon}} = 2\mu\boldsymbol{\varepsilon} + \lambda\mathrm{tr}(\boldsymbol{\varepsilon})\mathbf{I},\ \boldsymbol{\mu} = \frac{\partial\mathbb{W}^{int}}{\partial\nabla\boldsymbol{\varepsilon}} = \ell^2\nabla\bar{\boldsymbol{\sigma}} \tag{105}$$

where $\mathrm{tr}\boldsymbol{\varepsilon} = \nabla\cdot\boldsymbol{u}$, $\mathbf{I}$ is the symmetric second rank unit tensor.

The Cauchy-like stress tensor $\bar{\boldsymbol{\sigma}}$ and strain tensor $\boldsymbol{\varepsilon}$ are coupled typically by Hooke's law [76], but the double stress tensor $\boldsymbol{\mu}$ is expressed on the basis of the Cauchy stress tensor, yielding the total stress tensor $\boldsymbol{\sigma}$ in the form

$$\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} - \ell^2\Delta\bar{\boldsymbol{\sigma}} \tag{106}$$

According to the principle of virtual work, $\delta\mathcal{F}_{int} - \delta\mathcal{F}_{ext} = 0$ is valid for $\delta\boldsymbol{u}$ in $\Omega$ and on surface $\partial\Omega$, which lead to the following governing equation and boundary conditions in terms of the Cauchy-like stress are written below

$$\left(1 - \ell^2\Delta\right)(\mu\Delta\boldsymbol{u} + (\lambda+\mu)\nabla\nabla\cdot\mathbf{u}) + \boldsymbol{b} = 0 \qquad \text{in } \Omega \tag{107a}$$

$$\begin{aligned}&\boldsymbol{n}^*\cdot\left(\bar{\boldsymbol{\sigma}} - \ell^2\Delta\bar{\boldsymbol{\sigma}}\right) - \ell^2\nabla_{\partial\Omega}\cdot(\boldsymbol{n}^*\cdot\nabla\bar{\boldsymbol{\sigma}}) \\ &\quad + \ell^2\left(\nabla_{\partial\Omega}\cdot\boldsymbol{n}^*\right)\boldsymbol{n}^*\otimes\boldsymbol{n}^* : \nabla\bar{\boldsymbol{\sigma}} = \mathbb{P} \text{ on } \partial\Omega_{\mathbb{P}}\end{aligned} \tag{107b}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \partial\Omega_{\boldsymbol{u}} \tag{107c}$$

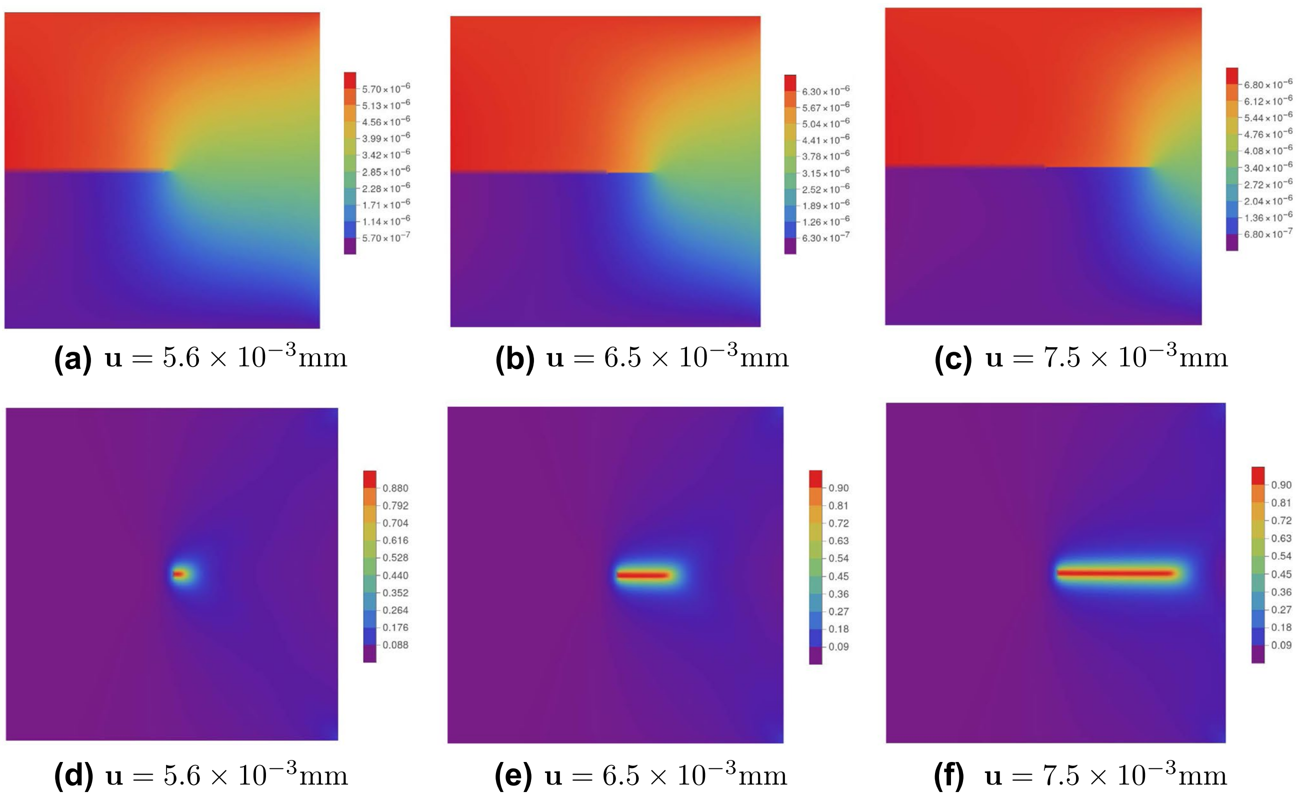


**Fig. 11** 2D single-edge-notched tension test. Crack pattern at displacement field and phase field for a length scale of $l_0 = 0.015$ mm

$$\ell^2 \boldsymbol{n}^* \otimes \boldsymbol{n}^* : \nabla \bar{\boldsymbol{\sigma}} = \mathbb{R} \qquad \text{on } \partial\Omega_{\mathbb{R}} \tag{107d}$$

$$\boldsymbol{n}^* \cdot \nabla \boldsymbol{u} = \frac{\partial \bar{\boldsymbol{u}}}{\partial \boldsymbol{n}^*} = \bar{\boldsymbol{v}} \qquad \text{on } \partial\Omega_{\boldsymbol{v}} \tag{107e}$$

where $\bar{\boldsymbol{u}}, \bar{\boldsymbol{v}}, \mathbb{P}, \mathbb{R}$ represent known functions, $\nabla_{\partial\Omega}(= \nabla - \boldsymbol{n}^* \otimes \boldsymbol{n}^* \cdot \nabla)$ denotes surface gradient. The boundary surface is composed of disjoint and have the follow relationships: $\partial\Omega = \partial\Omega_{\mathbb{P}} \cup \partial\Omega_{\mathbb{R}} \cup \partial\Omega_{\boldsymbol{u}} \cup \partial\Omega_{\boldsymbol{v}}$, $\partial\Omega_{\mathbb{P}} \cup \partial\Omega_{\boldsymbol{u}} = \partial\Omega_{\mathbb{R}} \cup \partial\Omega_{\boldsymbol{v}}$ and $\partial\Omega_{\mathbb{P}} \cap \partial\Omega_{\boldsymbol{u}} = \partial\Omega_{\mathbb{R}} \cap \partial\Omega_{\boldsymbol{v}} = \emptyset$ .

