

A neural network-based enrichment of reproducing kernel approximation for modeling brittle fracture

Jonghyuk Baek, Jiun-Shyan Chen ^{*}

Department of Structural Engineering, University of California, San Diego

ARTICLE INFO

Keywords:

Neural network
enrichment
reproducing kernel
fracture
damage

ABSTRACT

Numerical modeling of localizations is a challenging task due to the evolving rough solution in which the localization paths are not predefined. Despite decades of efforts, there is a need for innovative discretization-independent computational methods to predict the evolution of localizations. In this work, an improved version of the neural network-enhanced Reproducing Kernel Particle Method (NN-RKPM) is proposed for modeling brittle fracture. In the proposed method, a background reproducing kernel (RK) approximation defined on a coarse and uniform discretization is enriched by a neural network (NN) approximation under a Partition of Unity framework. In the NN approximation, the deep neural network automatically locates and inserts regularized discontinuities in the function space. The NN-based enrichment functions are then patched together with RK approximation functions using RK as a Partition of Unity patching function. The optimum NN parameters defining the location, orientation, and displacement distribution across location together with RK approximation coefficients are obtained via the energy-based loss function minimization. To regularize the NN-RK approximation, a constraint on the spatial gradient of the parametric coordinates is imposed in the loss function. Analysis of the convergence properties shows that the solution convergence of the proposed method is guaranteed. The NN enrichment allows the modeling of evolving cracks by a fixed coarse RK discretization without adaptive refinement for enhanced computational efficiency. The effectiveness of the proposed method is demonstrated by a series of numerical examples involving damage propagation and branching.

1. Introduction

Neural networks (NNs) have been shown to have powerful approximation ability [1,2]. The strong adaptivity and hidden information extraction capability have made deep neural networks a core element of machine learning in various applications. This feature also makes NNs appealing for solving challenging problems in computational mechanics. For example, data-driven computations for path-dependent material modeling [3–8], reduced order modeling [9,10], and parameter identification [11–13]. Additionally, the flexible adaptivity in NN allows an approximation space to be goal-specifically optimized. Utilizing this flexibility in the approximation space, NNs can be considered an alternative to traditional mesh-based methods in solving challenging problems involving localizations, such as fracture, for which special treatment is needed near the localizations.

Traditional approaches for fracture modeling can be divided into two broad categories: discrete crack approaches and diffuse crack

^{*} Corresponding author.

E-mail address: jsc137@ucsd.edu (J.-S. Chen).

approaches. The former category includes extended or generalized FEMs [14–16], partition of unity-based enrichment [17,18], and meshfree method with near-tip enrichment [19,20]. In these methods, strong discontinuities are directly inserted into the approximation, necessitating the detection and tracking of crack surfaces, significantly increasing the complexity of the computation for multidimensional problems. Nonlocal averaging [21], high order gradient models [22–24], and phase field methods [25–28] have been employed in the diffuse crack approaches. In this family of methods, nonlocal effects are typically introduced in the approximation or in the energy function, yielding diffused, regularized representation of cracks. This property enables traditional mesh-based or meshfree methods to approximate localizations without enrichment and the need for localization tracking. However, for sufficient accuracy, intense mesh refinement is required in the regions of localizations. For example, Geelen et al. [28] used an element size as small as one-tenth the width of the diffuse crack.

With their adaptive nature as an approximation, NNs provide a new paradigm in searching for solutions of mathematical models. Recently, NNs have been successfully applied as a solver of partial differential equations [11,12,29–33]. In physics-informed neural network (PINN) by Raissi et al. [11], the solution of a PDE is approximated by densely-connected deep neural networks with the residual-based loss function minimization. Haghighat and Juanes [34] developed the Python package SciANN for scientific computing using PINN and demonstrated its ability to capture strain and stress localization in a perfectly plastic material. More recently, PINNs have been extended to multi-physics problems [35,36]. However, one drawback of utilizing a deep neural network combined with a residual-based and collocated loss function is its computational cost, e.g., in [34], where 100 million unknown weights and biases were used. Samaniego et al. [29] demonstrated that potential-based loss functions produced superior results with significantly fewer unknowns than the residual-based loss function commonly used in PINN. Zhang et al. [30] proposed a deep neural network that reproduces standard approximations along with automatic refinement enabled by treating nodal positions as unknown network parameters, which, however, introduces sparsity into the neural network. Lu et al. [31], based on the universal approximation theorem [37], designed a new deep neural network architecture, in which the output of one deep neural network is multiplied by the output of another deep neural network, resulting effective approximations of nonlinear operators in partial differential equations.

Despite the growing interest in PINNs, there has been limited research on developing effective and computationally efficient NN-based approximation for modeling localizations. Baek et al. [33] proposed a neural network-enhanced reproducing kernel particle method (NN-RKPM) for modeling localizations. In this work, the approximation is constructed as the superposition of the NN approximation and the reproducing kernel (RK) approximation. For computational efficiency, NNs are limited to approximating localizations, while the RK approximation on a coarse and uniform discretization is employed to approximate the smooth solutions. In this approach, the NN approximation control parameters play the role in automatically capturing the location, orientation, and the localization profile at the localizations. These NN parameters are determined by the optimization of an energy-based loss function. In this work, we propose an improved version of NN-RKPM in which the NN approximation and the background RK approximation are patched together with Partition of Unity for ensured convergence. This approach is derived through an NN-based correction of standard RK shape functions. In the modified NN-RK approximation, the deep neural network automatically locates and inserts regularized discontinuities in the function space, and the NN enriched RK coefficient function provides varying magnitude of the discontinuity along the localization path. Additionally, convergence properties of the proposed method are analyzed.

The paper is organized as follows. In Section 2, the basic equations are provided, including the minimization problem for brittle fracture and the reproducing kernel particle method. In Section 3, a neural network-enriched Partition of Unity reproducing kernel approximation is proposed, along with convergence analysis and regularization technique. In Section 4, the implementation details including the neural network architecture and solution procedure are provided. This is followed by numerical examples in Section 5 and concluding remarks in Section 6.

2. Background

2.1. Minimization problem for fracture

For a domain $\Omega \in \mathbb{R}^d$ with the space dimension d and its boundary $\partial\Omega = \partial\Omega_g \cup \partial\Omega_h$ that consists of the Dirichlet boundary $\partial\Omega_g$ and the Neumann boundary $\partial\Omega_h$, let us consider the following minimization problem: for $\mathbf{u} \in H^1$, $\mathbf{u} = \mathbf{g}$ on $\partial\Omega_g$,

$$\min_{\mathbf{u}} \Pi(\mathbf{u}) = \int_{\Omega} \psi(\mathbf{u}) \, d\Omega - \int_{\Omega} \mathbf{u} \cdot \mathbf{b} \, d\Omega - \int_{\partial\Omega_h} \mathbf{u} \cdot \mathbf{h} \, d\Gamma, \quad (1)$$

where $\mathbf{u}$, $\psi(\mathbf{u})$, $\mathbf{b}$, and $\mathbf{h}$ are the displacement, energy density functional, body force, and traction, respectively. The energy density functional $\psi(\mathbf{u})$ has the following form:

$$\psi(\mathbf{u}) = g(\eta(\boldsymbol{\varepsilon}(\mathbf{u})))\psi_0^+(\mathbf{u}) + \psi_0^-(\mathbf{u}) + \bar{\psi}(\eta(\boldsymbol{\varepsilon}(\mathbf{u}))). \quad (2)$$

Herein, $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$, η , and g are the strain tensor, the (strain-dependent) damage variable, and the degradation function, respectively. Three energy density components ψ_0^+ , ψ_0^- , and $\bar{\psi}$ denote non-degraded tensile strain energy, compressive strain energy, and dissipation functional, respectively. The tensile and compressive strain energies are defined as

$$\begin{aligned}
\psi_0 &= \mu \bar{\epsilon}_i \bar{\epsilon}_i + \frac{\lambda}{2} \text{tr}(\bar{\epsilon})^2, \\
\psi_0^+ &= \mu \langle \bar{\epsilon}_i \rangle_+ \langle \bar{\epsilon}_i \rangle_+ + \frac{\lambda}{2} \langle \text{tr}(\bar{\epsilon}) \rangle_+^2, \\
\psi_0^- &= \psi_0 - \psi_0^+,
\end{aligned} \tag{3}$$

where the summation notation is adopted. In (3), $\bar{\epsilon}$, λ , and μ are principal strain, Lamé's first and second parameters, respectively. $\langle \cdot \rangle_+ = \max(\cdot, 0)$ and $\langle \cdot \rangle_- = \min(\cdot, 0)$ are additionally used. The stress is defined as

$$\boldsymbol{\sigma} = g(\eta(\boldsymbol{\epsilon})) \frac{\partial \psi_0^+}{\partial \boldsymbol{\epsilon}} + \frac{\partial \psi_0^-}{\partial \boldsymbol{\epsilon}}. \tag{4}$$

In this work, the damage variable, dissipation functional, and degradation function are defined as follows:

$$\eta = \frac{\psi_0^+}{\psi_0^+ + p} \tag{5}$$

$$\bar{\psi} = p\eta^2, \tag{6}$$

$$g = (1 - \eta)^2, \tag{7}$$

where p is a fracture energy-dependent material property. The adopted dissipation functional and degradation function in Eqs. (6) and (7) are the same as what is used in Miehe et al. [25] except the absence of the higher order term $\mathcal{O}(\nabla \eta^2)$ in the dissipation functional in Eq. (6). Therefore, it is straightforward to show that the damage model in Eqs. (5)-(7) is variationally consistent, i.e., for $\mathbf{u} \in H^1$, $\mathbf{u} = \mathbf{g}$ on $\partial\Omega_g$, for all $\delta \mathbf{u} \in H^1$, $\delta \mathbf{u} = \mathbf{0}$ on $\partial\Omega_g$,

$$\delta \Pi = \int_{\Omega} \delta \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\sigma}(\boldsymbol{\epsilon}) \, d\Omega - \int_{\Omega} \delta \mathbf{u} \cdot \mathbf{b} \, d\Omega - \int_{\partial\Omega_h} \delta \mathbf{u} \cdot \mathbf{h} \, d\Gamma = 0, \tag{8}$$

which leads to the following balance equation:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \text{ in } \Omega, \tag{9}$$

with the boundary conditions

$$\mathbf{u} = \mathbf{g} \text{ on } \partial\Omega_g, \tag{10}$$

$$\nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{h} \text{ on } \partial\Omega_h, \tag{11}$$

where $\mathbf{n}$ denotes the surface normal vector.

To achieve the irreversibility of the damage, a history variable

$$\mathcal{H} = \max \left(\max_{t \in [0, T]} \{ \psi_0^+(\boldsymbol{\epsilon}) - \psi_c \}, 0 \right) \tag{12}$$

is employed to describe the damage variable:

$$\eta = \frac{\mathcal{H}}{\mathcal{H} + p}. \tag{13}$$

For Eq. (12), the critical fracture energy ψ_c is defined as

$$\psi_c = \frac{f_t}{2E} \tag{14}$$

with the tensile strength of material f_t and Young's modulus E . The model parameter p takes the following form

$$p = \frac{\mathcal{G}_c}{\ell}, \tag{15}$$

with critical energy release rate $\mathcal{G}_c$ and length scale parameter ℓ . To take mixed mode fracture into account, we adopt the $\mathcal{F}$ -criterion [38], with the mode I critical energy release rate $\mathcal{G}_{cl}$ and the mode II critical energy release rate $\mathcal{G}_{cII}$:

$$\mathcal{F} \equiv \frac{\psi_0^+}{\mathcal{G}_c} \approx \frac{\psi_I^+}{\mathcal{G}_{cl}} + \frac{\psi_{II}^+}{\mathcal{G}_{cII}}, \tag{16}$$

with

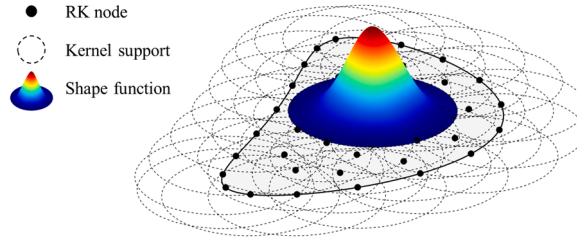


Fig. 1. Illustration of RK discretization and shape function

$$\psi_I^+ = \frac{\lambda}{2} \left\langle \sum \bar{\epsilon}_i \right\rangle_+^2, \quad (17)$$

$$\psi_{II}^+ = \mu \langle \bar{\epsilon}_i \rangle_+ \langle \bar{\epsilon}_i \rangle_+. \quad (18)$$

Eq. (16) leads to the following critical energy release rate:

$$\mathcal{G}_c = \frac{\psi_0^+}{\psi_I^+ / \mathcal{G}_{cl} + \psi_{II}^+ / \mathcal{G}_{cII}}. \quad (19)$$

Note that Eq. (19) implies $\mathcal{G}_c = \mathcal{G}_{cl}$ for pure mode I fracture when $\psi_0^+ = \psi_I^+$ and $\mathcal{G}_c = \mathcal{G}_{cII}$ for pure mode II fracture when $\psi_0^+ = \psi_{II}^+$.

Remark 1.1. With $\mathcal{G}_c$ defined in Eq. (19) which is a function of strain, the functional Π defined in Eq. (1) is not a minimization functional for the Euler-Lagrange Eq. (9). Therefore, in this work, we solve the minimization problem in Eq. (1) and the $\mathcal{G}_c$ calculation in Eq. (19) in a staggered manner.

Remark 1.2. Different from the phase field fracture methods, the damage model described in this section is a local model in the absence of the higher order term in the dissipation functional. Therefore, there is possibility of the loss of ellipticity and the discretization-dependence of the numerical solution. This issue will be addressed in Section 3.6.

2.2. Reproducing kernel particle method for background approximation

Here we review the standard reproducing kernel particle method (RKPM) that is used to approximate smooth part of the solution in the proposed approach (see Section 3).

