



Nonlinear field-split preconditioners for solving monolithic phase-field models of brittle fracture

Alena Kopaničáková*, Hardik Kothari, Rolf Krause

Euler Institute, Università della Svizzera italiana, Lugano, Switzerland

Received 19 March 2022; received in revised form 27 September 2022; accepted 19 October 2022

Available online 15 November 2022

Abstract

One of the state-of-the-art strategies for predicting crack propagation, nucleation, and interaction is the phase-field approach. Despite its reliability and robustness, the phase-field approach suffers from burdensome computational cost, caused by the non-convexity of the underlying energy functional and a large number of unknowns required to resolve the damage gradients. In this work, we propose to solve such nonlinear systems in a monolithic manner using the Schwarz preconditioned inexact Newton (SPIN) method. The proposed SPIN method leverages the field split approach and minimizes the energy functional separately with respect to displacement and the phase-field, in an additive and multiplicative manner. In contrast to the standard alternate minimization, the result of this decoupled minimization process is used to construct a preconditioner for a coupled linear system, arising at each Newton's iteration. The overall performance and the convergence properties of the proposed additive and multiplicative SPIN methods are investigated by means of several numerical examples. A comparison with widely-used alternate minimization is also performed showing a significant reduction in terms of execution time. Moreover, we also demonstrate that this reduction grows even further with increasing problem size.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Phase-field fracture; Monolithic scheme; Nonlinear preconditioning; Inexact Newton

1. Introduction

Computational modeling of complex fracture processes, such as crack propagation, branching, or merging, plays an important role in the field of computational mechanics and engineering. The phase-field approaches to fracture allow to model such complex processes without explicitly representing discontinuities, and have therefore become very popular. The main idea behind the phase-field approach is to introduce a smooth indicator function that characterizes the state of the material. This phase-field function is also used to regularize sharp crack interfaces, giving rise to a volumetric approximation of the crack zones. The regularization is typically based on Ambrosio–Tortorelli elliptic functional [1], which models the transition between the broken and intact parts of the domain in a diffused manner. The size of the transition zone is controlled by a localization limiter, called the length-scale parameter. Using this volumetric approximation of the sharp crack interfaces, the numerical modeling of the

* Corresponding author.

E-mail addresses: alena.kopanicakova@usi.ch (A. Kopaničáková), hardik.kothari@usi.ch (H. Kothari), rolf.krause@usi.ch (R. Krause).

phase-field fractures can be carried out using the standard finite element method (FEM), without any need for computationally tedious tracking/remeshing algorithms.

In this work, we consider a phase-field fracture model based on a variational approach, which has been originally proposed in [2] as an extension to classical Griffith's energy-based principle [3]. Using such a variational approach, the fracture processes are modeled by minimizing the total potential energy of an underlying system. The first numerical implementation of the variational phase-field model has been presented by Bourdin et al. [4]. Subsequently, a thermodynamically consistent phase-field fracture formulation has been developed by Miehe et al. [5,6]. Since then, the phase-field fracture models have been extensively used and extended in various directions, for instance dynamics [7,8], large-deformations [9,10], or coupled multi-physics applications [11–13]. For a detailed overview of the phase-field fracture models, we refer the reader to [14,15].

The biggest computational challenge associated with phase-field fracture modeling is related to its burdensome computational cost, which is caused by three main factors.

Firstly, meshes with high resolution are required to resolve the fracture zones, which gives rise to nonlinear systems with a large number of unknowns. Adaptive mesh refinement techniques [16–19] and multiscale finite element methods [20] can be employed to alleviate the computational cost by reducing the number of unknowns. However, the mesh resolution and the number of unknowns increase inevitably with the number of cracks in the structure. Therefore, in order to obtain an accurate solution of complex fracture problems with feasible computational cost, it is crucial to employ efficient and scalable solution strategies.

Secondly, the non-convexity of the total potential energy, resulting from the coupling between the displacement and the phase-field field, hinders the convergence of standard Newton's method. Several remedies have been proposed to overcome this difficulty; for example, line-search and trust-region methods [21–23], arc-length control and path-following approaches [24,25], primal–dual active set method [18,26], augmented Lagrangian approach [27], modified Newton method [28], quasi-Newton methods [29,30], and Newton methods with inertia correction [31,32]. Although all of these methods for solving the arising coupled phase-field fracture problems are promising, they have not been widely adopted in the literature. This is most likely due to their complexity or lack of robustness. As a consequence, the majority of the phase-field fracture simulators utilize a staggered solution scheme, also called the alternate minimization (AM) method [6,33]. The idea behind the AM method is to minimize the energy functional separately with respect to the displacement and the phase-field, while the other respective variable is kept fixed. This gives rise to two convex minimization problems, which can be solved efficiently using standard solution strategies. Despite its robustness, the AM method exhibits slow convergence, which even deteriorates with increasing problem size [34]. Several approaches to accelerate the AM method have been proposed recently in the literature, e.g., over-relaxation strategies [35,36], stabilization techniques [37], or sub-stepping algorithms [38]. However, the applicability of the AM method to large-scale problems remains limited.