To make the expression clearer, we simplify the Eq. 102 and make some modifications to following

$$\begin{aligned}
\mathbb{W}_i &= \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i1}\left(\tilde{\partial}_\alpha \mathbf{u}_{i1}\right)\mathrm{d}\Omega + \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i2}\left(\tilde{\partial}_\alpha \mathbf{u}_{i2}\right)\mathrm{d}\Omega - \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i3}\left(\tilde{\partial}_\alpha \mathbf{u}_{i3}\right)\mathrm{d}\Omega \\
&\quad - \int_{\partial\Omega_{\mathbb{P}}} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i4}\left(\tilde{\partial}_\alpha \mathbf{u}_{i4}\right)\mathrm{d}S - \int_{\partial\Omega_{u}} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i5}\left(\tilde{\partial}_\alpha \mathbf{u}_{i5}\right)\mathrm{d}S \\
&= \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i1}\left(\mathbf{B}_{\alpha i1}\mathbf{U}_{i1}\right)\mathrm{d}\Omega + \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i2}\left(\mathbf{B}_{\alpha i2}\mathbf{U}_{i2}\right)\mathrm{d}\Omega - \int_{\Omega} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i3}\left(\mathbf{B}_{\alpha i3}\mathbf{U}_{i3}\right)\mathrm{d}\Omega \\
&\quad - \int_{\partial\Omega_{\mathbb{P}}} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i4}\left(\mathbf{B}_{\alpha i4}\mathbf{U}_{i4}\right)\mathrm{d}S - \int_{\partial\Omega_{u}} \mathrm{w}_i(\xi_{ij})\mathbb{W}_{i5}\left(\mathbf{B}_{\alpha i5}\mathbf{U}_{i5}\right)\mathrm{d}S
\end{aligned} \tag{108}$$

where

$$
\begin{aligned}
&\mathrm{w}_i(\xi_{ij}) = (1-\xi_{ij}/\mathbb{l})^n \text{ denotes the weight function}\\
&\mathbb{W}_{i1}\left(\tilde{\partial}_\alpha \mathbf{u}_{i1}\right) = \tfrac{1}{2}\boldsymbol{\sigma}_i : \boldsymbol{\varepsilon}_i, \text{ and } \tilde{\partial}_\alpha \mathbf{u}_{i1} = \left(\tfrac{\partial u_i}{\partial x}, \tfrac{\partial u_i}{\partial y}, \tfrac{\partial u_i}{\partial z}, \tfrac{\partial v_i}{\partial x}, \tfrac{\partial v_i}{\partial y}, \tfrac{\partial v_i}{\partial z}, \tfrac{\partial w_i}{\partial x}, \tfrac{\partial w_i}{\partial y}, \tfrac{\partial w_i}{\partial z}\right)^T\\
&\mathbb{W}_{i2}\left(\tilde{\partial}_\alpha \mathbf{u}_{i2}\right) = \tfrac{l^2}{2}\nabla\boldsymbol{\sigma}_i : \nabla\boldsymbol{\varepsilon}_i, \text{ and } \tilde{\partial}_\alpha \mathbf{u}_{i2} = \left(\tfrac{\partial^2 u_i}{\partial x\partial x}, \tfrac{\partial^2 u_i}{\partial x\partial y}, \tfrac{\partial^2 u_i}{\partial x\partial z}, \tfrac{\partial^2 u_i}{\partial y\partial y}, \tfrac{\partial^2 u_i}{\partial y\partial z}, \tfrac{\partial^2 u_i}{\partial z\partial z}, \cdots\right)^T\\
&\mathbb{W}_{i3}\left(\tilde{\partial}_\alpha \mathbf{u}_{i3}\right) = -\boldsymbol{b}_i\cdot\boldsymbol{u}_i \text{ and } \tilde{\partial}_\alpha \mathbf{u}_{i3} = (u_i, v_i, w_i)^T\\
&\mathbb{W}_{i4}\left(\tilde{\partial}_\alpha \mathbf{u}_{i4}\right) = -\mathbb{P}_i\cdot\boldsymbol{u}_i, \text{ and } \tilde{\partial}_\alpha \mathbf{u}_{i4} = (u_i, v_i, w_i)^T\\
&\mathbb{W}_{i5}\left(\tilde{\partial}_\alpha \mathbf{u}_{i5}\right) = -\mathbb{P}_i\cdot(\boldsymbol{u}_i-\overline{\boldsymbol{u}_i}), \text{ and } \tilde{\partial}_\alpha \mathbf{u}_{i5} = (u_i, \tfrac{\partial u_i}{\partial x}, \tfrac{\partial u_i}{\partial y}, \tfrac{\partial u_i}{\partial z}, \tfrac{\partial^2 u_i}{\partial x\partial x}, \tfrac{\partial^2 u_i}{\partial x\partial y}, \tfrac{\partial^2 u_i}{\partial x\partial z}, \tfrac{\partial^2 u_i}{\partial y\partial y}, \tfrac{\partial^2 u_i}{\partial y\partial z}, \tfrac{\partial^2 u_i}{\partial z\partial z},\\
&\tfrac{\partial^3 u_i}{\partial x\partial x\partial x}, \tfrac{\partial^3 u_i}{\partial x\partial x\partial y}, \tfrac{\partial^3 u_i}{\partial x\partial x\partial z}, \tfrac{\partial^3 u_i}{\partial x\partial y\partial y}, \tfrac{\partial^3 u_i}{\partial x\partial y\partial z}, \tfrac{\partial^3 u_i}{\partial x\partial z\partial z}, \tfrac{\partial^3 u_i}{\partial y\partial y\partial y}, \tfrac{\partial^3 u_i}{\partial y\partial y\partial z}, \tfrac{\partial^3 u_i}{\partial y\partial z\partial z}, \tfrac{\partial^3 u_i}{\partial z\partial z\partial z} \cdots)^T
\end{aligned}
\tag{109}
$$

According to the Eq. 108, we can establish the residual and the corresponding tangent stiffness matrix at point $\mathbf{x}_i$ of the overall system:

$$
\begin{aligned}
\boldsymbol{R}_i &= \frac{\partial \mathbb{W}_i}{\partial \mathbf{U}_i}\\
&= \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i1}\right)^T \mathbf{D}_{i1}\mathbf{B}_{\alpha i1}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i2}\right)^T \mathbf{D}_{i2}\mathbf{B}_{\alpha i2}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i3}\right)^T \mathbf{D}_{i3}\mathbf{B}_{\alpha i3}\Delta V_j\\
&+ \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i4}\right)^T \mathbf{D}_{i4}\mathbf{B}_{\alpha i4}\Delta S_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i5}\right)^T \mathbf{D}_{i5}\mathbf{B}_{\alpha i5}\Delta S_j
\end{aligned}
\tag{110}
$$

$$
\begin{aligned}
&\mathscr{K}_i = \frac{\partial \boldsymbol{R}_i}{\partial \mathbf{U}_i}\\
&= \frac{\partial}{\partial \mathbf{U}_i}\Big(\sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i1}\right)^T \mathbf{D}_{i1}\mathbf{B}_{\alpha i1}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i2}\right)^T \mathbf{D}_{i2}\mathbf{B}_{\alpha i2}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i3}\right)^T \mathbf{D}_{i3}\mathbf{B}_{\alpha i3}\Delta V_j\\
&+ \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i4}\right)^T \mathbf{D}_{i4}\mathbf{B}_{\alpha i4}\Delta S_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\tilde{\partial}_\alpha \mathbf{u}_{i5}\right)^T \mathbf{D}_{i5}\mathbf{B}_{\alpha i5}\Delta S_j\Big)\\
&= \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\mathbf{B}_{\alpha i1}\right)^T \mathbf{D}_{i1}\mathbf{B}_{\alpha i1}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\mathbf{B}_{\alpha i2}\right)^T \mathbf{D}_{i2}\mathbf{B}_{\alpha i2}\Delta V_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\mathbf{B}_{\alpha i3}\right)^T \mathbf{D}_{i3}\mathbf{B}_{\alpha i3}\Delta V_j\\
&+ \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\mathbf{B}_{\alpha i4}\right)^T \mathbf{D}_{i4}\mathbf{B}_{\alpha i4}\Delta S_j + \sum_{j\in\mathcal{S}_i} \mathrm{w}_i\left(\mathbf{B}_{\alpha i5}\right)^T \mathbf{D}_{i5}\mathbf{B}_{\alpha i5}\Delta S_j
\end{aligned}
\tag{111}
$$