2.2.1. Reproducing kernel approximation

Let Ω be a domain discretized by NP nodes with nodal coordinate $\{\mathbf{x}_I\}_{I \in \mathcal{I}}$ with a node set $\mathcal{I} = \{1, \dots, NP\}$. The reproducing kernel (RK) approximation, $u^{RK}(\mathbf{x})$, of a function $u(\mathbf{x})$ is

$$u^{RK}(\mathbf{x}) = \sum_{I \in \mathcal{I}} \Psi_I(\mathbf{x}) d_I, \quad (20)$$

with an RK shape function $\Psi_I(\mathbf{x})$ and a generalized nodal coefficient d_I . The RK shape function is a correction of a kernel function, $\Phi_a(\mathbf{x} - \mathbf{x}_I)$, defined on the compact support of node I with a support size of a :

$$\Psi_I(\mathbf{x}) = C_I(\mathbf{x}) \Phi_a(\mathbf{x} - \mathbf{x}_I), \quad (21)$$

where the kernel correction function $C_I(\mathbf{x})$ is defined as

$$C_I(\mathbf{x}) \equiv \left\{ \sum_{|\alpha| \leq n} (\mathbf{x} - \mathbf{x}_I)^\alpha b_\alpha(\mathbf{x}) \right\}, \quad (22)$$

where $(\mathbf{x} - \mathbf{x}_I)^\alpha$ is a basis function, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ is a multi-dimensional index, and $|\alpha| \equiv \sum_{i=1}^d \alpha_i$. $\mathbf{x}^\alpha$ is defined as

$$\mathbf{x}^\alpha \equiv x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_d^{\alpha_d}. \quad (23)$$

The coefficients, $b_\alpha(\mathbf{x})$, are obtained by solving the following set of reproducing conditions:

$$\sum_{I \in \mathcal{I}} \Psi_I(\mathbf{x}) \mathbf{x}_I^\alpha = \mathbf{x}^\alpha, \quad |\alpha| \leq n. \quad (24)$$

The results RK shape function takes the following explicit form:

$$\Psi_I(\mathbf{x}) = \mathbf{H}^T(0) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \Phi_a(\mathbf{x} - \mathbf{x}_I), \quad (25)$$

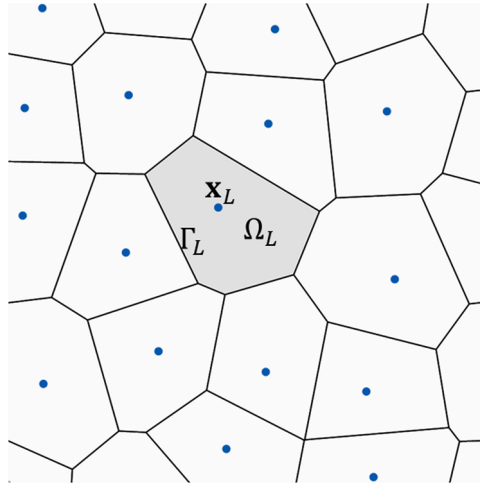


Fig. 2. Conforming integration cell used in SCNI: Ω_L , Γ_L , and $\mathbf{x}_L$ denote the domain, the boundary, and the centroid of the integration cell L , respectively.

where the moment matrix $\mathbf{M}(\mathbf{x})$ and the basis vector $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ are defined as

$$\mathbf{M}(\mathbf{x}) = \sum_{I \in \mathcal{I}} \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \Phi_a(\mathbf{x} - \mathbf{x}_I), \quad (26)$$

$$\mathbf{H}(\mathbf{x} - \mathbf{x}_I) = [1, (x_1 - x_{1I}), (x_2 - x_{2I}), (x_3 - x_{3I}), \dots, (x_3 - x_{3I})^n]^T. \quad (27)$$

The kernel function $\Phi_a(\mathbf{x} - \mathbf{x}_I)$ determines the order of continuity, while the basis vector $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ determines the polynomial completeness. Thus, it is straightforward to introduce high order continuity into the approximation space, independent of the basis order, which makes the RK approximation more appealing for approximating the smooth part of solution than the C^0 interpolation-type approximations used in finite element methods. Fig. 1 shows a smooth RK shape function constructed on the linear basis.

For a quasi-uniform RK points distribution, the following global error estimation of standard RK approximation u^{RK} holds, for $u \in H^r$, [39]

$$\|u^{RK} - u\|_{L^2} \leq Cka^\gamma \|u\|_{p+1, \Omega}, \quad (28)$$

where a , C , k , p , and $\gamma = \min(p+1-l, r-l)$ are the support size, a generic constant, the number of overlapping points, the order of RK basis, and the convergence rate, respectively.

2.2.2. Stabilized conforming nodal integration

When the Gauss integration (GI) is used for RKPM, a significantly high-order rule is required to yield optimal solution convergence, due to the rational shape function given in Eq. (25). This, in turn, leads to a significant increase in computational cost. To address this issue, the stabilized conforming nodal integration (SCNI) was proposed in [40]. SCNI enables optimal solution convergence for RKPM with a linear basis by satisfying the linear integration constraint. Compared to high-order GI, SCNI is computationally much more efficient as it eliminates the need to evaluate direct derivatives of RK shape functions at a large number of integration points. Additionally, Wei and Chen [41] show that the strain smoothing employed in SCNI helps to suppress spurious stress oscillation that can arise in localization problems. For this reason, SCNI is utilized to perform the domain integration required in Eq. (1).

In SCNI, the domain is partitioned into N_{IC} conforming smoothing cells, such as Voronoi cells, as illustrated in Fig. 2 where N_{IC} denotes the number of smoothing cells. Note that, while N_{IC} coincides with the number of particles for standard meshfree methods, the smoothing cells can be further refined to improve accuracy.

The integration of the loss function by SCNI is performed as follows:

$$\int_{\Omega} \psi(\mathbf{u}^h, \nabla \mathbf{u}^h) d\Omega \approx \sum_L^{N_{IC}} \psi(\mathbf{u}^h(\mathbf{x}_L), \tilde{\nabla} \mathbf{u}^h(\mathbf{x}_L)) V_L, \quad (29)$$

where $\tilde{\nabla} \mathbf{u}^h$ is the smoothed gradient of $\mathbf{u}$ defined as

$$\tilde{\nabla} \mathbf{u}^h(\mathbf{x}_L) \equiv \frac{1}{V_L} \int_{\Gamma_L} \mathbf{u}^h(\mathbf{x}) \otimes \mathbf{n}(\mathbf{x}) d\Gamma \approx \frac{1}{V_L} \sum_{k=1}^{N_{reg}^L} \mathbf{u}^h(\mathbf{x}_L^k) \otimes \mathbf{n}_L^k, \quad (30)$$

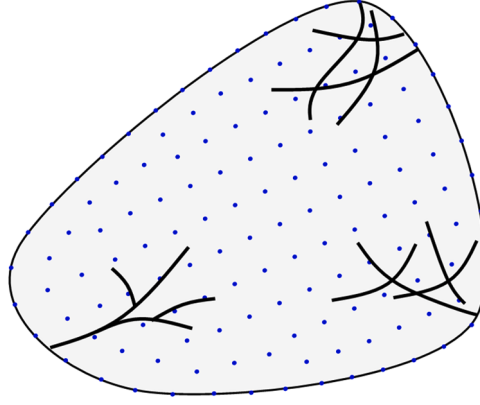


Fig. 3. Schematic illustration of the NN-RK approximation: quasi-uniform background RK node distribution (blue dots) for smooth solution approximation and NN enrichment of the solution space for capturing localizations (black solid curves).

where V_L , $\mathbf{x}_L^k$, $\mathbf{n}_L^k$, and N_{seg}^L are the cell volume, the centroid of k -th boundary segment, the surface normal of k -th boundary segment, and the number of boundary segments of the integration cell L , respectively.

3. Neural network-enhanced reproducing kernel approximation

Fig. 3 schematically illustrates a domain discretization by quasi-uniformly distributed background RK nodes, along with the evolving localizations in the domain. It is expected that the true solution would be rough near localizations and smooth in the remaining part of the domain. As discussed in Section 2.2.1, the RK approximation is intended to capture the smooth part of the solution. With the enrichment function (to be constructed) near the evolving localizations, the total solution is constructed by superposing a background RK approximation $u^{RK}(\mathbf{x})$ and a neural network (NN) approximation $u^{NN}(\mathbf{x})$ as follows: for $\mathbf{x} \in \Omega$,

$$u^h(\mathbf{x}) = u^{RK}(\mathbf{x}) + u^{NN}(\mathbf{x}), \quad (31)$$

where $u^h(\mathbf{x})$ is an NN-enhanced RK (NN-RK) approximation. With this construction, uniform RK discretization is considered as a background discretization, and the localized solution will be represented by the NN approximation. The NN-RK approximation utilizes the RK approximation's flexibility in selecting the order of continuity and the order of monomial bases.

3.1. A neural network-based correction of RK approximation

In this section, we derive the NN-RK approximation through a neural network-based correction (NN-correction) of an RK approximation. Let Ω be a domain discretized by NP background RK nodes with nodal coordinate $\{\mathbf{x}_I\}_{I \in \mathcal{I}}$ in a node set $\mathcal{I} = \{1, \dots, NP\}$. In addition, define a node subset $\overline{\mathcal{I}}$ that contains the nodes with the associated RK shape functions to be corrected near localization. In this work, $\overline{\mathcal{I}} = \{J \mid \exists \mathbf{x} \in \text{supp}(\Psi_J), \psi_0^+(\mathbf{x}) \geq \kappa \psi_c\}$ with $\kappa = 0.5$ is applied. We start with the following NN-corrected RK approximation:

$$u^h(\mathbf{x}) = \sum_{I \in \mathcal{I}} \overline{\Psi}_I(\mathbf{x}) \overline{d}_I, \quad (32)$$

where the NN-corrected RK shape function $\overline{\Psi}_I(\mathbf{x})$ is defined as follows:

$$\overline{\Psi}_I(\mathbf{x}) = \begin{cases} \overline{C}_I(\mathbf{x}) \Psi_I(\mathbf{x}), & I \in \overline{\mathcal{I}}, \\ \Psi_I(\mathbf{x}), & I \in \mathcal{I} \setminus \overline{\mathcal{I}}, \end{cases} \quad (33)$$

where $\Psi_I(\mathbf{x})$ and $\overline{C}_I(\mathbf{x})$ denote the original RK shape function defined in Section 2.2.1 and an NN-correction function, respectively. The NN-correction function takes the following form of a neural network with n neurons possessed by the last hidden layer:

$$\overline{C}_I(\mathbf{x}) \equiv \overline{b}_I + \sum_{k=1}^n \overline{w}_{IK} \zeta_{IK}(\mathbf{x}), \quad (34)$$

where $\overline{b}_I$, $\overline{w}_{IK}$, and $\zeta_{IK}(\mathbf{x})$ denote bias, weight, and last hidden layer's output. By substituting Eqs. (33) and (34) into (32) and defining $\overline{d}_I = \overline{b}_I \overline{d}_I$ and $\overline{w}_{IK}^C = \overline{w}_{IK} \overline{d}_I$, we have a general expression of NN-RK approximation as follows:

$$u^h(\mathbf{x}) = u^{RK}(\mathbf{x}) + u^{NN}(\mathbf{x}), \quad (35)$$

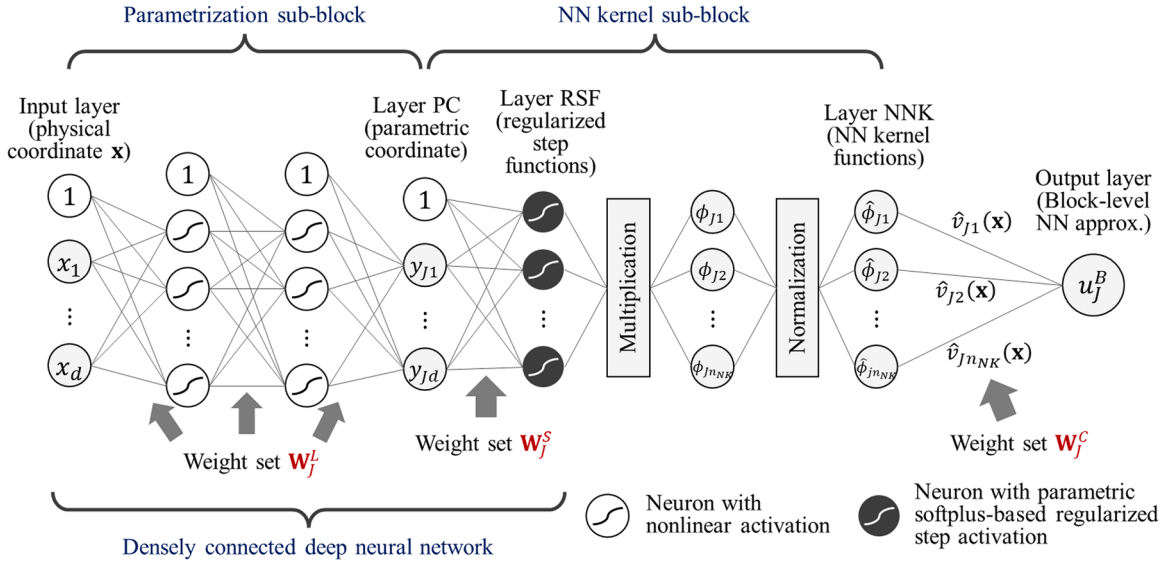


Fig. 4. Modified neural network architecture of J -th NN block. The unknown parameters introduced in each part are denoted in red color.

$$u^{RK} = \sum_{I \in \mathcal{I}} \Psi_I(\mathbf{x}) d_I, \quad (36)$$

$$u^{NN} = \sum_{I \in \mathcal{I}} \sum_{k=1}^n \Psi_I(\mathbf{x}) \zeta_{IK}(\mathbf{x}) w_{IK}^C. \quad (37)$$

Remark 3.1. The background RK approximation $u^{RK}(\mathbf{x})$ in Eq. (36) is a standard RK approximation based on a polynomial RK basis. Meanwhile, the NN approximation $u^{NN}(\mathbf{x})$ in Eq. (37) contains nonstandard *adaptive* basis functions, which enables it to capture localized material responses with a coarse background RK discretization.

Remark 3.2. As the RK shape functions possess the property of partition of unity, the NN-RK approximation

$$u^h(\mathbf{x}) = u^{RK}(\mathbf{x}) + u^{NN}(\mathbf{x}) = \sum_{I \in \mathcal{I}} \Psi_I(\mathbf{x}) \left(d_I + \sum_{k=1}^n \zeta_{IK}(\mathbf{x}) w_{IK}^C \right), \text{ with } w_{IK}^C = 0 \forall I \in \mathcal{I} \setminus \overline{\mathcal{I}} \quad (38)$$

can be viewed as patching the RK and NN approximations under the Partition of Unity framework.

Remark 3.3. In Eq. (37), $\zeta_{IK}(\mathbf{x})$ is the activated output of K -th neuron in the last hidden layer of a neural network associated with node I . By having $\zeta_{IK}(\mathbf{x}) \equiv \zeta_K(\mathbf{x})$ for all $I \in \overline{\mathcal{I}}$, $\zeta_K(\mathbf{x})$ is detached from a specific background node and becomes a flexible *foreground* quantity. Then, the NN approximation in Eq. (37) can be rewritten as follows:

$$u^{NN} = \sum_{K=1}^n \zeta_K(\mathbf{x}) v_K(\mathbf{x}), \quad (39)$$

$$v_K(\mathbf{x}) \equiv \sum_{I \in \overline{\mathcal{I}}} \Psi_I(\mathbf{x}) w_{IK}^C. \quad (40)$$

Remark 3.4. The neural network to generate $\zeta_{IK}(\mathbf{x})$ can be either a traditional or a nonstandard neural network. In Section 3.2, we present a modified deep neural network designed to effectively capture localizations.