Thirdly, we point out that the linear and nonlinear systems arising after FEM discretization are often severely ill-conditioned. In particular, for second-order partial differential equations (PDEs), the condition number grows as $\mathcal{O}(h^{-2})$, where h denotes the mesh size. Moreover, in the case of phase-field fracture problems, the ill-conditioning is further aggravated due to the strongly varying stiffness in the structure. Consequently, the convergence rate of standard iterative solvers deteriorates with the increasing mesh resolution. As a remedy, preconditioning strategies [39], such as multilevel or domain decomposition methods, can be employed. In the case of phase-field fracture simulations, the efficiency of various linear and nonlinear multilevel methods has been demonstrated by several authors, see for example [22,35,40–44]. In the context of domain decomposition methods, standard linear (block) Jacobi and additive Schwarz methods have been investigated in [40,44]. A variant of the linear additive Schwarz method has been also developed in [45], where authors have decomposed the domain into two subdomains, related to localized and healthy parts of the domain. To the best of our knowledge, there have been no attempts to apply nonlinear domain decomposition methods to solve phase-field fracture problems so far.

Motivated by the observations above, the goal of this work is to propose a solution strategy, which can tackle the non-convexity of the coupled energy functional in a robust manner, and which acts as a scalable preconditioner. In particular, we develop a variant of the Schwarz Preconditioned Inexact Newton (SPIN) method [46–49], specifically tailored to solve phase-field fracture simulations. The proposed solution strategy is designed to solve the arising nonlinear problems in a monolithic manner by taking advantage of the underlying structure of the coupled problem. More precisely, we construct a nonlinear preconditioner by partitioning the degrees of freedom into two sets, related to the displacement and the phase-field. Similar to the AM method, the energy functional is minimized separately

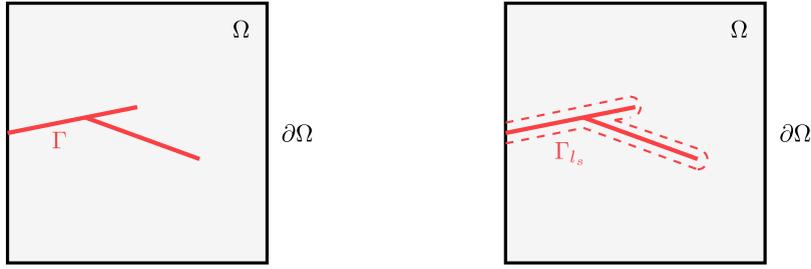


Fig. 1. Left: Computational domain Ω with the sharp crack Γ . Right: Phase-field regularization of the same crack, denoted by Γ_{l_s} . The size of the regularized zone is controlled by the length-scale parameter l_s .

with respect to the displacement and the phase-field. In contrast to the AM method, the result of this minimization process is used to construct the preconditioner for a coupled linear system, arising at each Newton’s iteration. Using several representative benchmark problems, we show that the newly developed additive and multiplicative SPIN methods are robust, significantly more efficient than the standard AM method and that their performance does not deteriorate with increasingly larger loading steps or increasing mesh resolution.

This paper is organized as follows. In Section 2, we review the variational phase-field fracture model, including the finite element discretization of the arising Euler–Lagrange equations. In Section 3, we briefly recall the solution strategies, which are typically used for solving phase-field fracture problems. Section 4 introduces additive and multiplicative variants of the SPIN method, specifically tailored for the phase-field fracture simulations. The benchmark problems used to test the proposed SPIN methods are introduced in Section 5. The convergence and performance studies of the SPIN methods, including a comparison with the standard AM method, are presented in Section 6. Eventually, the presented work and possible future extensions are summarized in Section 7.

2. The phase-field fracture model of brittle fracture

In this section, we briefly introduce the phase-field fracture model for brittle fracture. To this aim, we consider a linear elastic body $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, with boundary $\partial\Omega$. The boundary $\partial\Omega$ is further decomposed into Dirichlet boundary $\partial\Omega_D$ and Neumann boundary $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$, on which Dirichlet and Neumann boundary conditions are applied, respectively. Due to the application of these boundary conditions or the external body forces, the body Ω undergoes deformation $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$, which in turn causes crack propagation. In this work, we consider a quasi-static loading process, where the pseudo time $t = 1, \dots, T$ is used to index the deformation state. Thus, the Dirichlet and Neumann boundary conditions are prescribed at time t as $\mathbf{u}_D^t : \partial\Omega_D \rightarrow \mathbb{R}^d$ and $\mathbf{t}_N^t : \partial\Omega_N \rightarrow \mathbb{R}^d$, respectively.