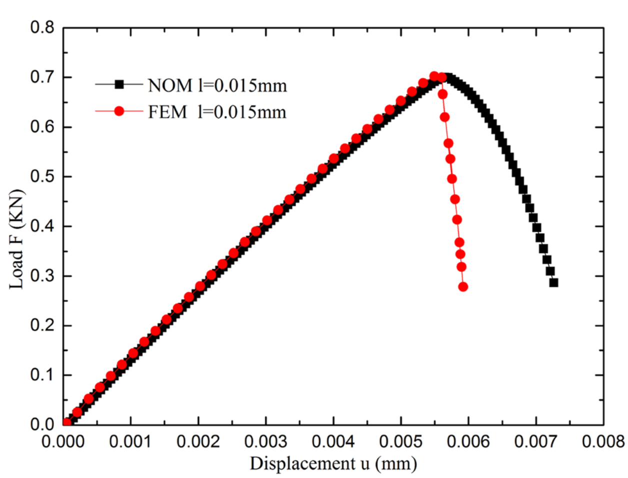


**Fig. 12** Reaction force-displacement curves for $l_0 = 0.015$ mm

A 3D gradient elasticity cantilever beam under uniform shear load is considered. The cantilever beam with dimensions of $25\times10\times3$ m$^3$. The material parameters are considered: $E = 6\times10^4$ MPa, $\nu = 0.33$. The uniform shear force $P_y = -1.0\times10^{-3}$ MPa. The higher-order NOM based on numerical integration [77] is employed to investigate the effectiveness of the current method for gradient elasticity cantilever beam. The cantilever beam is discretized into 3744 points and hexahedral background meshes are generated for the numerical integration. We employ 40 Gauss neighbor points in the numerical test. For separate variables $(u, v, w)$ in 3D cantilever beam, Eq. 108 include the differential operators $\nabla u$, $\nabla v$,$\nabla w$, $\nabla^2 u$, $\nabla^2 v$, $\nabla^2 w$, $\nabla^3 u$, $\nabla^3 v$, $\nabla^3 w$. The maximal order of partial derivatives in Eq. 108 is three, hence we select the third order of nonlocal operators in Eq. 31. In addition, various gradient coefficients

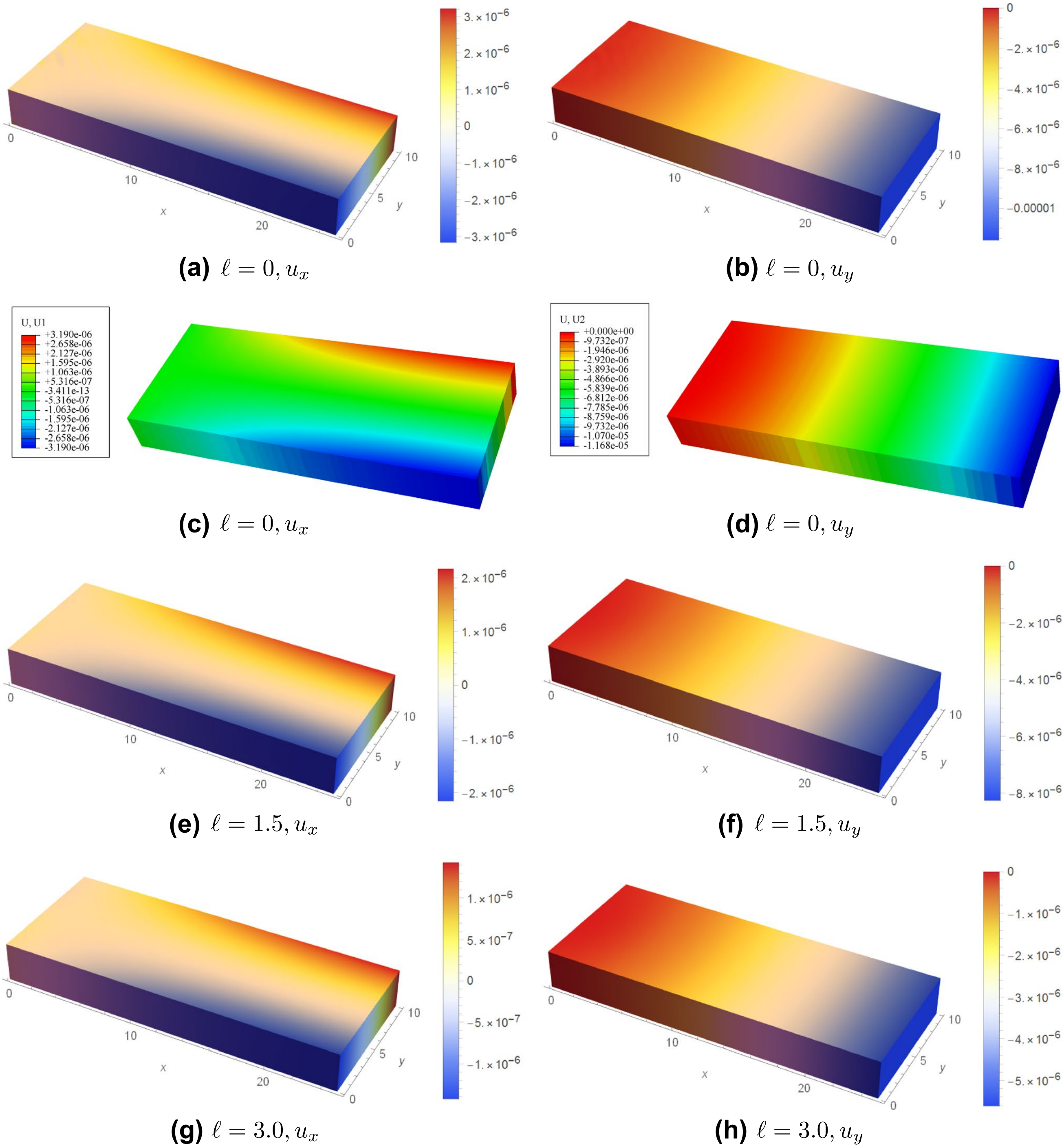


**Fig. 13** Displacement for various different gradient coefficients

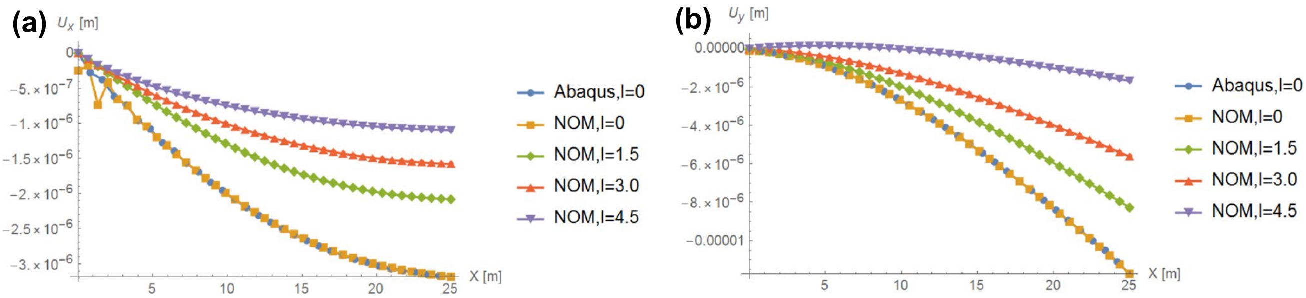


**Fig. 14** Displacement for points on the line $(y=10, z=0)$

$\ell = 0, 1.5, 3.0, 4.5$ are studied by current method and the displacement field for various gradient coefficients is given in Fig. 13. Figure 14 presents the displacement for points on the line $(y=10, z=0)$. The results obtained by higher-order NOM correspond well with those obtained by ABAQUS and with the gradient coefficient raising, the deform of the beam become more uniformly.