3.2. Block-level neural network approximation

In this work, we introduce a modified deep neural network to increase the sparsity of the network architecture, improve the interpretability, and capture localizations effectively. In this regard, the following block-level NN approximation is introduced.

$$u^{NN} = \sum_{J=1}^{n_B} u_J^B(\mathbf{x}), \quad (41)$$

where n_B is the number of NN blocks, and the block-level NN approximation $u_j^B(\mathbf{x})$ is defined as follows:

$$u_j^B(\mathbf{x}) = \sum_{K=1}^{n_{NK}} \hat{\phi}_{JK}(\mathbf{x}) \hat{v}_{JK}(\mathbf{x}), \quad (42)$$

$$\hat{v}_{JK}(\mathbf{x}) = \sum_{I \in \mathcal{I}} \Psi_I(\mathbf{x}) \hat{w}_{JK}^C, \quad (43)$$

where $\hat{\phi}_{JK}(\mathbf{x})$ and n_{NK} are K -th NN kernel function in J -th NN block and the number of NN kernel functions per NN block, respectively. Note that Eqs. (41)–(43) are shown to be equivalent to Eqs. (39) and (40) by flattening the indices JK in Eqs. (42) and (43) into K .

Fig. 4 illustrates the modified network architecture of J -th NN block, for which the construction is made so that the neural network approximation can capture complicated localization topologies effectively. Also, the construction of the neural network at the block level significantly increases the sparsity of the weight matrices, compared to the densely connected standard deep neural networks utilized in many previous studies in literature [29,34]. As shown in Fig. 4, three sets of unknown parameters are involved in the NN approximation: the location-control weight set $\mathbf{W}_J^L$, the shape-control weight set $\mathbf{W}_J^S$ as well as the NN-correction weight set $\mathbf{W}_J^C = \{\{\hat{w}_{JK}^C\}_{I \in \mathcal{I}}\}_{K=1}^{n_{NK}}$ in Eq. (43). These parameters are to be automatically determined by solving the minimization problem (Eq. (1)). Details on the sub-blocks described in Fig. 4 and their associated unknown parameters are explained in the following subsections.

3.3. Parametrization sub-block

As shown in Fig. 4, the parametric coordinate $\mathbf{y}_J$ in Layer PC is the output of the parametrization sub-block, which is an intermediate variable of a densely connected deep neural network $\mathcal{N}: \mathbf{x} \rightarrow \mathbf{y}_J$ that takes $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y}_J \equiv \mathbf{y}(\mathbf{x}; \mathbf{W}_J^L) \in \mathbb{R}^d$ as its input and output, respectively. The parametrization projects complicated localization patterns onto a parametric space, so that complicated localizations can be captured with NN kernel functions in a simple mathematical form. With n_{HL} hidden layers, the function $\mathbf{y}(\mathbf{x}; \mathbf{W}_J^L)$ is defined as

$$\mathbf{y}(\mathbf{x}; \mathbf{W}_J^L) = \mathbf{f}\left(\cdot; \left\{\mathbf{w}_{J(n_{HL}+1)}^L, b_{J(n_{HL}+1)}^L\right\}\right) \circ \mathbf{h}\left(\cdot; \left\{\mathbf{w}_{Jn_{HL}}^L, b_{Jn_{HL}}^L\right\}\right) \circ \cdots \circ \mathbf{h}\left(\cdot; \left\{\mathbf{w}_{J1}^L, b_{J1}^L\right\}\right) \quad (44)$$

with

$$\mathbf{h}(\xi; \{\mathbf{w}_{Jl}^L, b_{Jl}^L\}) = \sigma(\mathbf{f}(\xi; \{\mathbf{w}_{Jl}^L, b_{Jl}^L\})), \quad (45)$$

$$\mathbf{f}(\xi; \{\mathbf{w}_{Jl}^L, b_{Jl}^L\}) = \mathbf{w}_{Jl}^L \xi + b_{Jl}^L. \quad (46)$$

In Eq. (44), $\mathbf{w}_{Jl}^L$ and b_{Jl}^L denote weight and bias of layer l , respectively, and the location-control parameter set $\mathbf{W}_J^L$ in Fig. 4 is defined as $\mathbf{W}_J^L = \{\mathbf{w}_{Jl}^L, b_{Jl}^L\}_{l=1}^{n_{HL}+1}$. In Eq. (45), $\sigma(\cdot)$ denote an activation function. In this work, the hyperbolic tangent activation function is used.

3.4. NN kernel function

As shown in Fig. 4, the NN kernel functions $\hat{\phi}_{JK}(\mathbf{x})$ in Layer NNC is the outcome of the normalization of unnormalized NN kernel functions $\phi_{JK}(\mathbf{x})$. The normalization is defined as

$$\hat{\phi}_{JK}(\mathbf{x}) = \frac{\phi_{JK}(\mathbf{x})}{\sum_{I=1}^{n_B} \sum_{L=1}^{n_{NK}} \phi_{IL}(\mathbf{x})}, \quad (47)$$

and the NN kernel function $\phi_{JK}(\mathbf{x})$ is defined as

$$\phi_{JK}(\mathbf{x}) = \prod_{\alpha=1}^d \prod_{i=1}^2 \bar{\phi}_i(y_{J\alpha}; \{\bar{y}_{ai}^{JK}, c_{ai}^{JK}, \beta_{ai}^{JK}\}), \quad (48)$$

where $\bar{\phi}_i$ and $\{\bar{y}_{ai}^{JK}, c_{ai}^{JK}, \beta_{ai}^{JK}\}$ denote a regularized step function and shape-control parameters, respectively. The shape-control weight set $\mathbf{W}_J^S$ in Fig. 4 is defined as $\mathbf{W}_J^S = \{\{\{\bar{y}_{ai}^{JK}, c_{ai}^{JK}, \beta_{ai}^{JK}\}_{\alpha=1}^d\}_{i=1}^2\}_{K=1}^{n_{NK}}$. In this work, the regularized step function is constructed based on the parametric softplus activation function S defined as follows:

$$\bar{\phi}_i(y; \{\bar{y}_i, c_i, \beta_i\}) = S\left(z_i(y) + \frac{1}{2}; \beta_i\right) - S\left(z_i(y) - \frac{1}{2}; \beta_i\right), \quad (49)$$

$$z_i(y) = (-1)^i (y - \bar{y}) / c, \quad i = 1, 2, \quad (50)$$

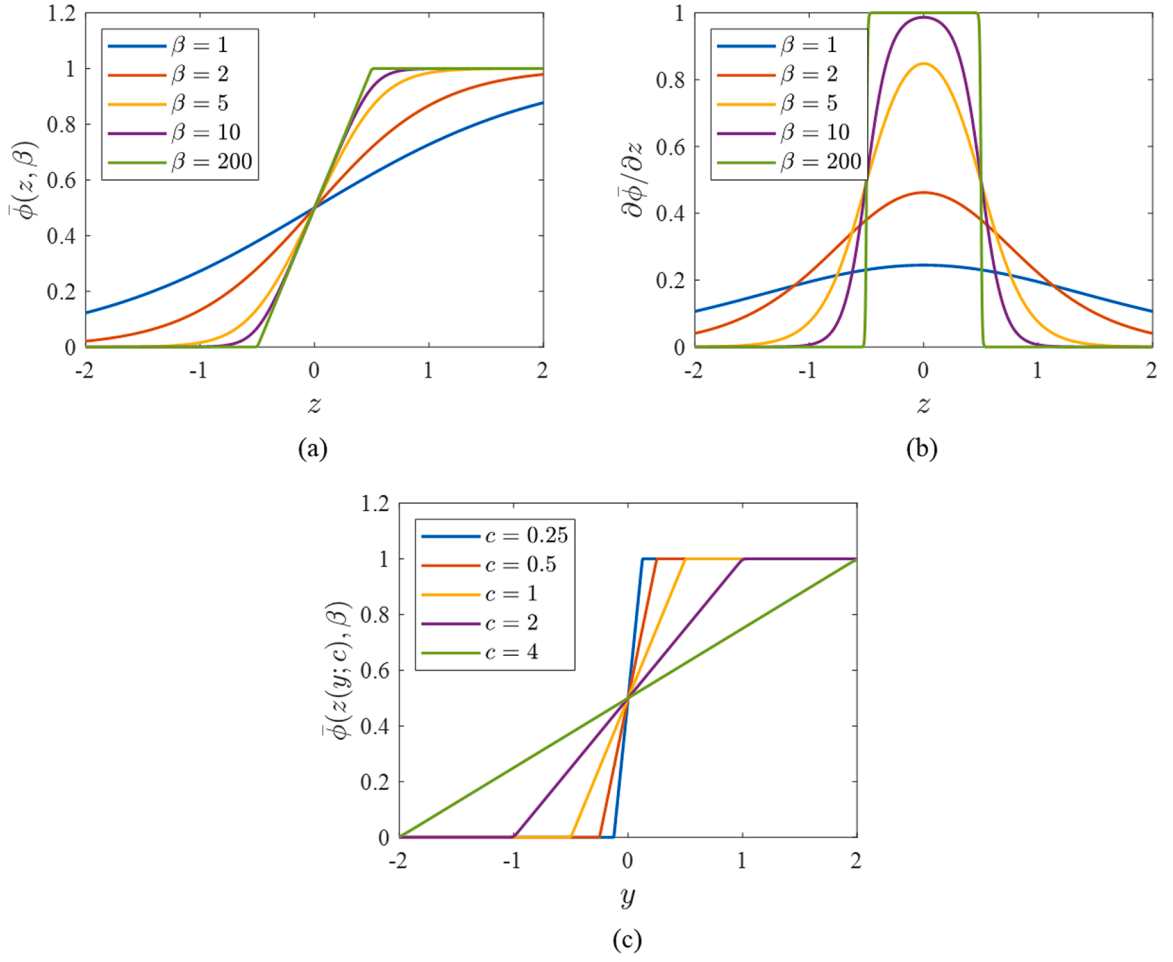


Fig. 5. The influence of the control parameters on solution transition: (a) the influence of β on $\bar{\phi}$, (b) the influence of β on $\partial \bar{\phi} / \partial z$, and (c) the influence of c on $\bar{\phi}$ with $\beta = 200$

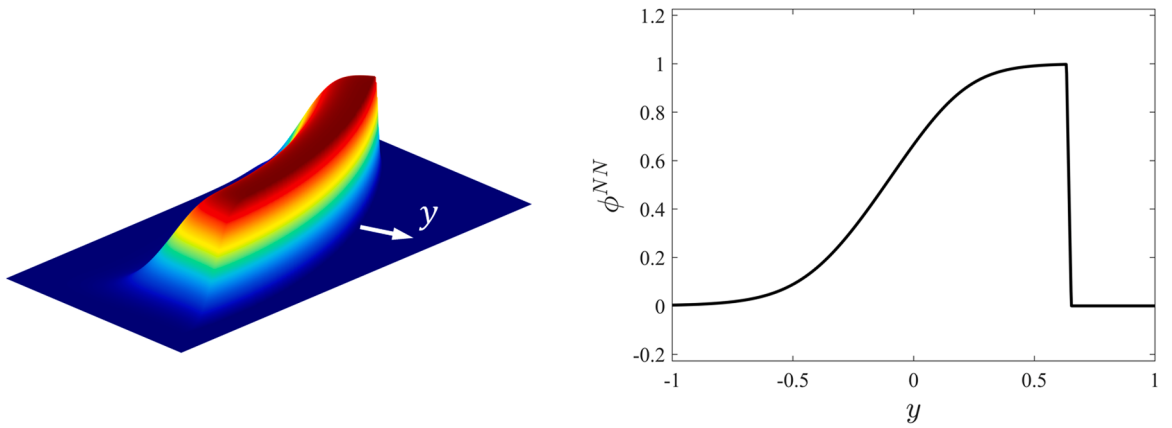


Fig. 6. Schematic illustration of an NN kernel function: (left) two-dimensional NN kernel function ϕ and (right) its cross-sectional value across y .

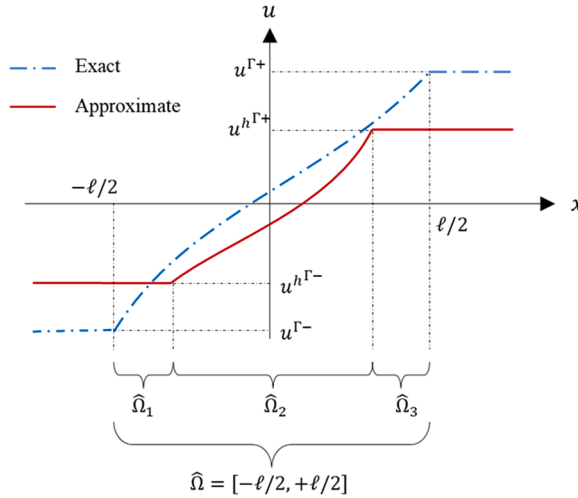


Fig. 7. Arbitrary u with a sharp transition occurring in the transition zone $\hat{\Omega}$ and its approximation u^h with a sharp transition occurring in $\hat{\Omega}_2$

$$S(z; \beta) = \frac{1}{\beta} \log(1 + e^{\beta z}). \quad (51)$$

In Eqs. (49)–(51), β_i controls the sharpness in the transition of derivative as shown in Fig. 5 (a-b), and c_i controls the sharpness of the solution transition as shown in Fig. 5 (c). In addition, $\bar{y}_i$ influences the support of $\bar{\phi}_i$. Note that $\bar{\phi}_i$ is the output of Layer RSF in Fig. 4, and $(1/c_i)$ and $(-\bar{y}_i/c_i)$ are respectively the weight and the bias of Layer RSF. Fig. 6 shows a schematic illustration of a two-dimensional NN kernel which possesses a sharp transition in direction y . Interested readers refer to [33] for more details on the NN kernel functions.