## 5 Conclusions

In this paper, we present the implementation procedure of first-order and higher-order NOM, and an open-source Mathematica code is presented and explained in detail. The performance of first-order NOM and higher-order NOM results are demonstrated compare with the corresponding analytical solutions or the results of the FEM commercial software. Concluding remarks can be stated as follows:

Similar to the FEM and meshless method, NOM can establish the operator energy functional and tangent stiffness matrix by some matrix multiplications. However, unlike FEM and the meshless method, NOM can derive differential operators directly without employing shape functions. Hence, the complexity of the NOM is significantly reduced. The NOM requires only the definition of the energy, for a given energy functional, the nonlocal operators can be established automatically by the highest order of partial derivative and dimensions. Support, dual-support, nonlocal differential operators, and operator energy functional are the fundamental components of NOM. According to several numerical examples, which illustrate the method's high performance and capabilities. In conclusion, NOM is an easy-to-use, flexible, and efficient numerical method, further development of this method will incorporate anisotropic material and elastic-plastic material.

## Appendix A: The 2D form of the nonlocal gradient operators for vector field $\tilde{\nabla}\mathbf{u}$ and scalar field $\tilde{\nabla}u$

$$\tilde{\nabla}\mathbf{u}_i = \left[\frac{\partial u_i}{\partial x_i}, \frac{\partial u_i}{\partial y_i}, \frac{\partial v_i}{\partial x_i}, \frac{\partial v_i}{\partial y_i}\right]^T$$

$$= \begin{bmatrix} -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & 0 & \xi_{x_{j1}} & 0 & \cdots & \xi_{x_{jn}} & 0 \\ 0 & -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & 0 & \xi_{y_{j1}} & \cdots & 0 & \xi_{y_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & 0 & \xi_{x_{j1}} & 0 & \cdots & \xi_{x_{jn}} & 0 \\ 0 & -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & 0 & \xi_{y_{j1}} & \cdots & 0 & \xi_{y_{jn}} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ u_{j1} \\ v_{j1} \\ \cdots \\ u_{jn} \\ v_{jn} \end{bmatrix} = \mathscr{B}_i\mathscr{U}_i \tag{112}$$

$$\tilde{\nabla}u_i = \begin{bmatrix} \frac{\partial u_i}{\partial x} \\ \frac{\partial u_i}{\partial y} \end{bmatrix} = \begin{bmatrix} -\sum_{j\in\mathcal{S}_i}\xi_{x_j} & \xi_{x_{j1}} & \cdots & \xi_{x_{jn}} \\ -\sum_{j\in\mathcal{S}_i}\xi_{y_j} & \xi_{y_{j1}} & \cdots & \xi_{y_{jn}} \end{bmatrix} \begin{bmatrix} u_i \\ u_{j1} \\ \cdots \\ u_{jn} \end{bmatrix} = \mathscr{B}_i\mathscr{U}_i \tag{113}$$

$$\text{where}(\xi_{x_j}, \xi_{y_j}) = \mathrm{w}(\xi_{ij})\xi_{ij}^T V_j \cdot \left[\sum_{j\in\mathcal{S}_i}\mathrm{w}(\xi_{ij})\xi_{ij}\otimes\xi_{ij}\Delta V_j\right]^{-1}$$

## Appendix B: The derivation of the hourglass tangent stiffness matrix and the global tangent stiffness matrix in 2D form

$$\begin{aligned} \mathrm{d}\mathscr{U}_i &= \left(\frac{\partial u_i}{\partial x}, \frac{\partial u_i}{\partial y}, \frac{\partial v_i}{\partial x}, \frac{\partial v_i}{\partial y}\right)^T \\ \mathscr{U}_i &= (u_i, v_i, u_{j1}, v_{j1} \cdots u_{jn}, v_{jn})^T \end{aligned} \tag{114}$$

$$\begin{aligned} \Theta_i =& \frac{1}{2}\begin{bmatrix} \frac{\partial u_i}{\partial x} & \frac{\partial u_i}{\partial y} \\ \frac{\partial v_i}{\partial x} & \frac{\partial v_i}{\partial y} \end{bmatrix} \left( \begin{bmatrix} \frac{\partial u_i}{\partial x} & \frac{\partial u_i}{\partial y} \\ \frac{\partial v_i}{\partial x} & \frac{\partial v_i}{\partial y} \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{bmatrix} \right) \\ =& \frac{1}{2}\left( \frac{\partial u_i}{\partial x}[k_{11}\frac{\partial u_i}{\partial x} + k_{12}\frac{\partial u_i}{\partial y}] + \frac{\partial u_i}{\partial y}[k_{12}\frac{\partial u_i}{\partial x} + k_{22}\frac{\partial u_i}{\partial y} + \right. \\ & \left. + \frac{\partial v_i}{\partial x}[k_{11}\frac{\partial v_i}{\partial x} + k_{12}\frac{\partial v_i}{\partial y}] + \frac{\partial v_i}{\partial y}[k_{12}\frac{\partial v_i}{\partial x} + k_{22}\frac{\partial v_i}{\partial y}] \right) \end{aligned} \tag{115}$$

$$
\begin{aligned}
\bar{\delta}\Theta_i =& \partial_{\mathrm{d}\mathscr{U}_i}\Theta_i \\
=& \begin{bmatrix} [k_{11}\frac{\partial u_i}{\partial x} + k_{12}\frac{\partial u_i}{\partial y}] & [k_{12}\frac{\partial u_i}{\partial x} + k_{22}\frac{\partial u_i}{\partial y}] \\ [k_{11}\frac{\partial v_i}{\partial x} + k_{12}\frac{\partial v_i}{\partial y}] & [k_{12}\frac{\partial v_i}{\partial x} + k_{22}\frac{\partial v_i}{\partial y}] \end{bmatrix}
\end{aligned} \tag{116}
$$

$$
\begin{aligned}
\bar{\delta}^2\Theta_i =& \partial_{\mathrm{d}\mathscr{U}_i\mathrm{d}\mathscr{U}_i}\Theta_i \\
=& \partial_{\mathrm{d}\mathscr{U}_i}\begin{bmatrix} [k_{11}\frac{\partial u_i}{\partial x} + k_{12}\frac{\partial u_i}{\partial y}] & [k_{12}\frac{\partial u_i}{\partial x} + k_{22}\frac{\partial u_i}{\partial y}] \\ [k_{11}\frac{\partial v_i}{\partial x} + k_{12}\frac{\partial v_i}{\partial y}] & [k_{12}\frac{\partial v_i}{\partial x} + k_{22}\frac{\partial v_i}{\partial y}] \end{bmatrix} \\
=& \begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ k_{12} & k_{22} & 0 & 0 \\ 0 & 0 & k_{11} & k_{12} \\ 0 & 0 & k_{12} & k_{22} \end{bmatrix}
\end{aligned} \tag{117}
$$

$$
\begin{aligned}
\delta\Upsilon_i =& \partial_{\mathscr{U}_i}\Upsilon_i \\
=& \partial_{\mathscr{U}_i}\{\frac{1}{2}[(u_j - u_i)^2 + (v_j - v_i)^2]\} \\
=& [(u_i - u_j), (v_i - v_j), (-u_i + u_j), (-v_i + v_j)]
\end{aligned} \tag{118}
$$

$$
\begin{aligned}
\delta^2\Upsilon_i =& \partial_{\mathscr{U}_i\mathscr{U}_i}\Upsilon_i \\
=& \partial_{\mathscr{U}_i}\{(u_i - u_j), (v_i - v_j), (-u_i + u_j), (-v_i + v_j)\} \\
=& \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{119}
$$