3.5. Convergence properties

An error bound of the proposed NN-RK approximation is estimated. Let $\hat{\Omega}$ be the transition zone near the localization domain. Then, we have

$$\|u^h - u\|_{0,\Omega} \leq \|u^h - u\|_{0,\Omega \setminus \hat{\Omega}} + \|u^h - u\|_{0,\hat{\Omega}}. \quad (52)$$

As shown in Fig. 7, we consider an arbitrary u with a sharp transition occurring in $\hat{\Omega} = [-\ell/2, +\ell/2]$ and its approximation u^h with a transition occurring in $\hat{\Omega}_2$. For both u and u^h , it is assumed that there are weak discontinuities on the boundaries of the transition zones. For brevity, let us introduce the following function w :

$$w(\chi; \xi) \equiv \frac{\llbracket \xi \rrbracket}{\ell} \chi + \langle \langle \xi \rangle \rangle, \quad (53)$$

where $\llbracket \xi \rrbracket \equiv \xi^+ - \xi^-$ and $\langle \langle \xi \rangle \rangle \equiv (\xi^+ + \xi^-)/2$ are a difference operator and an average operator, respectively, with $\xi^+ \equiv \xi(x = +\ell/2)$ and $\xi^- \equiv \xi(x = -\ell/2)$. Using Eq. (53), the true solution u in the transition domain $\hat{\Omega}$ can be written in a parametric coordinate y^u as

$$u(x) = w(y^u(x); u^\Gamma), \quad (54)$$

where u^Γ is the value of u on the boundary of $\hat{\Omega}$, and, from Eqs. (53) and (54), $y^u(x)$ is obtained as

$$y^u(x) = \frac{\ell}{\llbracket u^\Gamma \rrbracket} (u(x) - \langle \langle u^\Gamma \rangle \rangle). \quad (55)$$

Similarly, the approximated solution $u^h(x)$ in the transition domain $\hat{\Omega}$ is written in an approximated parametric coordinate Y as

$$u^h(x) = w(Y(x); u^{h\Gamma}), \quad (56)$$

with

$$Y(x) = \begin{cases} -\ell/2, & x \in \hat{\Omega}_1 \\ y(x), & x \in \hat{\Omega}_2 \\ \ell/2, & x \in \hat{\Omega}_3 \end{cases} \quad (57)$$

where $y(x)$ is the neural network-based parametrization defined in Eq. (44), and $u^{h\Gamma}$ is the value of u^h on the boundary of $\hat{\Omega}$. In Eq. (57),

the subdomains are defined as $\widehat{\Omega}_1 = \{x \mid y(-\ell/2) \leq y(x) \leq -\ell/2\}$, $\widehat{\Omega}_2 = \{x \mid -\ell/2 < y(x) \leq \ell/2\}$, and $\widehat{\Omega}_3 = \{x \mid \ell/2 < y(x) \leq \ell/2\}$. Note that, with $\beta \rightarrow \infty$, the NN kernel function defined in Eqs. (48)–(50) introduces weak discontinuities on $y(x) = \pm\ell/2$.

With Eqs. (54) and (56), the last term in Eq. (52) becomes

$$\begin{aligned} \|u^h - u\|_{0,\widehat{\Omega}} &= \|w(Y(x); u^{h^{\Gamma^-}}) - w(y^\mu(x); u^\Gamma)\|_{0,\widehat{\Omega}} \\ &= \|w(Y(x); u^{h^{\Gamma^-}}) - w(Y(x); u^\Gamma) + w(Y(x); u^\Gamma) - w(y^\mu(x); u^\Gamma)\|_{0,\widehat{\Omega}} \\ &= \|w(Y(x); u^{h^{\Gamma^-}} - u^\Gamma) + w(Y(x); u^\Gamma) - w(y^\mu(x); u^\Gamma)\|_{0,\widehat{\Omega}} \\ &\leq \|w(Y(x); u^{h^{\Gamma^-}} - u^\Gamma)\|_{0,\widehat{\Omega}} + \|(Y(x); u^\Gamma) - w(y^\mu(x); u^\Gamma)\|_{0,\widehat{\Omega}}. \end{aligned} \quad (58)$$

The first term on the right-hand side of Eq. (58) is bounded as follows:

$$\begin{aligned} \|w(Y(x); u^{h^{\Gamma^-}} - u^\Gamma)\|_{0,\widehat{\Omega}} &= \|([u^{h^{\Gamma^-}} - u^\Gamma]/\ell)Y(x) + \langle\langle u^{h^{\Gamma^-}} - u^\Gamma \rangle\rangle\|_{0,\widehat{\Omega}} \\ &\leq \| |u^{h^{\Gamma^-}} - u^{\Gamma^-}| + |u^{h^{\Gamma^+}} - u^{\Gamma^+}| \|_{0,\widehat{\Omega}} \\ &\leq \|u^{h^{\Gamma^-}} - u^{\Gamma^-}\|_{0,\widehat{\Omega}} + \|u^{h^{\Gamma^+}} - u^{\Gamma^+}\|_{0,\widehat{\Omega}} \\ &= \ell^{1/2}(|u^{h^{\Gamma^-}} - u^{\Gamma^-}| + |u^{h^{\Gamma^+}} - u^{\Gamma^+}|) \end{aligned} \quad (59)$$

The second term on the right-hand side of Eq. (58) is bounded as follows:

$$\begin{aligned} \|w(Y(x); u^\Gamma) - w(y^\mu(x); u^\Gamma)\|_{0,\widehat{\Omega}} &= \left\| \frac{[[u^\Gamma]]}{\ell} (Y(x) - y^\mu(x)) \right\|_{0,\widehat{\Omega}} \\ &= \frac{[[[u^\Gamma]]]]}{\ell} \|Y(x) - y^\mu(x)\|_{0,\widehat{\Omega}} \\ &\leq \frac{[[[u^\Gamma]]]]}{\ell} \|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}. \end{aligned} \quad (60)$$

Therefore, for $\widehat{\Omega}$, the following error bound is obtained.

$$\|u^h - u\|_{0,\widehat{\Omega}} \leq \ell^{1/2}(|u^{h^{\Gamma^-}} - u^{\Gamma^-}| + |u^{h^{\Gamma^+}} - u^{\Gamma^+}|) + \frac{[[[u^\Gamma]]]]}{\ell} \|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}. \quad (61)$$

For multi-dimensions, we have

$$\|u^h - u\|_{0,\widehat{\Omega}} \leq \ell^{1/2} \|u^h - u\|_{0,\widehat{\Gamma}} + \frac{[[[u^\Gamma]]]]}{\ell} \|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}, \quad (62)$$

where $\widehat{\Gamma} \equiv \partial\widehat{\Omega} \setminus \partial\Omega$ denotes the interface of weak discontinuity. Using the Sobolev trace inequality and Eq. (28), the first term on the right-hand side of Eq. (62) is bounded as follows: with a generic constant $\widehat{C}$ and $\widehat{\widehat{C}}$,

$$\begin{aligned} \|u^h - u\|_{0,\widehat{\Gamma}} &\leq \|u^h - u\|_{0,\partial(\Omega \setminus \widehat{\Omega})} \leq \widehat{C} \|u^h - u\|_{0,\Omega \setminus \widehat{\Omega}}^{1/2} \|u^h - u\|_{1,\Omega \setminus \widehat{\Omega}}^{1/2} \\ &\leq \widehat{\widehat{C}} k a^{\widehat{\gamma}} |u|_{p+1,\Omega \setminus \widehat{\Omega}}, \end{aligned} \quad (63)$$

where $\widehat{\gamma} = \max(p+0.5, \widetilde{r})$ where $\widetilde{r}$ and p denotes the regularity of u in $\Omega \setminus \widehat{\Omega}$ and the order of basis of the background RK discretization, respectively. With Eqs. (28), (62), and (63), the global error Eq. (52) has the following error bound:

$$\|u^h - u\|_{0,\Omega} \leq (Ca^\gamma + \widehat{\widehat{C}} a^{\widehat{\gamma}} k |u|_{p+1,\Omega \setminus \widehat{\Omega}} + \frac{[[[u^\Gamma]]]]}{\ell} \|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}, \quad (64)$$

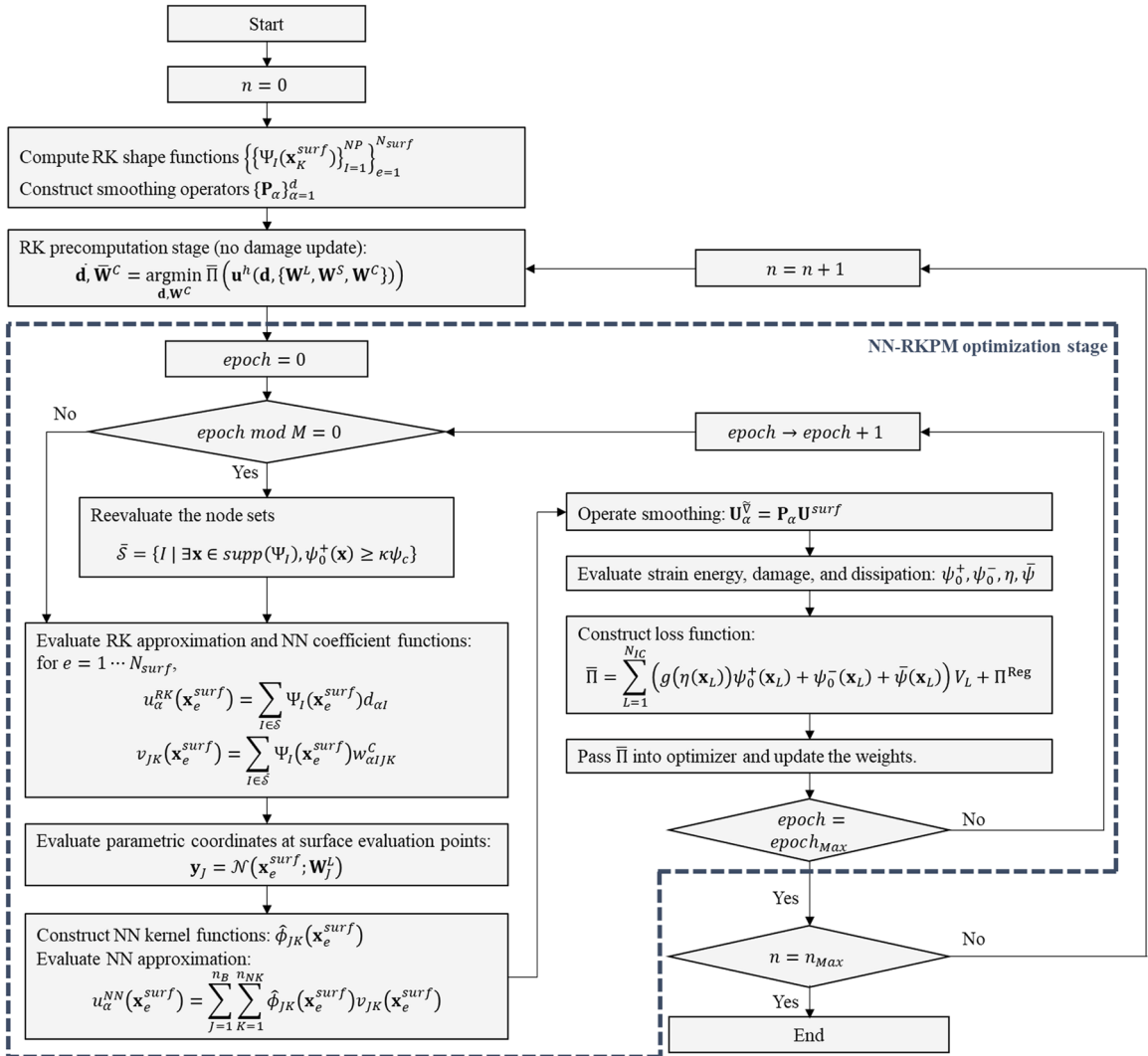
where $\gamma = \max(p+1, \widetilde{r})$. For smooth u in $\Omega \setminus \widehat{\Omega}$, $\widehat{\gamma} = \gamma - 0.5$ holds, which means that $\widehat{\gamma}$ dominates the first term on the right-hand side of Eq. (64), leading to

$$\|u^h - u\|_{0,\Omega} \leq (C + \widehat{\widehat{C}}) a^{\widehat{\gamma}} k |u|_{p+1,\Omega \setminus \widehat{\Omega}} + \frac{[[[u^\Gamma]]]]}{\ell} \|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}, \quad (65)$$

In the last term of Eq. (65), $\|y(x) - y^\mu(x)\|_{0,\widehat{\Omega}}$ denotes the parametrization error. Eq. (65) implies that, when the parametrization error is relatively large, the solution convergence will be governed by the convergence of the parametrization. Conversely, for $\|y(x) - y^\mu(x)\|_{0,\Gamma} \rightarrow 0$, the convergence will be governed by the background RK discretization with a rate of $\widehat{\gamma}$, e.g., 1.5 when a linear RK basis is used. The error bound of $\|y(x) - y^\mu(x)\|_{0,\Gamma}$ follows the universal approximation theorem [1,37] when a neural network is used for parametrization. For example, for a neural network with a single hidden layer, the error bound is estimated as follows [1]: with a generic constant $C_y < \infty$,

$$\|y(x) - y^\mu(x)\|_{0,\Gamma} \leq C_y n_{NR}^{-1/2}, \quad (66)$$

which leads to the following error estimation of NN-RK approximation



* For brevity, the flowchart does not include the body force, traction, and regularization contribution for the loss function computation.

Fig. 8. Flowchart of the solution procedure

$$\|u^h - u\|_{0,\Omega} \leq (C + \hat{C}) a^{\frac{1}{2}} k |u|_{p+1,\Omega \setminus \hat{\Omega}} + C_y \frac{[[[u^\Gamma]]]]}{\ell} n_{NR}^{-1/2}. \quad (67)$$

3.6. Regularization

To avoid the potential loss of ellipticity of the problem and the resulting discretization sensitivity in the numerical solution of the local problem defined in Section 2, a regularization treatment is needed. A straightforward remedy is to impose a proper constraint such that the physical bandwidth of the damage does not become narrower than a certain limit. To analyze a localization width possessed by the NN-RK approximation, we start with a Taylor expansion of the parametric coordinate as follows:

$$y(\mathbf{x}) \approx \bar{y} + (\mathbf{x} - \bar{\mathbf{x}}) \cdot \nabla^{\mathbf{x}} y(\bar{\mathbf{x}}), \quad (68)$$

where $\bar{y} = y(\bar{\mathbf{x}})$, and $\bar{\mathbf{y}}$ is defined in Section 3.4, for which the superscripts and subscripts are omitted for brevity. With Eq. (68), z defined in Eq. (50) is written as

$$z(y(\mathbf{x}); \{\bar{y}, c\}) = \frac{(y(\mathbf{x}) - \bar{y})}{c} \approx \frac{(\mathbf{x} - \bar{\mathbf{x}}) \cdot \nabla^{\mathbf{x}} y(\bar{\mathbf{x}})}{c} \equiv \frac{\bar{\xi}(\mathbf{x}; \bar{\mathbf{x}})}{c}, \quad (69)$$

with $\bar{\xi}(\mathbf{x}; \bar{\mathbf{x}}) \equiv (\mathbf{x} - \bar{\mathbf{x}}) \cdot \nabla^{\mathbf{x}} y(\bar{\mathbf{x}})$. When $\|\nabla^{\mathbf{x}} y(\bar{\mathbf{x}})\| = 1$, $\bar{\xi}(\mathbf{x}; \bar{\mathbf{x}})$ in Eq. (69) is a projection of the physical coordinate onto the direction

normal to the localization. Therefore, by satisfying conditions

$$\begin{aligned} \|\nabla^{\mathbf{x}} y(\bar{\mathbf{x}})\| &\leq 1, \\ c &\geq \ell, \end{aligned} \quad (70)$$

the transition width of $\bar{\phi}$ in Eq. (49) has a lower bound of ℓ , and thus the localization width in the NN-RK approximation has the same lower bound. In this work, a constraint $\|\nabla^{\mathbf{x}} y\| \leq 1$ is imposed in the loss function (Eq. (1)), and the lower bound of the sharpness control parameter c in Eq. (50) is set to an NN length scale parameter ℓ . The modified loss function with regularization reads:

$$\min_{\mathbf{u}} \bar{\Pi}(\mathbf{u}, \mathbf{y}) = \Pi(\mathbf{u}) + \Pi^{\text{Reg}}(\mathbf{y}), \quad \Pi^{\text{Reg}}(\mathbf{y}) = \frac{\kappa \mu}{2} \sum_{a,j} \int_{\Omega} \langle \|\nabla^{\mathbf{x}} y_{ja}(\mathbf{x})\| - 1 \rangle_+^2 d\Omega, \quad (71)$$

where Π is the potential function defined in Eq. (1), and κ is the normalized penalty parameter. In this work, $\kappa = 10^4$ is used. Note that this approach is different from the $\hat{H}$ -regularization introduced by Baek et al. (2022) [33] in which the parametric coordinates are directly scaled by $\hat{H}$ as follows:

$$z = \frac{(y - \bar{y})\hat{H}}{c}, \quad \text{where } \hat{H} \equiv 1 / \max(\|\nabla^{\mathbf{x}} y\|, 1). \quad (72)$$

An advantage of the regularization designed in this work over the $\hat{H}$ -regularization is that the necessity to compute the second order gradient of y for the evaluation of the strain energy in the loss function is avoided.

4. Numerical implementation

The minimization problem is rewritten as follows:

$$\min_{\mathbf{d}, \mathbf{W}} [\Pi(\mathbf{u}^h(\mathbf{d}, \mathbf{W})) + \Pi^{\text{Reg}}(\mathbf{y}(\mathbf{x}; \mathbf{W}^L))], \quad (73)$$

where $\mathbf{u}^h(\mathbf{d}, \mathbf{W}) = \mathbf{u}^{\text{RK}}(\mathbf{d}) + \mathbf{u}^{\text{NN}}(\mathbf{W})$ is the NN-RK approximation with the RK coefficient set, $\mathbf{d}$, and the neural network weight set, $\mathbf{W} = \{\mathbf{W}^L, \mathbf{W}^S, \mathbf{W}^C\}$ with $\mathbf{W}^L = \{\mathbf{W}_J^L\}_{J=1}^{n_B}$, $\mathbf{W}^S = \{\mathbf{W}_J^S\}_{J=1}^{n_B}$, and $\mathbf{W}^C = \{\mathbf{W}_J^C\}_{J=1}^{n_B}$. In Eq. (73), ψ and F denote the energy density and the external work defined in Eq. (1), respectively.

Fig. 8 shows the flowchart of the solution procedure. In the flowchart n and n_{Max} denotes the loading step and the maximum loading step, respectively. At loading step $n + 1$, the solution procedure mainly consists of two parts: RK precomputation stage and NN-RKPM optimization stage.

A RK precomputation stage

To obtain the initial guesses $\bar{\mathbf{d}}^{(n+1)}$ and $\bar{\mathbf{W}}^{C(n+1)}$ to be used in the NN-RKPM optimization stage, the minimization problem (Eq. (73)) is first solved only for $\mathbf{d}^{(n+1)}$ and $\mathbf{W}^{C(n+1)}$:

$$\begin{aligned} \bar{\mathbf{d}}^{(n+1)}, \mathbf{W}^{C(n+1)} &= \underset{\mathbf{d}, \mathbf{W}^C}{\operatorname{argmin}} \left[\Pi(\mathbf{u}^h(\mathbf{d}, \{\mathbf{W}^{L(n)}, \mathbf{W}^{S(n)}, \mathbf{W}^C\})) + \Pi^{\text{Reg}}(\mathbf{y}(\mathbf{x}; \mathbf{W}^{L(n)})) \right] \\ &\text{subjected to } \mathbf{u}(\mathbf{x}) = \mathbf{g}^{(n+1)} \text{ on } \partial\Omega_g. \end{aligned} \quad (74)$$

In this stage, the weight sets $\{\mathbf{W}^L, \mathbf{W}^S\}$ and the damage η from the previous loading step are used. Also, the damage is not updated. This is equivalent to the standard Galerkin-based RKPM problem and can be solved by a standard matrix solver.

B NN-RKPM optimization stage

In the second stage, the minimization problem (Eq. (73)) is solved for the entire unknown parameters $\mathbf{d}$ and $\mathbf{W}$.

$$\begin{aligned} \bar{\mathbf{d}}^{(n+1)}, \mathbf{W}^{(n+1)} &= \underset{\mathbf{d}, \mathbf{W}}{\operatorname{argmin}} [\Pi(\mathbf{u}^h(\mathbf{d}, \mathbf{W})) + \Pi^{\text{Reg}}(\mathbf{y}(\mathbf{x}; \mathbf{W}^L))] \\ &\text{subjected to } \mathbf{u}(\mathbf{x}) = \mathbf{g}^{(n+1)} \text{ on } \partial\Omega_g. \end{aligned} \quad (75)$$

In this stage, the damage is updated as well. The minimization problem can be solved iteratively by a suitable optimizer. In this work, *Adam* [42], a first-order optimizer with adaptive learning rate, is used for the first several epochs. Then, the optimizer is switched to limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) [43], a second-order optimizer, for the remaining optimization.

For domain integration involved in Eq. (73), SCNI introduced in Section 2.2.2 is used with refined smoothing cells near localization. As discussed in Section 2.2.2, the advantage of using SCNI for the proposed method is twofold: 1) it eliminates the requirement of computing the computationally expensive direct derivative of $\mathbf{u}^{\text{NN}}$ with the automatic differentiation to evaluate strain and stress, and 2) it suppresses stress oscillations.

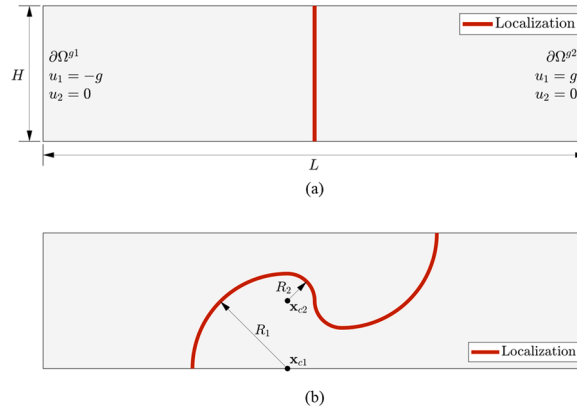


Fig. 9. Geometry and boundary conditions for problem of elasticity with pre-existing damaged zone: (a) Case I and (b) Case II

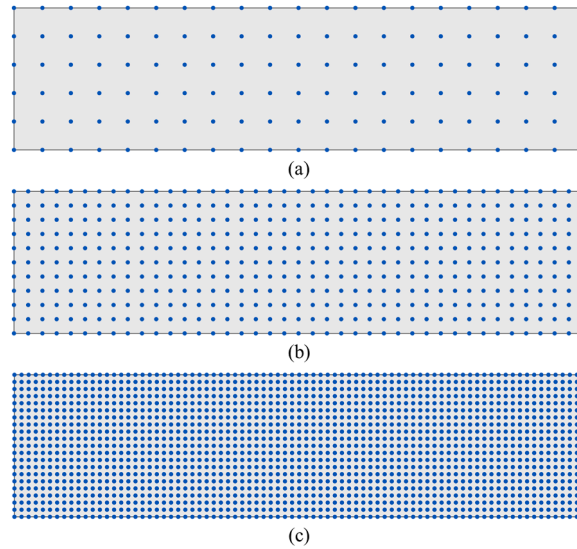


Fig. 10. Background RK discretizations used for the elasticity with pre-damaged material: (a) 21×6 RK nodes with $h = H/5$, (b) 41×11 nodes with $h = H/10$, and (c) 81×21 RK nodes with $h = H/20$

For computationally efficient implementation of the strain smoothing operation in SCNI, precomputed sparse smoothing matrices $\mathbf{P}_\alpha$ with $\alpha = 1 \dots d$ can be considered to perform the following global smoothing:

$$\mathbf{U}_\alpha^\nabla = \mathbf{P}_\alpha \mathbf{U}^{surf}, \quad (76)$$

by which the strain smoothing in all the smoothing cells as discussed in section 2 are conducted simultaneously. In Eq. (76), $\mathbf{U}_\alpha^\nabla = [\widetilde{u}_{,\alpha}^h(\mathbf{x}_1), \dots, \widetilde{u}_{,\alpha}^h(\mathbf{x}_L), \dots, \widetilde{u}_{,\alpha}^h(\mathbf{x}_{N_{IC}})]^T$ is a column vector containing the smoothed gradients of u^h with respect to x_α for all the smoothing cells in the domain, i.e., $L = 1 \dots N_{IC}$. $\mathbf{U}^{surf} = [u^h(\mathbf{x}_1^{surf}), \dots, u^h(\mathbf{x}_e^{surf}), \dots, u^h(\mathbf{x}_{N_{surf}}^{surf})]^T$ is a column vector containing u^h evaluated at a smoothing cell surface evaluation point $\mathbf{x}_e^{surf}$ for $e = 1 \dots N_{surf}$, where N_{surf} denotes the total number of smoothing cell surface evaluation points in the domain. The (L, e) component of the smoothing operator $\mathbf{P}_\alpha$ is

$$P_{\alpha Le} = \begin{cases} \frac{1}{V_L} A_e n_\alpha^K, & \text{if } \Gamma_e \subset \Omega_L, \\ 0, & \text{otherwise} \end{cases}, \quad (77)$$

where Γ_e , Ω_L , n_α^K , and A_e denote e -th smoothing cell surface segment, L -th smoothing cell domain, α -th component of the surface normal, and the area of e -th smoothing cell surface segment, respectively. The same procedure can be used to compute $\widetilde{\nabla} y_i$ for Eq. (71).

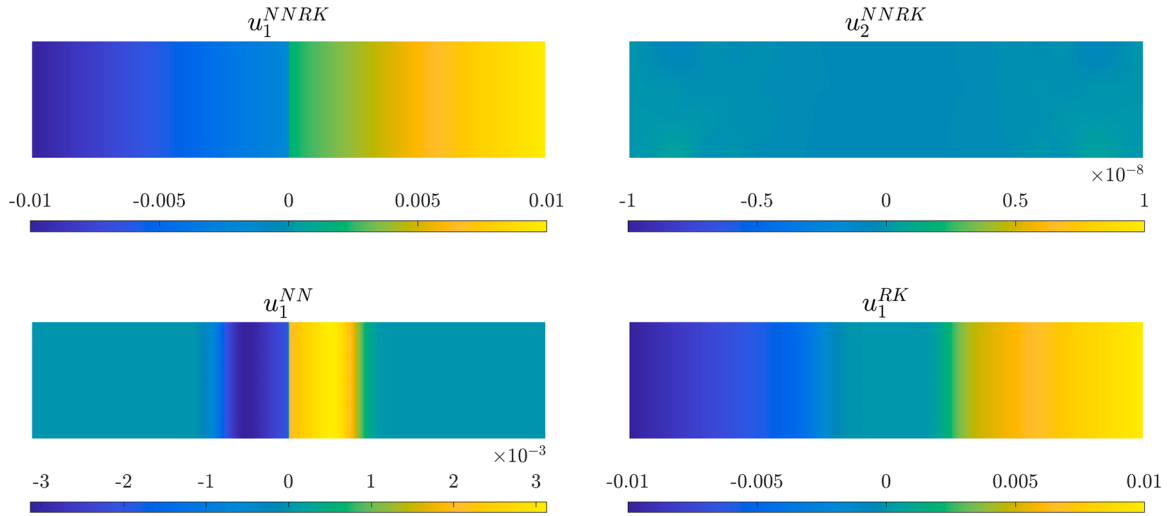
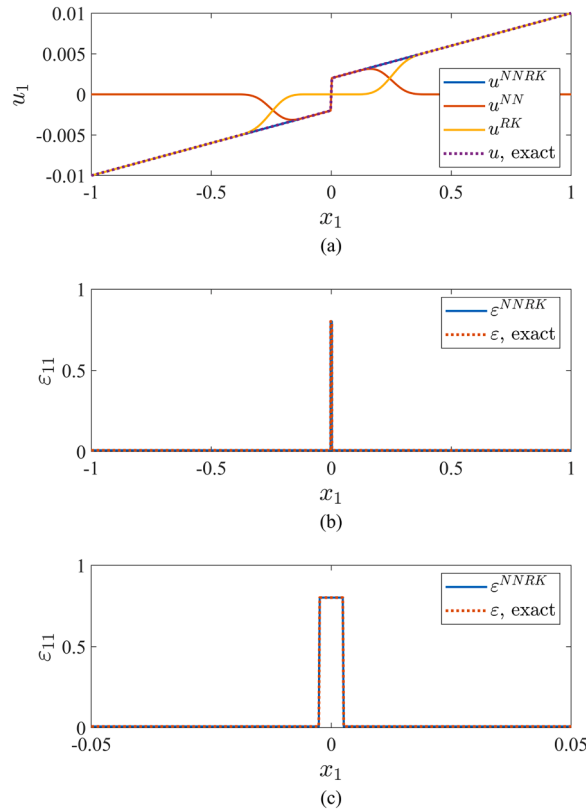


Fig. 11. Predicted displacement (Case I)

Fig. 12. Numerical solution along $y = 0$ (Case I): (a) u_1 , (b) ϵ_{11} , and (c) ϵ_{11} (zoom-in)

5. Numerical examples

Several numerical examples are presented to demonstrate the proposed method's accuracy, regularization ability, and capability to capture complicated localization patterns. Unless otherwise specified, for the RK approximation, the linear basis with cubic B-spline kernel function of normalized support size 2.0 is used, and, for the NN approximation, a single 4-kernel NN block is used along with a densely connected neural network with the hyperbolic tangent activation function for the parametrization sub-block. For the domain integration, SCNI is used with refined smoothing cells in the zone along the expected damage path.