According to Eqs. 117 and 119, Eq. 47 can be rewritten as

$$
\begin{aligned}
\mathcal{F}_i^{hg} =& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\sum_{j\in\mathcal{S}_i} \mathrm{w}(\boldsymbol{\xi}_{ij})\left((u_j - u_i)^2 + (v_j - v_i)^2\right)\Delta V_j - \nabla\otimes\mathbf{u}_i : \nabla\otimes\mathbf{u}_i\cdot\mathbf{K}_i\right) \\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\mathscr{U}_i^T\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n} \\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix}\mathscr{U}_i - \mathrm{d}\mathscr{U}_i^T\begin{bmatrix} \mathbf{K}_i & 0 \\ 0 & \mathbf{K}_i \end{bmatrix}\mathrm{d}\mathscr{U}_i\right) \\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\left(\mathscr{U}_i^T\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n} \\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix}\mathscr{U}_i - \mathscr{U}_i^T\mathscr{B}_i^T\begin{bmatrix} \mathbf{K}_i & 0 \\ 0 & \mathbf{K}_i \end{bmatrix}\mathscr{B}_i\mathscr{U}_i\right) \\
=& \frac{p^{hg}}{2m_{\mathbf{K}_i}}\mathscr{U}_i^T\left(\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n} \\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix} - \mathscr{B}_i^T\begin{bmatrix} \mathbf{K}_i & 0 \\ 0 & \mathbf{K}_i \end{bmatrix}\mathscr{B}_i\right)\mathscr{U}_i
\end{aligned} \tag{120}
$$

where $\mathbf{I}_j = \mathrm{w}(\boldsymbol{\xi}_{ij})\Delta V_j(1,1)\otimes(1,1)^T$, $\mathbf{K}_i = \begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{bmatrix}$.

$$
\begin{aligned}
\overset{hg}{\mathscr{K}_i} =& \partial_{\mathscr{U}_i\mathscr{U}_i}\mathcal{F}_i^{hg} \\
=& \frac{p^{hg}}{m_{\mathbf{K}_i}}\left(\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n} \\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix} - \mathscr{B}_i^T\begin{bmatrix} \mathbf{K}_i & 0 \\ 0 & \mathbf{K}_i \end{bmatrix}\mathscr{B}_i\right)
\end{aligned} \tag{121}
$$

Finally, we can obtain the summation of the first-order global tangent stiffness matrix and hourglass tangent stiffness matrix in support $\mathcal{S}_i$

$$
\begin{aligned}
\mathbb{K}_i =& \mathscr{K}_i + \overset{hg}{\mathscr{K}_i} \\
=& \sum_{j\in\mathcal{S}_i}\left(\mathscr{B}_i^T\left(\mathscr{D} - \frac{p^{hg}}{m_{\mathbf{K}_i}}\begin{bmatrix} \mathbf{K}_i & 0 \\ 0 & \mathbf{K}_i \end{bmatrix}\right)\mathscr{B}_i\right. \\
&\left. + \frac{p^{hg}}{m_{\mathbf{K}_i}}\begin{bmatrix} \sum_{j\in\mathcal{S}_i}\mathbf{I}_j & -\mathbf{I}_{j1} & \cdots & -\mathbf{I}_{j_n} \\ -\mathbf{I}_{j1} & \mathbf{I}_{j1} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ -\mathbf{I}_{j_n} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{j_n} \end{bmatrix}\right)\Delta V_j
\end{aligned} \tag{122}
$$

# Appendix C: Mathematica code for higher-order nonlocal operator method

```
FuncRK[uvw_List, uvw0_List] := Block[{El = uvw[[1, 1]], nu = uvw[[2, 1]], u01 = uvw[[3, 2]], u10 = uvw[[3, 3]],
   v01 = uvw[[4, 2]], v10 = uvw[[4, 3]], $67, $92, $93, $97, $98, $99, $101, $102, $100, $103, $133, $105,
   $124, $127, $128, $162, $180, $173, $174, $175, $176, $177, $178, $179, $192, $202, $203, $204, $205,
   $221, $181, $182, $183, $184, $185, $206, $207, $208, $209, $210, $211, $186, $187, $188, $189, $190,
   $212, $213, $214, $215, $216, $217, $223, $222, $218, $219, $220}, $67 = nu^2; $92 = -$67; $93 = 1 + $92;
  $97 = 1/$93; $98 = -nu; $99 = 1 + $98; $101 = u10 + v01; $102 = nu $101; $100 = $99 u10; $103 = $100 + $102;
  $133 = 1/$93^2; $105 = $99 v01; $124 = $105 + $102; $127 = u01 + v10; $128 = $127^2; $162 = 1/2 El $99 $97 $127;
  $180 = ($99 $97 $127)/2; $173 = 2 $97 u10 v01; $174 = 2 nu $133 u10 $103; $175 = 2 nu $133 v01 $124;
  $176 = $99 nu $133 $128; $177 = -1/2 ($97 $128); $178 = $173 + $174 + $175 + $176 + $177;
  $179 = $178/2; $192 = 1/$93^3; $202 = 2 El $99 nu $133 $127; $203 = -(El $97 $127);
  $204 = $202 + $203; $205 = $204/2; $221 = (El $99 $97)/2; $181 = $97 u10; $182 = nu $97 v01;
  $183 = $97 $103; $184 = $181 + $182 + $183; $185 = $184/2; $206 = 2 El nu $133 u10; $207 = 2 El $67 $133 v01;
  $208 = 2 El $97 v01; $209 = 2 El nu $133 $103; $210 = $206 + $207 + $208 + $209; $211 = $210/2;
  $186 = nu $97 u10; $187 = $97 v01; $188 = $97 $124; $189 = $186 + $187 + $188; $190 = $189/2;
  $212 = 2 El $67 $133 u10; $213 = 2 El $97 u10; $214 = 2 El nu $133 v01; $215 = 2 El nu $133 $124;
  $216 = $212 + $213 + $214 + $215; $217 = $216/2; $223 = El nu $97; $222 = El $97; $218 = El $99 nu $133 $127;
  $219 = -1/2 (El $97 $127); $220 = $218 + $219;
  {{1/2 ($97 u10 $103 + $97 v01 $124 + ($99 $97 $128)/2), 1/2 (2 El $97 u10 v01 + 2 El nu $133 u10 $103 + 2 El nu $133 v01 $124 + El $99 nu $133 $128 - (El $97 $128)/2),
    $162, 1/2 (El $97 u10 + El nu $97 v01 + El $97 $103), 1/2 (El nu $97 u10 + El $97 v01 + El $97 $124), $162},
   {{0, $179, $180, $185, $190, $180},
    {$179, 1/2 (8 El nu $133 u10 v01 + 8 El $67 $192 u10 $103 + 2 El $133 u10 $103 + 8 El $67 $192 v01 $124 + 2 El $133 v01 $124 + 4 El $99 $67 $192 $128 +
        El $99 $133 $128 - 2 El nu $133 $128), $205, $211, $217, $205}, {$180, $220, $221, 0, 0, $221}, {$185, $211, 0, $222, $223, 0},
    {$190, $217, 0, $223, $222, 0}, {$180, $220, $221, 0, 0, $221}}}]
```

```
MultiIndex[d_, sum_] := Module[{a, b, c}, a = Subsets[Range[d + sum], {d}];
   Do[c = a[[i]];
    b = c - 1;
    b[[2 ;;]] -= c[[1 ;; -2]];
    a[[i]] = b, {i, Length[a]}]; a];
GFD0CoeffHalf[ndim_Integer, diffMax_Integer] := Module[{a, ai, iDf, iDffact, iDfsum, Py, xy, len, io},
   If[ndim < 1, Print["Error, ndim should be positive!"]; Return[]];
   xy = {x, y, z, l, m, n, o, p, q}[[1 ;; ndim]];
   a = MultiIndex[ndim, diffMax][[2 ;;]]; len = Length[a];
   iDf = ConstantArray[0, len];
   iDfsum = ConstantArray[0, len];
   iDffact = ConstantArray[0, len];
   Py = ConstantArray[0, len];
   Do[ai = a[[i]];
    iDf[[i]] = FromDigits[ai];
    iDfsum[[i]] = Total[ai];
    iDffact[[i]] = Times @@ Factorial[ai];
    Py[[i]] = Times @@ (xy^ai), {i, len}];
   io = Ordering[iDfsum];
   {iDf[[io]], iDffact[[io]], iDfsum[[io]], Py[[io]]}];

NOMPwKhg[np0_, coord_List, vol_, Nei_List, pfun_, WeiF_, hgPen_] :=
  Module[{tl, ndof, Nnode, ndim, udim, kk, np}, If[np0 < 0, np = Ceiling[Length[coord] / 1000], np = np0];
   tl = Ngroup[Length[coord], np];
   kk = ParallelTable[NOMPwKhgPart[tl[[i]], coord, vol, Nei, pfun, WeiF, hgPen], {i, np}];
   {Flatten[kk[[All, 1]], 1], Flatten[kk[[All, 2]], 1]}];
NOMPwKhgPart[nodeList_, coord_List, vol_, Nei_List, pfun_, WeiF_, hgPen_] :=
  Module[{pvol, Nnode, pwList, hgList, kp = 1, coordi, pw, khg, NeiI, hg, hi, voli = 0},
   Nnode = Length[coord];
   pwList = EmptyList[Length[nodeList]];
   hgList = EmptyList[Length[nodeList]];
   If[Length[vol] == 0, pvol = ConstantArray[vol, Nnode];];
   If[Length[vol] == Nnode, pvol = vol];
   Do[
    NeiI = Nei[[i]];
    coordi = ListMinus[coord[[NeiI]], coord[[i]]];
    voli = pvol[[NeiI]];
    hi = Norm[coordi[[-1]]];
    {pw, khg} = Khgs[coordi, pfun, WeiF, voli, hgPen, hi];
    pw = PhuToPu[pw, iDfsum, iDffact, hi];
    pwList[[kp]] = pw;
    hgList[[kp++]] = khg;
    , {i, nodeList}
   ];
   {pwList, hgList}];
```

```
Ngroup[nmax_, ngroup_] := Module[{tl = {}, i1, i2, inc}, inc = Round[nmax / ngroup];
   Do[AppendTo[tl, Range[inc (i - 1) + 1, inc i]], {i, 1, ngroup - 1}];
   AppendTo[tl, Range[inc (ngroup - 1) + 1, nmax]]; tl];
EmptyList[n_Integer] := Block[{el = {}}, Do[AppendTo[el, {}], {i, n}]; el];
ListMinus[v1_List, v0_List] := Module[{v2 = v1}, Do[v2[[i]] -= v0, {i, Length[v1]}]; v2];
PhuToPu[pw_List, iDfsum_List, iDffact_List, h_] := Module[{p = {}, i},
   p = pw;
   Do[p[[i]] *= 1. / (h^iDfsum[[i]] / iDffact[[i]]), {i, Length[pw]}]; p];
Khgs[v1_List, pfun_, wfun_, vol_List, penalty_, h_] := Module[{
    len = Length[pfun[v1[[1]], 1]],
    num = Length[v1], k, p, trH = 0, r1, p1, w1, pkp = {}, pw0, wl},
   pkp = ConstantArray[0, {num, num}];
   k = ConstantArray[0, {len, len}];
   p = ConstantArray[0, {num, len}];
   wl = Norm /@ v1;
   Do[wl[[i]] = wfun[wl⟦i⟧, h] vol⟦i⟧, {i, num}];
   wl *= 1.0 / Total[wl];
   Do[r1 = Norm[v1[[i]]]; w1 = wl[[i]];
    trH += w1 r1 r1;
    pkp[[i, i]] = w1;
    p1 = pfun[v1[[i]], h]; p[[i]] = w1 p1;
    k += w1 TensorProduct[p1, p1], {i, num}];
   pw0 = Inverse[k].pᵀ;
   pkp -= p.pw0;
   pkp *= penalty / trH;
   {pw0, pkp}];

NOMRK[np_, nodes_List, volSet_List, nonvars_List, coord_List, uvw_List, PW_List,
   KHG_List, Nei_List, idf_List, pfun_, WeiF_, FuncRK_, hgPen_] := Module[{tl, vtl, nvs,
    ndof, Nnode, ndim, udim, kk, ksp, rsp},
   {tl, vtl, nvs} = NgroupL[nodes, volSet, nonvars, np];
   kk = ParallelTable[NOMRKPartNonVars[tl[[i]], vtl⟦i⟧, nvs⟦i⟧, coord,
      uvw, PW, KHG, Nei, idf, pfun, WeiF, FuncRK, hgPen], {i, Length[tl]}];
   ksp = kk[[1, 1]];
   rsp = kk[[1, 2]];
   Do[ksp += kk[[i, 1]];
    rsp += kk[[i, 2]], {i, 2, Length[tl]}];
   {ksp, rsp}
  ];
```

```
NOMRKPartNonVars[nodeList_List, volSet_List, nonvars_List, coord_List, uvw_List,
   PW_List, KHG_List, Nei_List, idf2_List, pfun_, WeiF_, FuncRK_, hgPen_] :=
  Module[{pvol, Nnode, ndim, udim, ndof, Ksp, Rsp, coordi, pw, khg, nablaU,
    NeiI, NeiI2, Kst, hg, HG, hi, uvwi, duvwi, Ri, Di, Rst, Rhg, k = 1, ishgPen1 = True},
   If[Length[nonvars] ≠ Length[nodeList], Print["Error"]; Return[]];
   Nnode = Length[coord];
   ndim = Length[coord[[1]]];
   udim = Length[idf2];
   ndof = udim Nnode;
   Ksp = SparseArray[{}, {ndof, ndof}];
   Rsp = SparseArray[{}, ndof];
   If[Length[hgPen] > 0, ishgPen1 = False];
   Do[
    NeiI = Nei[[i]];
    uvwi = uvw[[Prepend[NeiI, i]]];
    {pw, khg} = {PW[[i]], KHG[[i]]};
    duvwi = pw.uvwi;
    PrependTo[duvwi, uvwi[[1]]];
    {Ri, Di} = FuncRK[duvwiᵀ, nonvars[[k]]];
    nablaU = UNablaU[idf2, pw];
    Rst = Ri.nablaU;
    Kst = nablaUᵀ.Di.nablaU;
    If[hgPen ≠ 0.,
     hg = If[ishgPen1, hgPen KhgUdim[khg, udim], KhgUdim[khg, udim, hgPen]];
     Rhg = hg.Flatten[uvwi]; Kst += hg; Rst += Rhg];
    Rst *= volSet[[k]];
    Kst *= volSet[[k]];
    k++;
    PrependTo[NeiI, i];
    NeiI2 = NeiIIndex[NeiI, udim];
    Rsp[[NeiI2]] += Rst;
    Ksp += SparseArray[Tuples[NeiI2, 2] → Flatten[Kst], {ndof, ndof}], {i, nodeList}
   ];
   {Ksp, Rsp}];
```

```
NgroupL[nodes_List, volSet_List, nonvars_List, ngroup0_] := Module[{tl = {},
    vtl = {}, nv = {}, i1, i2, inc, nmax, ngroup},
   nmax = Length[nodes];
   If[ngroup0 < 0, ngroup = Ceiling[nmax / 1000], ngroup = ngroup0];
   inc = Round[nmax / ngroup];
   Do[i1 = Range[inc (i - 1) + 1, inc i];
    AppendTo[tl, nodes[[i1]]];
    AppendTo[vtl, volSet[[i1]]];
    AppendTo[nv, nonvars[[i1]]];, {i, 1, ngroup - 1}];
   i1 = Range[inc (ngroup - 1) + 1, nmax];
   AppendTo[tl, nodes[[i1]]]; AppendTo[vtl, volSet[[i1]]];
   AppendTo[nv, nonvars[[i1]]]; {tl, vtl, nv}];
UNablaU[idf_List, pw_List] := Module[{udim = Length[idf], ulen,
    nNode = Length[pw[[1]]], nablaU = {}, kn = 1, i, j},
   ulen = Length[Flatten[idf]];
   nablaU = ConstantArray[0, {ulen, udim nNode}];
   Do[Do[If[j ≠ 1, nablaU[[kn++, i ;; -1 ;; udim]] = pw[[j - 1]],
      nablaU[[kn++, i]] = 1], {j, idf[[i]]}], {i, udim}]; nablaU];
KhgNdim[khg_, ndim_] := Module[{kn}, kn = ConstantArray[0., ndim Dimensions[khg]];
   Do[kn[[i ;; -1 ;; ndim, i ;; -1 ;; ndim]] = khg, {i, ndim}]; kn];
KhgUdim[khg_, dim_, hgList_] := Module[{kn}, kn = ConstantArray[0., dim Dimensions[khg]];
   Do[kn[[i ;; -1 ;; dim, i ;; -1 ;; dim]] = hgList[[i]] khg, {i, dim}]; kn];
NeiIIndex[NeiI_List, udim_] := Module[{n2, n3}, n2 = ConstantArray[0, udim Length[NeiI]];
   n3 = udim (NeiI - 1);
   Do[n2[[i ;; -1 ;; udim]] = n3 + i, {i, udim}];
   n2];
```