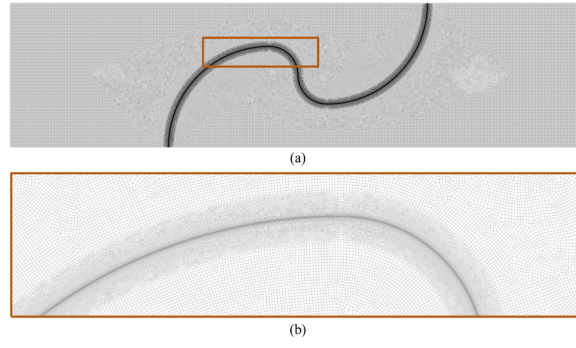


Fig. 13. Body-fitted Q8-FEM discretization used to compute reference solution of Case II: (a) entire domain discretized by 1,070,298 finite elements with $h = w/12$ near the localization and (b) a zoom-in plot

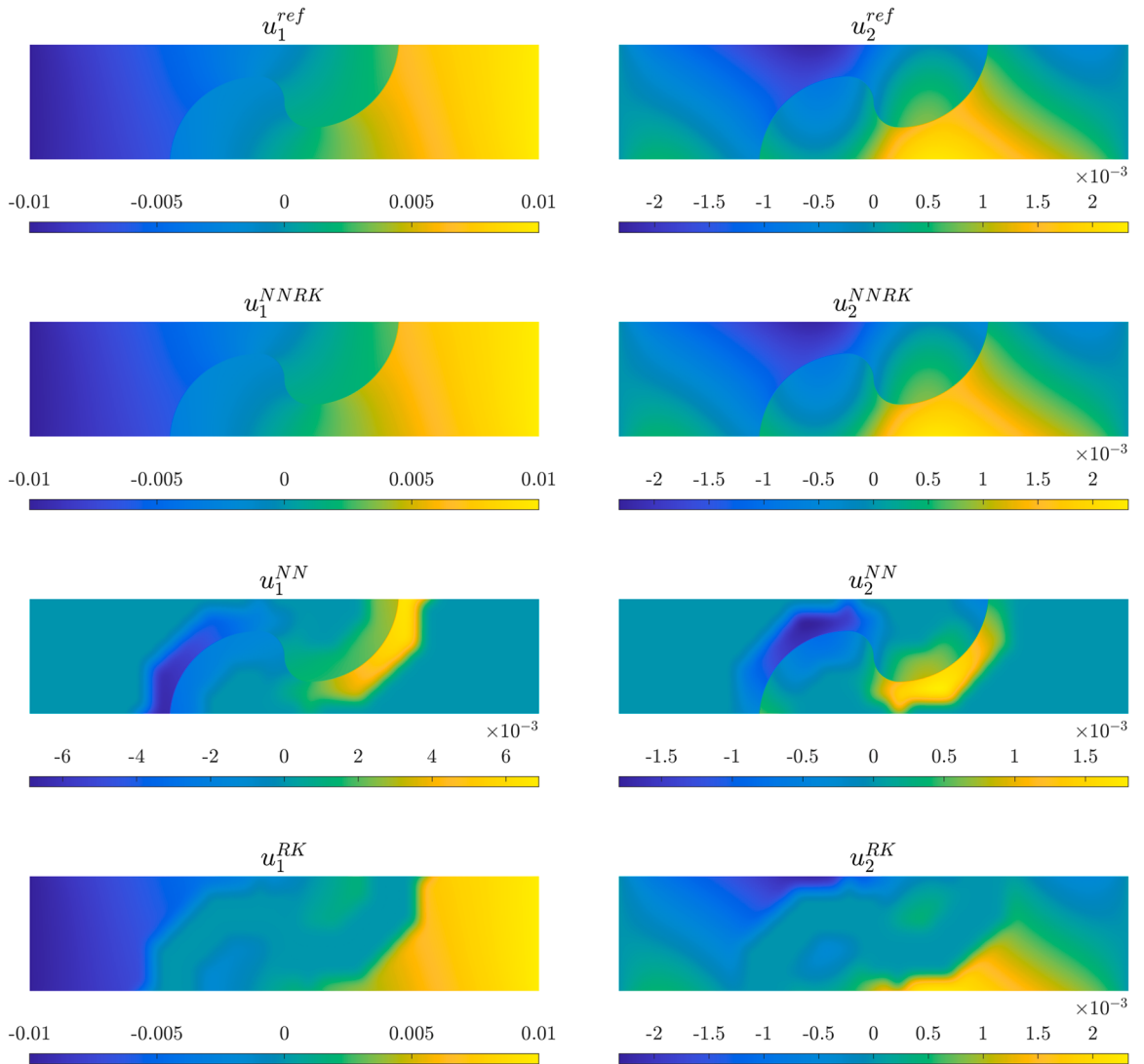


Fig. 14. Displacement field (Case II): reference solution and NNRK solution (41×11)

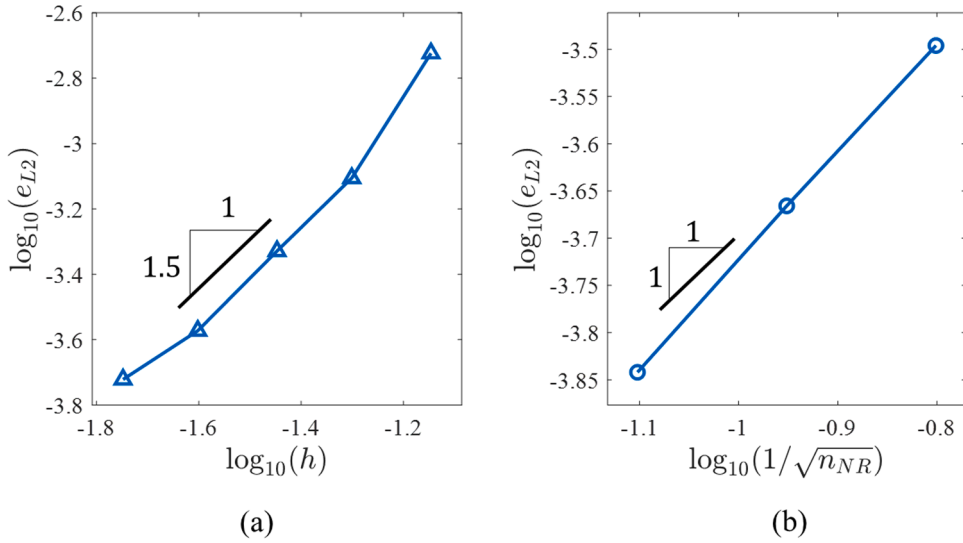


Fig. 15. L_2 convergence rates: (a) for varying background RK nodal spacing with a fixed width of hidden layer ($n_{NR} = 160$) and (b) for varying n_{NR} with a fixed RK discretization ($h = H/40$). The values enclosed by the parentheses in the legend denote the average convergence rates.

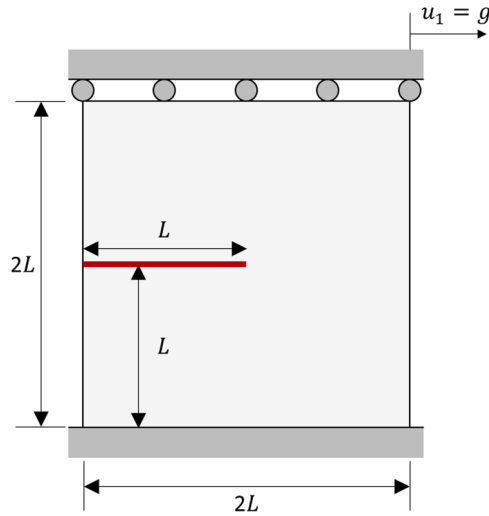


Fig. 16. A pre-notched specimen for simple shear problem

5.1. Elasticity with pre-existing damaged zone

Consider a domain $[-L/2, L/2] \times [-H/2, H/2]$ with a degraded zone with width w . We consider two different cases of pre-existing damaged zone geometry, as show in Fig. 9 (a) and (b). For both cases, $L = 2$ mm and $H = 0.5$ mm are used. For Case I, the degraded zone is vertically aligned at the center of the domain. For Case II, the anti-symmetric degraded zone is centered at the origin with $\mathbf{x}_{c1} = (-0.1, -0.5)$, $R_1 = 0.35$, $\mathbf{x}_{c2} = (-0.1, 0)$, and $R_2 = 0.1$ in unit of mm. For both cases, Dirichlet boundary conditions are applied to the left and right surfaces with $g = 1 \times 10^{-2}$ mm, and zero traction boundary conditions are applied to the top and bottom surfaces. For Case I, $w = H/100$, $E = 210$ GPa, and $\nu = 0$ are used, and for Case II, $w = H/1000$, $E = 210$ GPa, and $\nu = 0.3$ are used. The Young's modulus within the degraded zones is kE with $k = 10^{-2}$ for Case I and $k = 10^{-3}$ for Case II.

For Case I, the exact solution is as follows:

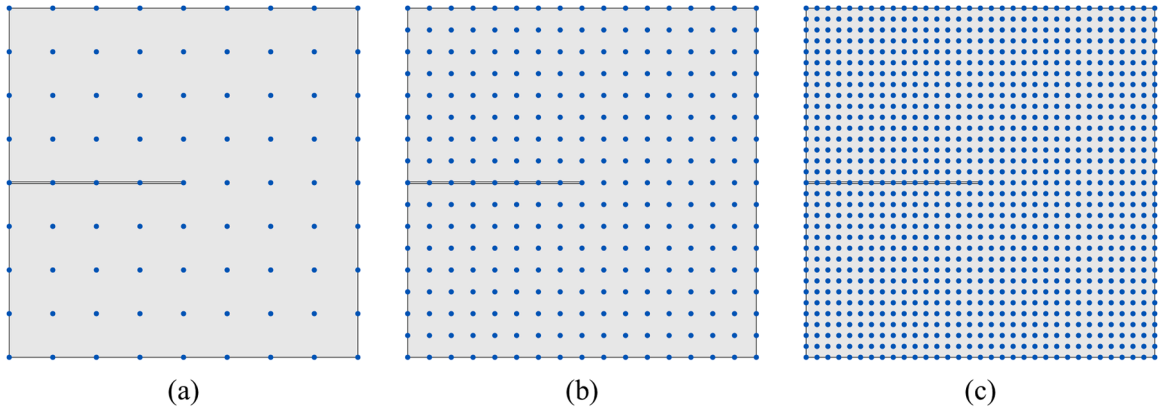


Fig. 17. Background RK discretizations employed for simple shear problem. (a) M1: $h = L/4$, (b) M2: $h = L/8$, (c) M3: $h = L/16$

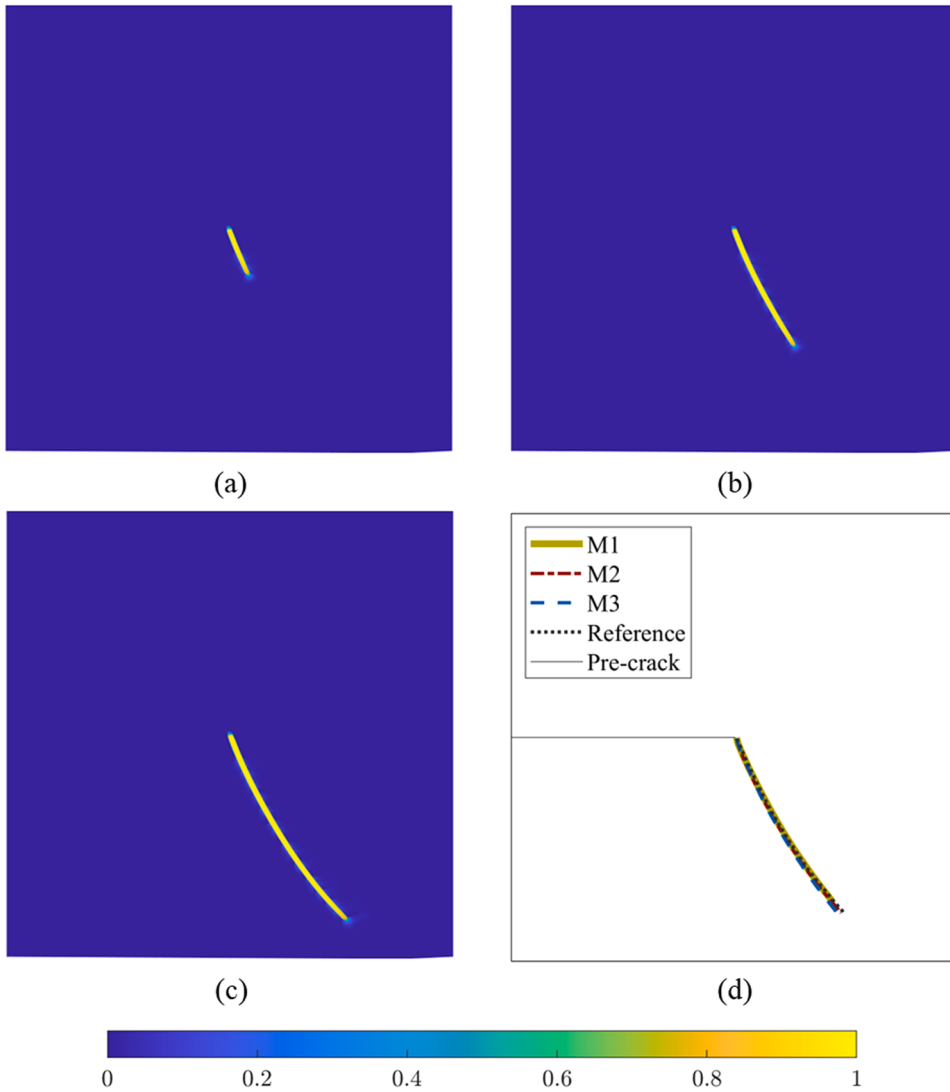


Fig. 18. Damage evolution in simple shear problem (M2) for (a) $g = 9 \times 10^{-3}$, (b) $g = 10 \times 10^{-3}$, (c) $g = 11.5 \times 10^{-3}$, and (d) comparison of the predicted damage paths and the reference solution

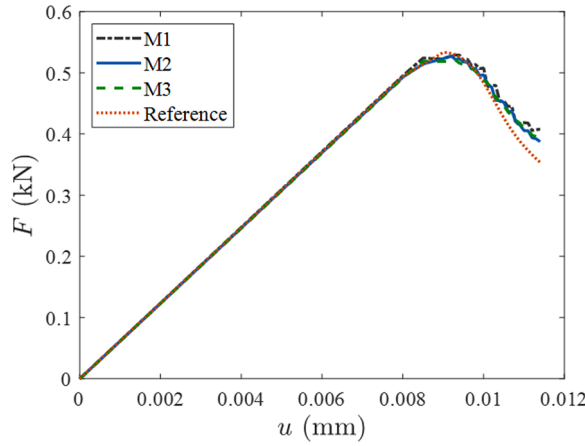


Fig. 19. Load-displacement curve in simple shear problem

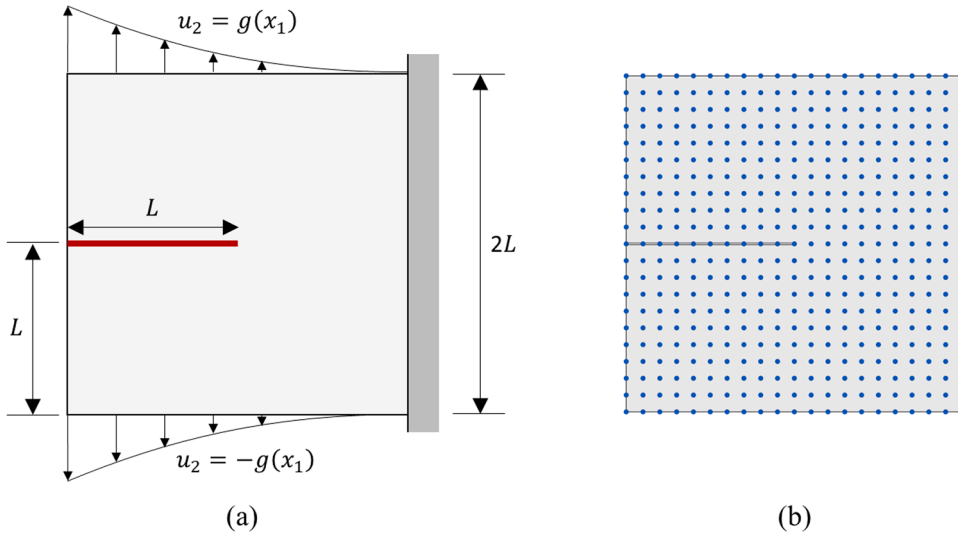


Fig. 20. A pre-notched specimen for static branching problem: (a) geometry and boundary conditions and (b) background RK discretization

$$u_1(\mathbf{x}) = \begin{cases} b(x_1 + L) - g, & x_1 \leq -w/2 \\ (b/k)x_1, & -w/2 < x_1 \leq w/2 \\ b(x_1 - L) + g, & x_1 > w/2 \end{cases} \quad (78)$$

$$u_2(\mathbf{x}) = 0$$

where $b = 2g/((1/k - 1)w + 2L)$. For the numerical solution, the domain is uniformly discretized by 21×6 RK nodes (see Fig. 10 (a)), and a single 10-neuron hidden layer is used for the parametrization sub-block. Fig. 11 shows the displacement predicted by the proposed method. The numerical solution captures the sharp transition in the horizontal displacement very well along with the zero vertical displacement due to zero Poisson's ratio. As shown in the figures in the 2nd row in Fig. 11, the NN approximation appears near the localization capturing the sharp transition of u_1 , and the RK approximation captures the solution in the other area, with smooth transition between two approximations. Fig. 12 shows the horizontal displacement and normal strain along $y = 0$ in which the numerical solution is shown to be highly accurate compared to the exact solution. The computed L_2 norm and H^1 semi-norm of the solution error are 2.921×10^{-4} and 2.437×10^{-6} , respectively.