**Acknowledgements** The financial support provided by China Scholarship Council is gratefully acknowledged.

**Funding** Open Access funding enabled and organized by Projekt DEAL.



## References

1. Rogula D (1982) Introduction to nonlocal theory of material media, In: Nonlocal theory of material media. Springer, pp 123–222
2. Eringen AC (1972) Linear theory of nonlocal elasticity and dispersion of plane waves. Int J Eng Sci 10:425–435
3. Faghidian SA (2018) Integro-differential nonlocal theory of elasticity. Int J Eng Sci 129:96–110
4. Povstenko YZ (1999) The nonlocal theory of elasticity and its applications to the description of defects in solid bodies. J Math Sci 97:3840–3845
5. Eringen AC (1984) Theory of nonlocal elasticity and some applications, technical report. Princeton University NJ Department of Civil Engineering
6. Silling SA (2000) Reformulation of elasticity theory for discontinuities and long-range forces. J Mech Phys Solids 48:175–209
7. Silling SA, Epton M, Weckner O, Xu J, Askari E (2007) Peridynamic states and constitutive modeling. J Elasticity 88:151–184
8. Silling SA (2010) Linearized theory of peridynamic states. J Elasticity 99:85–111
9. Rabczuk T, Zi G, Bordas S, Nguyen-Xuan H (2010) A simple and robust three-dimensional cracking-particle method without enrichment. Comput Methods Appl Mech Eng 199:2437–2455
10. Rabczuk T, Belytschko T (2004) Cracking particles: a simplified meshfree method for arbitrary evolving cracks. Int J Numer Methods Eng 61:2316–2343
11. Rabczuk T, Belytschko T (2007) A three-dimensional large deformation meshfree method for arbitrary evolving cracks. Comput Methods Appl Mech Eng 196:2777–2799
12. Areias P, Reinoso J, Camanho P, De Sá JC, Rabczuk T (2018) Effective 2D and 3D crack propagation with local mesh