For Case II, the background RK discretizations employed in this section are plotted in Fig. 10 (a-c), and a 1,070,298-node, body-fitted Q8-FEM solution with a minimum nodal spacing of $H/2000$ near the localization (see Fig. 13 for discretization) is used as a reference solution. Fig. 14 shows the numerical solution for Case II, using 41×11 uniformly distributed background RK nodes

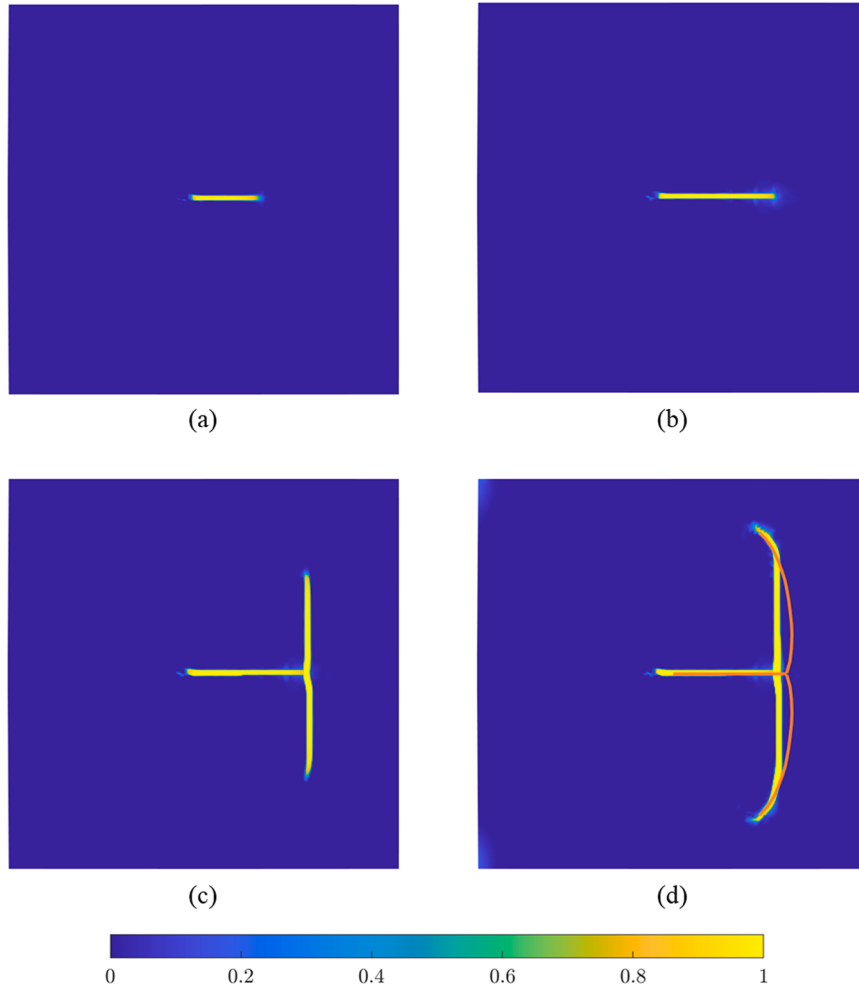


Fig. 21. Predicted damage propagation and branching: g_D of (a) 0.02 mm, (b) 0.036 mm, (c) 0.04 mm, and (d) 0.08 mm with a reference solution [47] superimposed in orange color

(Fig. 10 (b)) and a single 40-neuron hidden layer. Although the background RK discretizations shown in Fig. 8 are relatively coarse compared to the width of degraded zone, the displacements predicted by the proposed method match the reference solution very well. The convergence curve for varying background RK nodal spacing (h) and the convergence curve for the varying number of neurons (n_{NR}) are plotted in Fig. 15 (a) and (b), respectively. For the convergence study shown in Fig. 15 (a), a fixed value of $n_{NR} = 160$ is used, and for the study shown in Fig. 15 (b), a fixed value of $h = H/40$ is used. Both results show convergence behaviors consistent with the error analysis result presented in Section 3.2.

5.2. Pre-notched specimen subjected to simple shear

A benchmark problem of pre-notched specimen under simple shear is considered. As shown in Fig. 16, a specimen with domain $\Omega = [-L, L] \times [-L, L]$ with a pre-existing crack of length L is subjected to Dirichlet boundary conditions on the top and bottom surfaces. Specimen dimension $L = 0.5$ mm is used in this problem. The horizontal boundary value g applied to the top surface is increased up to 15×10^{-3} mm with an increment of 1×10^{-4} mm. The material properties of $E = 210$ GPa, $\nu = 0.3$, $\mathcal{E}_c = 2.7$ N/mm are used. As shown in Fig. 17, three levels of RK discretizations are used to study the regularization capability of the proposed method. For NN approximation, the parametrization subblock consists of a neural network with two 40-neuron hidden layers along with the hyperbolic tangent activation function, which involves 1,842 unknown weights and biases for all three models. For verification, a reference solution based on the reproducing kernel strain regularization (RKSR) [44] method is employed using 160,801 uniformly distributed RK nodes with nodal spacing of $h = L/200$.

Fig. 18 (a-c) shows the damage propagation predicted by the proposed method. The damage is initiated with an orientation of approximately 65° and gradually changes the direction to the lower right corner during the propagation. The predicted damage paths plotted in Fig. 18 (d) are not sensitive to the background RK discretization and agree very well with the reference solution. In addition,

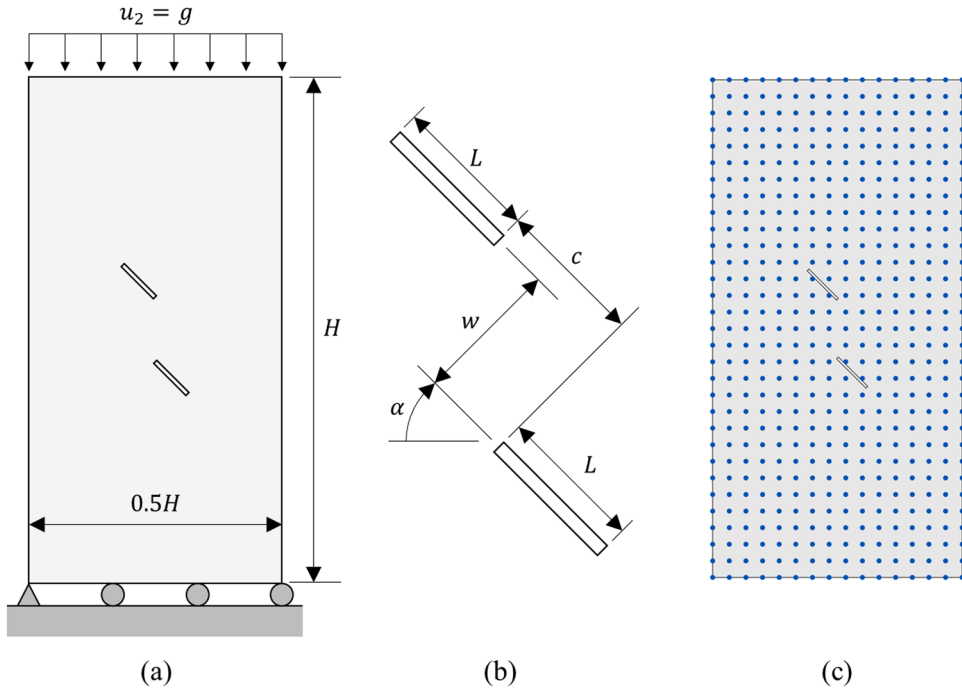


Fig. 22. A rock specimen with double preexisting cracks: (a) geometry and boundary conditions, (b) details of preexisting notch, and (c) background RK discretization

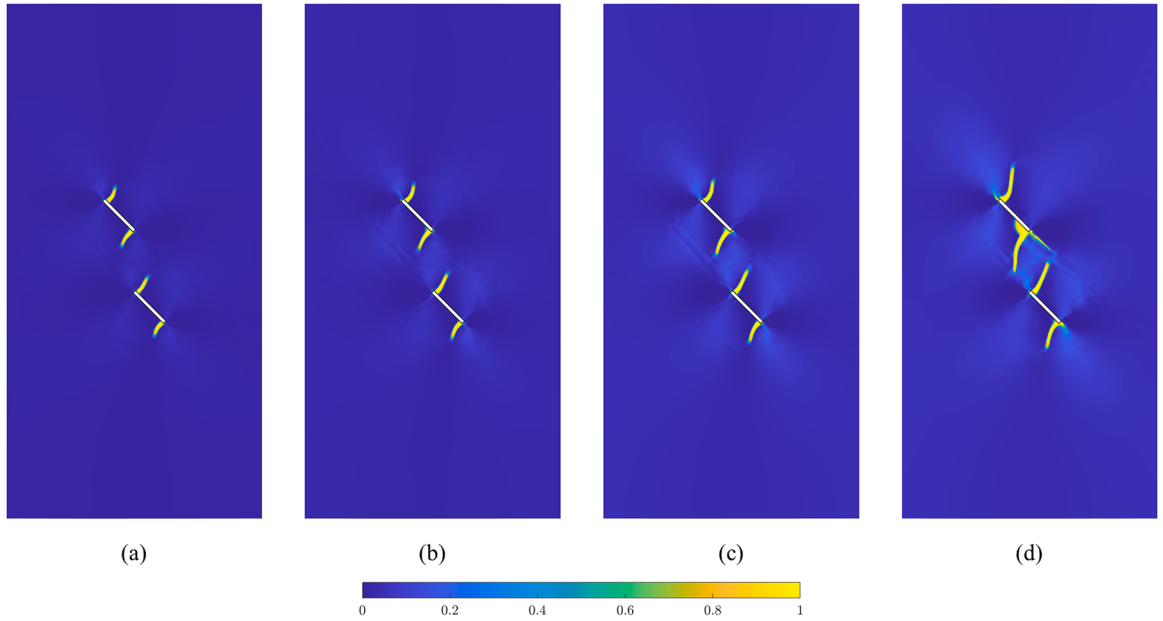


Fig. 23. Progressive damage in rock-like specimen induced by uniaxial compression: $g =$ (a) -0.4 mm, (b) -0.5 mm, (c) -0.6 mm, and (d) -0.65 mm

as shown in Fig. 19, the load-displacement curves also demonstrate the good regularization capability of the proposed method and present reasonable agreement with the RKSR reference solution [44].

The computations were performed using Python with TensorFlow on a single NVIDIA A100 GPU. The runtimes taken by NN-RKPM with M1, M2, and M3 are 144, 148, and 155 minutes, respectively. The small discrepancy among the models is expected, since the optimization of the modified deep neural network is the primary contributor to the computational expense, particularly with such coarse background discretizations. The RKSR reference simulation consumed 597 minutes, which is within the same order of

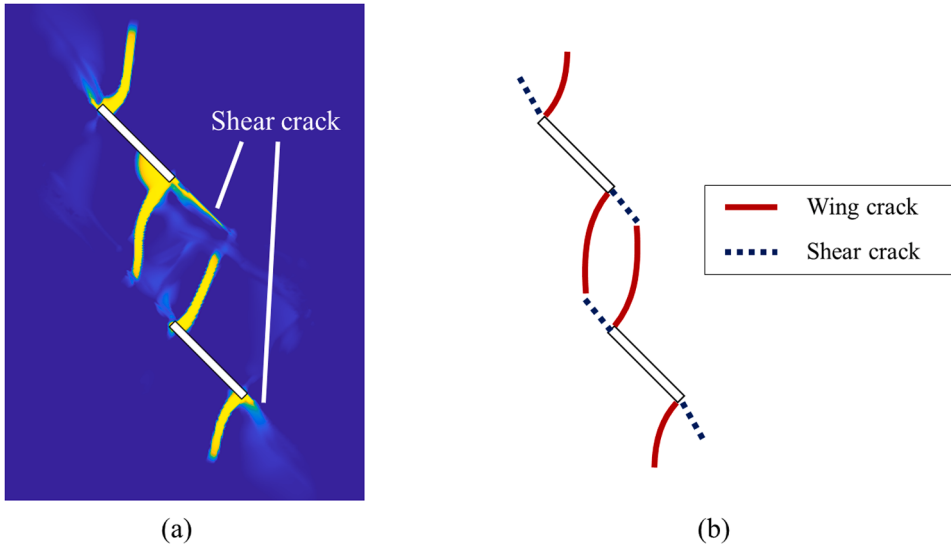


Fig. 24. Comparison of (a) numerical results and (b) experimental observation [48]

magnitude as the reported runtime of 957 minutes for the adaptively refined FEM phase field simulation reported in [45]. While it is hard to make direct comparisons due to variations in hardware and software platforms used for different simulations, these observations highlight the computational efficiency of the proposed method that leverages the flexibility of neural network function spaces to significantly reduce the background RKPM degrees of freedom.

5.3. Quasi-static crack branching problem

In this section, the proposed method's ability to capture branching is demonstrated through a numerical example inspired by the problem proposed by Muixi et al [46,47]. Consider a square domain $\Omega = [-L, L] \times [-L, L]$ with a pre-existing notch with a length of L , as shown in Fig. 20. The specimen is subjected to vertical displacement boundary conditions $g(x) = g_D(1 - x^2)/8$ on the top and bottom surfaces while the right surface is fixed in both directions. Herein, $L = 1$ mm is considered, and g_D is applied up to 0.08 mm with $\Delta g_D = 4 \times 10^{-3}$ mm. The material properties $E = 20$ GPa, $\nu = 0.3$, and $\mathcal{G}_c = 8.9 \times 10^{-5}$ kN/mm are used.

In Fig. 21, a progressive damage field is plotted in which the fracture initially propagates horizontally and branched near the fixed boundary as the accumulated strain energy associated with the vertical strain decreases due to the displacement constraint, which prevents further propagation of the fracture toward the fixed boundary. The branching is predicted to occur abruptly, then the propagation rate slows down. At the late stage of simulation, two branches switch the direction to the left. The overall trend of the damage propagation agrees with the reference PF-XFEM solution [47] superimposed in Fig. 21 (d).

5.4. Mixed-mode fracture of a doubly notched rock-like specimen subjected to uniaxial compression

A uniaxial compression of a rock-like specimen with double pre-existing cracks [48] is simulated. As shown in Fig. 22, a rectangular specimen with $H = 152.4$ mm consists of two 1-mm thick pre-existing cracks with $L = c = w = 12.7$ mm and $\alpha = 45^\circ$. The Dirichlet boundary condition on the top surface is prescribed up to $g = -0.65$ mm with the increment $\Delta g = -1 \times 10^{-2}$ mm. Material parameters are Young's modulus of $E = 5.96$ GPa, Poisson's ratio of $\nu = 0.24$, the mode-I fracture energy of $\mathcal{G}_I = 5$ N/m, and the mode-II fracture energy of $\mathcal{G}_{II} = 20\mathcal{G}_I$. The domain is uniformly discretized by 16×31 RK particles. For NN approximation, the parametrization subblock consists of a neural network with two 40-neuron hidden layers along with the hyperbolic tangent activation function, which involves 1,842 unknown weights and biases. The NN length scale of 1 mm is employed.

Fig. 23 shows the predicted damage propagation in the rock specimen. At the initial stage, four wing cracks are initiated from the four corners of the pre-existing notches and propagates with curved paths. Then, secondary shear cracks start to develop approximately at $g = -0.65$ mm. As shown in Fig. 24, the predicted fracture pattern agrees reasonably well with the experimental observation [48].

6. Conclusion

An improved neural network-enhanced reproducing kernel particle method has been proposed for modeling brittle fracture. Derived through an NN-based correction of standard RK shape functions, the proposed method enriches a background reproducing kernel (RK) approximation with a coarse and uniform discretization by a neural network (NN) approximation equipped with a Partition of Unity property. The NN approximation is constructed by a deep neural network designed to capture localization, and the NN based enrichment functions are then patched together with RK approximation functions using RK as a Partition of Unity patching

function. In the NN approximation, the deep neural network locates and inserts regularized discontinuities in the approximation function automatically, and the resulting NN enriched RK coefficient function provides varying magnitude of the discontinuities along the localization path.

To automatically capture the location, orientation, and solution transition across and along the localization, the optimum values of the control parameters contained in the deep neural network as well as the RK coefficients are obtained via minimization of the energy-based loss function. A regularization by introducing a constraint on the spatial gradient of the parametric coordinates to the loss function is employed to ensure a discretization-independent solution. Error analysis of the proposed NN-RK approximation is performed, and its verification with the numerical results show good agreement on the convergence rates. The numerical examples demonstrate the effectiveness of the proposed method in modeling damage evolution and branching with a fixed background discretization without conventional adaptive refinement, achieving an enhanced computational efficiency.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The support from the National Science Foundation under award #1826221 to University of California, San Diego, is greatly acknowledged.

References

- [1] A.R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inf. Theory*. 39 (1993) 930–945.
- [2] O. Calin, *Deep learning architectures*, Springer, 2020.
- [3] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, Deep learning predicts path-dependent plasticity, *Proc. Natl. Acad. Sci.* 116 (2019) 26414–26420, <https://doi.org/10.1073/pnas.1911815116>.
- [4] Q. He, J.-S. Chen, A physics-constrained data-driven approach based on locally convex reconstruction for noisy database, *Comput. Methods Appl. Mech. Eng.* 363 (2020), 112791, <https://doi.org/10.1016/j.cma.2019.112791>.
- [5] L. Wu, V.D. Nguyen, N.G. Kilinger, L. Noels, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, *Comput. Methods Appl. Mech. Eng.* 369 (2020), 113234, <https://doi.org/10.1016/j.cma.2020.113234>.
- [6] X. He, Q. He, J.-S. Chen, Deep autoencoders for physics-constrained data-driven nonlinear materials modeling, *Comput. Methods Appl. Mech. Eng.* 385 (2021), 114034, <https://doi.org/10.1016/j.cma.2021.114034>.
- [7] D.W. Abueidda, S. Koric, N.A. Sobh, H. Sehitoglu, Deep learning for plasticity and thermo-viscoplasticity, *Int. J. Plast.* 136 (2021), 102852, <https://doi.org/10.1016/j.ijplas.2020.102852>.
- [8] X. He, J.-S. Chen, Thermodynamically consistent machine-learned internal state variable approach for data-driven modeling of path-dependent materials, *Comput. Methods Appl. Mech. Eng.* 402 (2022), 115348, <https://doi.org/10.1016/j.cma.2022.115348>.
- [9] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, *J. Comput. Phys.* 404 (2020), 108973, <https://doi.org/10.1016/j.jcp.2019.108973>.
- [10] Y. Kim, Y. Choi, D. Widemann, T. Zohdi, A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, *J. Comput. Phys.* 451 (2022), 110841, <https://doi.org/10.1016/j.jcp.2021.110841>.
- [11] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707, <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [12] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, *Comput. Methods Appl. Mech. Eng.* 379 (2021), 113741, <https://doi.org/10.1016/j.cma.2021.113741>.
- [13] K. Taneja, X. He, Q. He, X. Zhao, Y.-A. Lin, K.J. Loh, J.-S. Chen, A Feature-Encoded Physics-Informed Parameter Identification Neural Network for Musculoskeletal Systems, *J. Biomech. Eng.* 144 (2022), <https://doi.org/10.1115/1.4055238>.
- [14] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, *Int. J. Numer. Methods Eng.* 48 (2000) 1741–1760, [https://doi.org/10.1002/1097-0207\(20000830\)48:12<1741::AID-NME956>3.0.CO;2-L](https://doi.org/10.1002/1097-0207(20000830)48:12<1741::AID-NME956>3.0.CO;2-L).
- [15] N. Sukumar, N. Moës, B. Moran, T. Belytschko, Extended finite element method for three-dimensional crack modelling, *Int. J. Numer. Methods Eng.* 48 (2000) 1549–1570, [https://doi.org/10.1002/1097-0207\(20000830\)48:11<1549::AID-NME955>3.0.CO;2-A](https://doi.org/10.1002/1097-0207(20000830)48:11<1549::AID-NME955>3.0.CO;2-A).
- [16] C.A. Duarte, O.N. Hamzeh, T.J. Liszka, W.W. Tworzydło, A generalized finite element method for the simulation of three-dimensional dynamic crack propagation, *Comput. Methods Appl. Mech. Eng.* 190 (2001) 2227–2262, [https://doi.org/10.1016/S0045-7825\(00\)00233-4](https://doi.org/10.1016/S0045-7825(00)00233-4).
- [17] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, *Int. J. Numer. Methods Eng.* 45 (1999) 601–620, [https://doi.org/10.1002/\(SICI\)1097-0207\(19990620\)45:5<601::AID-NME598>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S).
- [18] J. Dolbow, N. Moës, T. Belytschko, Discontinuous enrichment in finite elements with a partition of unity method, *Finite Elem. Anal. Des.* 36 (2000) 235–260, [https://doi.org/10.1016/S0168-874X\(00\)00035-4](https://doi.org/10.1016/S0168-874X(00)00035-4).
- [19] D. Organ, M. Fleming, T. Terry, T. Belytschko, Continuous meshless approximations for nonconvex bodies by diffraction and transparency, *Comput. Mech.* 18 (1996) 225–235, <https://doi.org/10.1007/BF00369940>.
- [20] N. Sukumar, B. Moran, T. Black, T. Belytschko, An element-free Galerkin method for three-dimensional fracture mechanics, *Comput. Mech.* 20 (1997) 170–175, <https://doi.org/10.1007/s004660050235>.
- [21] B.Z. P., J. Milan, Nonlocal Integral Formulations of Plasticity and Damage: Survey of Progress, *J. Eng. Mech.* 128 (2002) 1119–1149, [https://doi.org/10.1061/\(ASCE\)0733-9399\(2002\)128:11\(1119\)](https://doi.org/10.1061/(ASCE)0733-9399(2002)128:11(1119)).
- [22] R.D. Mindlin, Second gradient of strain and surface-tension in linear elasticity, *Int. J. Solids Struct.* 1 (1965) 417–438, [https://doi.org/10.1016/0020-7683\(65\)90006-5](https://doi.org/10.1016/0020-7683(65)90006-5).
- [23] E.C. Aifantis, On the Microstructural Origin of Certain Inelastic Models, *J. Eng. Mater. Technol.* 106 (1984) 326–330, <https://doi.org/10.1115/1.3225725>.

- [24] R. De Borst, H.-B. Mühlhaus, Gradient-dependent plasticity: Formulation and algorithmic aspects, *Int. J. Numer. Methods Eng.* 35 (1992) 521–539, <https://doi.org/10.1002/nme.1620350307>.
- [25] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, *Comput. Methods Appl. Mech. Eng.* 199 (2010) 2765–2778, <https://doi.org/10.1016/j.cma.2010.04.011>.
- [26] C. Miehe, F. Welschinger, M. Hofacker, Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations, *Int. J. Numer. Methods Eng.* 83 (2010) 1273–1311, <https://doi.org/10.1002/nme.2861>.
- [27] M.J. Borden, C.V. Verhoosel, M.A. Scott, T.J.R. Hughes, C.M. Landis, A phase-field description of dynamic brittle fracture, *Comput. Methods Appl. Mech. Eng.* 217–220 (2012) 77–95, <https://doi.org/10.1016/j.cma.2012.01.008>.
- [28] R.J.M. Geelen, Y. Liu, T. Hu, M.R. Tupek, J.E. Dolbow, A phase-field formulation for dynamic cohesive fracture, *Comput. Methods Appl. Mech. Eng.* 348 (2019) 680–711, <https://doi.org/10.1016/j.cma.2019.01.026>.
- [29] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, *Comput. Methods Appl. Mech. Eng.* 362 (2020), 112790, <https://doi.org/10.1016/j.cma.2019.112790>.
- [30] L. Zhang, L. Cheng, H. Li, J. Gao, C. Yu, R. Domel, Y. Yang, S. Tang, W.K. Liu, Hierarchical deep-learning neural networks: finite elements and beyond, *Comput. Mech.* 67 (2021) 207–230, <https://doi.org/10.1007/s00466-020-01928-9>.
- [31] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (2021) 218–229, <https://doi.org/10.1038/s42256-021-00302-5>.
- [32] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis, DeepXDE: A Deep Learning Library for Solving Differential Equations, *SIAM Rev* 63 (2021) 208–228, <https://doi.org/10.1137/19M1274067>.
- [33] J. Baek, J.-S. Chen, K. Susuki, A neural network-enhanced reproducing kernel particle method for modeling strain localization, *Int. J. Numer. Methods Eng.* 123 (2022) 4422–4454, <https://doi.org/10.1002/nme.7040>.
- [34] E. Haghighat, R. Juanes, SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, *Comput. Methods Appl. Mech. Eng.* 373 (2021), 113552, <https://doi.org/10.1016/j.cma.2020.113552>.
- [35] M.M. Almajid, M.O. Abu-Al-Saud, Prediction of porous media fluid flow using physics informed neural networks, *J. Pet. Sci. Eng.* 208 (2022), 109205, <https://doi.org/10.1016/j.petrol.2021.109205>.
- [36] E. Haghighat, D. Amini, R. Juanes, Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training, *Comput. Methods Appl. Mech. Eng.* 397 (2022), 115141, <https://doi.org/10.1016/j.cma.2022.115141>.
- [37] T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Trans. Neural Networks.* 6 (1995) 911–917, <https://doi.org/10.1109/72.392253>.
- [38] B. Shen, O. Stephansson, Modification of the G-criterion for crack propagation subjected to compression, *Eng. Fract. Mech.* 47 (1994) 177–189, [https://doi.org/10.1016/0013-7944\(94\)90219-4](https://doi.org/10.1016/0013-7944(94)90219-4).
- [39] H.-Y. Hu, J.-S. Chen, W. Hu, Error analysis of collocation method based on reproducing kernel approximation, *Numer. Methods Partial Differ. Equ.* 27 (2011) 554–580, <https://doi.org/10.1002/num.20539>.
- [40] J.-S. Chen, C.-T. Wu, S. Yoon, Y. You, A stabilized conforming nodal integration for Galerkin mesh-free methods, *Int. J. Numer. Methods Eng.* 50 (2001) 435–466, [https://doi.org/10.1002/1097-0207\(20010120\)50:2<435::AID-NME32>3.0.CO;2-A](https://doi.org/10.1002/1097-0207(20010120)50:2<435::AID-NME32>3.0.CO;2-A).
- [41] H. Wei, J.S. Chen, A damage particle method for smeared modeling of brittle fracture, *Int. J. Multiscale Comput. Eng.* 16 (2018) 303–324, <https://doi.org/10.1615/IntJMultCompEng.2018026133>.
- [42] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: *BT - 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, <http://arxiv.org/abs/1412.6980>.
- [43] J. Nocedal, S.J. Wright, Large-Scale Unconstrained Optimization. *Numer. Optim.*, Springer New York, New York, NY, 2006, pp. 164–192, https://doi.org/10.1007/978-0-387-40065-5_7.
- [44] J.-S. Chen, X. Zhang, T. Belytschko, An implicit gradient model by a reproducing kernel strain regularization in strain localization problems, *Comput. Methods Appl. Mech. Eng.* 193 (2004) 2827–2844, <https://doi.org/10.1016/j.cma.2003.12.057>.
- [45] R.U. Patil, B.K. Mishra, I.V. Singh, An adaptive multiscale phase field method for brittle fracture, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 254–288, <https://doi.org/10.1016/j.cma.2017.09.021>.
- [46] A. Muixí, A. Rodríguez-Ferran, S. Fernández-Méndez, A hybridizable discontinuous Galerkin phase-field model for brittle fracture with adaptive refinement, *Int. J. Numer. Methods Eng.* 121 (2020) 1147–1169, <https://doi.org/10.1002/nme.6260>.
- [47] A. Muixí, O. Marco, A. Rodríguez-Ferran, S. Fernández-Méndez, A combined XFEM phase-field computational model for crack growth without remeshing, *Comput. Mech.* 67 (2021) 231–249, <https://doi.org/10.1007/s00466-020-01929-8>.
- [48] A. Bobet, H.H. Einstein, Numerical modeling of fracture coalescence in a model rock material, *Int. J. Fract.* 92 (1998) 221–252, <https://doi.org/10.1023/A:1007460316400>.