refinement and the screened Poisson equation. Eng Fracture Mech 189:339–360
13. Areias P, Rabczuk T (2017) Steiner-point free edge cutting of tetrahedral meshes with applications in fracture. Finite Elem Anal Des 132:27–41
14. Areias P, Rabczuk T, Msekh M (2016) Phase-field analysis of finite-strain plates and shells including element subdivision. Comput Methods Appl Mech Eng 312:322–350
15. Areias P, Msekh M, Rabczuk T (2016) Damage and fracture algorithm using the screened Poisson equation and local remeshing. Eng Fracture Mech 158:116–143
16. Ghorashi SS, Valizadeh N, Mohammadi S, Rabczuk T (2015) T-spline based xiga for fracture analysis of orthotropic media. Comput Struct 147:138–146
17. Areias P, Rabczuk T, Camanho P (2014) Finite strain fracture of 2D problems with injected anisotropic softening elements. Theor Appl Fracture Mech 72:50–63
18. Amiri F, Millán D, Shen Y, Rabczuk T, Arroyo M (2014) Phase-field modeling of fracture in linear thin shells. Theor Appl Fract Mech 69:102–109
19. Areias P, Rabczuk T, Dias-da Costa D (2013) Element-wise fracture algorithm based on rotation of edges. Eng Fract Mech 110:113–137
20. Areias P, Rabczuk T (2013) Finite strain fracture of plates and shells with configurational forces and edge rotations. Int J Numer Methods Eng 94:1099–1122
21. Gerstle W, Sau N, Silling S (2007) Peridynamic modeling of concrete structures. Nucl Eng Des 237:1250–1258
22. Yile H, Yin Y, Hai W (2013) Damage analysis method for laminates based on peridynamic theory. Chin J Theor Appl Mech 45(4):624–628
23. Oterkus E, Madenci E, Weckner O, Silling S, Bogert P, Tessler A (2012) Combined finite element and peridynamic analyses for predicting failure in a stiffened composite curved panel with a central slot. Compos Struct 94:839–850
24. Shen F, Zhang Q, Huang D, Zhao J-J (2013) Damage and failure process of concrete structure under uni-axial tension based on peridynamics modeling. Chin J Comput Mech 30:79–83
25. Lindsay P, Parks M, Prakash A (2016) Enabling fast, stable and accurate peridynamic computations using multi-time-step integration. Comput Methods Appl Mech Eng 306:382–405
26. Ren H, Zhuang X, Cai Y, Rabczuk T (2016) Dual-horizon peridynamics. Int J Numer Methods Eng 108:1451–1476
27. Ren H, Zhuang X, Rabczuk T (2017) Dual-horizon peridynamics: a stable solution to varying horizons. Comput Methods Appl Mech Eng 318:762–782
28. Di Paola M, Failla G, Zingales M (2009) Physically-based approach to the mechanics of strong non-local linear elasticity theory. J Elasticity 97:103–130
29. Di Paola M, Failla G, Zingales M (2010) The mechanically-based approach to 3D non-local linear elasticity theory: long-range central interactions. Int J Solids Struct 47:2347–2358
30. Emmrich E, Weckner O (2007) Analysis and numerical approximation of an integro-differential equation modeling non-local effects in linear elasticity. Math Mech Solids 12:363–384
31. El-Nabulsi RA (2018) Jerk in planetary systems and rotational dynamics, nonlocal motion relative to earth and nonlocal fluid dynamics in rotating earth frame. Earth Moon Planets 122:15–41
32. Lee H-L, Chang W-J (2008) Free transverse vibration of the fluid-conveying single-walled carbon nanotube using nonlocal elastic theory. J Appl Phys 103:024302
33. Eringen AC (1973) Theory of nonlocal electromagnetic elastic solids. J Math Phys 14:733–740
34. Van Mechelen T, Jacob Z (2019) Nonlocal topological electromagnetic phases of matter. Phys Rev B 99:205146
35. Pijaudier-Cabot G, Haidar K, Dubé J-F (2004) Non-local damage model with evolving internal length. Int J Numer Anal Methods Geomech 28:633–652
36. Pijaudier-Cabot G, Bažant ZP (1987) Nonlocal damage theory. J Eng Mech 113:1512–1533
37. Lorentz E (2017) A nonlocal damage model for plain concrete consistent with cohesive fracture. Int J Fract 207:123–159
38. Du Q, Gunzburger M, Lehoucq RB, Zhou K (2013) A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. Math Models Methods Appl Sci 23:493–540
39. Alali B, Liu K, Gunzburger M (2015) A generalized nonlocal vector calculus. Zeitschrift für angewandte Mathematik und Physik 66:2807–2828
40. D'Elia M, Gulian M, Olson H, Karniadakis GE (2020) A unified theory of fractional, nonlocal, and weighted nonlocal vector calculus. arXiv:2005.07686
41. Nagaraj S (2020) Optimization and learning with nonlocal calculus. arXiv:2012.07013
42. Kilic B, Madenci E (2010) An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. Theor Appl Fract Mech 53:194–204
43. Chang S-Y (2007) Improved explicit method for structural dynamics. J Eng Mech 133:748–760
44. Hull J, White A (1990) Valuing derivative securities using the explicit finite difference method. J Financ Quant Anal 25(1):87–100
45. Chang S-Y (2009) An explicit method with improved stability property. Int J Numer Methods Eng 77:1100–1120
46. Askar A, Cakmak AS (1978) Explicit integration method for the time-dependent Schrodinger equation for collision problems. J Chem Phys 68:2794–2798
47. Gravouil A, Combescure A (2001) Multi-time-step explicit-implicit method for non-linear structural dynamics. Int J Numer Methods Eng 50:199–225
48. Briley W, McDonald H (1977) Solution of the multidimensional compressible Navier-Stokes equations by a generalized implicit method. J Comput Phys 24:372–397
49. Luo H, Baum JD, Löhner R (1998) A fast, matrix-free implicit method for compressible flows on unstructured grids. J Comput Phys 146:664–690
50. Ciment M, Leventhal SH, Weinberg BC (1978) The operator compact implicit method for parabolic equations. J Comput Phys 28:135–166
51. Breitenfeld M, Geubelle P, Weckner O, Silling S (2014) Non-ordinary state-based peridynamic analysis of stationary crack problems. Comput Methods Appl Mech Eng 272:233–250
52. Madenci E, Dorduncu M, Phan N, Gu X (2019) Weak form of bond-associated non-ordinary state-based peridynamics free of zero energy modes with uniform or non-uniform discretization. Eng Fract Mech 218:106613
53. Madenci E, Barut A, Dorduncu M (2019) Peridynamic differential operator for numerical analysis. Springer, Berlin
54. Rabczuk T, Ren H, Zhuang X (2019) A nonlocal operator method for partial differential equations with application to electromagnetic waveguide problem. Comput Mater Continua 59(1):31–55
55. Ren H, Zhuang X, Rabczuk T (2020) A nonlocal operator method for solving partial differential equations. Comput Methods Appl Mech Eng 358:112621
56. Ren H, Zhuang X, Rabczuk T (2020) A higher order nonlocal operator method for solving partial differential equations. Comput Methods Appl Mech Eng 367:113132
57. Pian TH, Chen D (1983) On the suppression of zero energy deformation modes. Int J Numer Methods Eng 19:1741–1752
58. Vignjevic R, Campbell J, Libersky L (2000) A treatment of zero-energy modes in the smoothed particle hydrodynamics method. Comput Methods Appl Mech Eng 184:67–85

59. Timoshenko S, Goodier J (1970) Theory of elasticity, vol 412, 3rd edn. McGraw-Hill, New York
60. Zhuang X, Augarde C (2010) Aspects of the use of orthogonal basis functions in the element-free Galerkin method. Int J Numer Methods Eng 81:366–380
61. Boresi AP, Chong K, Lee JD (2010) Elasticity in engineering mechanics. Wiley, New York
62. Hibbett, Karlsson, Sorensen (1998) ABAQUS/standard: User's Manual, volume 1, Hibbitt, Karlsson & Sorensen
63. Wheeler M, Wick T, Wollner W (2014) An augmented-Lagrangian method for the phase-field approach for pressurized fractures. Comput Methods Appl Mech Eng 271:69–85
64. Miehe C, Schänzel L-M, Ulmer H (2015) Phase field modeling of fracture in multi-physics problems. Part I. Balance of crack surface and failure criteria for brittle crack propagation in thermo-elastic solids. Comput Methods Appl Mech Eng 294:449–485
65. Miehe C, Mauthe S, Teichtmeister S (2015) Minimization principles for the coupled problem of Darcy-Biot-type fluid transport in porous media linked to phase field modeling of fracture. J Mech Phys Solids 82:186–217
66. Zhou S, Zhuang X, Rabczuk T (2019) Phase field modeling of brittle compressive-shear fractures in rock-like materials: a new driving force and a hybrid formulation. Comput Methods Appl Mech Eng 355:729–752
67. Goswami S, Anitescu C, Chakraborty S, Rabczuk T (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor Appl Fract Mech 106:102447
68. Ren H, Zhuang X, Anitescu C, Rabczuk T (2019) An explicit phase field method for brittle dynamic fracture. Comput Struct 217:45–56
69. Ambrosio L, Tortorelli V (1990) Approximation of functional depending on jumps by elliptic functional via t-convergence. Commun Pure Appl Math 43:999–1036
70. Miehe C, Welschinger F, Hofacker M (2010) Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. Int J Numer Methods Eng 83:1273–1311
71. Bourdin B, Francfort GA, Marigo J-J (2008) The variational approach to fracture. J Elasticity 91:5–148
72. Miehe C, Hofacker M, Welschinger F (2010) A phase field model for rate-independent crack propagation: robust algorithmic implementation based on operator splits. Comput Methods Appl Mech Eng 199:2765–2778
73. Aravas N (2011) Plane-strain problems for a class of gradient elasticity models a stress function approach. J Elasticity 104:45–70
74. Khakalo S, Niiranen J (2017) Gradient-elastic stress analysis near cylindrical holes in a plane under bi-axial tension fields. Int J Solids Struct 110:351–366
75. Mindlin RD (1963) Microstructure in linear elasticity, technical report, Columbia University of New York Department of Civil Engineering and Engineering Mechanics
76. Vardoulakis I, Exadaktylos G, Aifantis E (1996) Gradient elasticity with surface energy: mode-iii crack problem. Int J Solids Struct 33:4531–4559
77. Ren H, Zhuang X, Rabczuk T (2020) Nonlocal operator method with numerical integration for gradient solid. Comput Struct 233:106235


**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.