Through this work, we consider linear elastic materials, constitutive law of which is provided by Hooke’s law. The Cauchy stress tensor is then given as $\boldsymbol{\sigma}(\mathbf{u}) := 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) + \lambda\text{tr}(\boldsymbol{\varepsilon}(\mathbf{u}))\mathbf{I}$, where λ and μ are the Lamé parameters, $\text{tr}(\cdot)$ denotes the trace operator, and $\mathbf{I} \in \mathbb{R}^{d \times d}$ stands for the identity tensor. The second-order strain tensor is given as $\boldsymbol{\varepsilon}(\mathbf{u}) := \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top)$.

2.1. Variational formulation for brittle fracture

The variational approach to fracture [2] allows us to find the displacement \mathbf{u}^t and fracture surface Γ^t at time t by minimizing the total potential energy of the body Ω . Thus, at each pseudo-time-step, the pair (\mathbf{u}^t, Γ^t) can be obtained by solving the following minimization problem:

$$(\mathbf{u}^t, \Gamma^t) = \underset{\substack{\mathbf{u}=\mathbf{u}_D^t \text{ on } \partial\Omega_D \\ \Gamma \supset \Gamma^{t-1}}}{\arg \min}}{\mathcal{E}(\mathbf{u}, \Gamma) := \int_{\Omega \setminus \Gamma} \psi_e(\mathbf{u}) \, d\Omega + \int_{\Gamma} \mathcal{G}_c \, d\Gamma}, \tag{1}$$

where the first term denotes the strain energy functional, given as $\psi_e(\mathbf{u}) = \frac{1}{2}\boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})$. The second term describes the fracture energy, defined as the critical energy release \mathcal{G}_c integrated over the sharp fracture surface Γ .

Remark 1. For simplicity, the energy formulation (1) neglects external forces.

The direct minimization of the energy functional (1) is computationally challenging as it requires explicit tracking of the crack paths and the cumbersome remeshing of the domain becomes essential. The phase-field approach overcomes this difficulty by replacing the fracture energy in (1) with its volumetric approximation [33], see also illustration depicted on Fig. 1. To this aim, we introduce a phase-field variable $c : \Omega \rightarrow [0, 1]$, which represents the state of the domain, where $c = 0$ denotes the intact state, $c = 1$ identifies fully broken state and $c \in (0, 1)$ denotes a smooth transition between the both states. Using this phase-field variable c , the fracture is modeled in a diffused manner and the fracture energy from (1) can be approximated as

$$\int_{\Gamma} \mathcal{G}_c \, d\Gamma \approx \frac{\mathcal{G}_c}{c_\omega} \int_{\Omega} \left(\frac{\omega(c)}{l_s} + l_s |\nabla c|^2 \right) d\Omega. \tag{2}$$

Here, the symbol l_s denotes the length-scale parameter that controls the width of the transition zone between the broken and unbroken states. The function $\omega(\cdot)$ defines the decaying profile of the phase-field and $c_\omega := 4 \int_0^1 \sqrt{\omega(c)} \, dc$ denotes an induced normalization constant. Using (2), we can now reformulate the energy functional (1) as

$$\mathcal{E}(\mathbf{u}, c) := \int_{\Omega} g(c) \, \psi_e(\mathbf{u}) \, d\Omega + \frac{\mathcal{G}_c}{c_\omega} \int_{\Omega} \left(\frac{\omega(c)}{l_s} + l_s |\nabla c|^2 \right) d\Omega, \tag{3}$$

where $g(\cdot)$ is a degradation function, which accounts for the loss of stiffness in the fracture. Particular choices of the functions $g(\cdot)$ and $\omega(\cdot)$ give rise to different phase-field formulations [50,51]. In this work, we adopt the widely used AT-2 model with $g(c) := (1 - c)^2$, $\omega(c) := c^2$ and $c_\omega = 2$, see [1,4,5] for details. Given these particular choices of $g(\cdot)$, $\omega(\cdot)$ and c_ω , the minimizer of (3) asymptotically converges to the minimizer of (1) as $l_s \rightarrow 0$, see Γ -convergence results in [52].

The main limitation of the phase-field fracture model (3) is that it allows for crack interpenetration under compressive loading and that the nucleation threshold is symmetric in tension and compression [53]. To avoid this unphysical behavior, several modifications of the model (3) have been proposed. The most common approach is to decompose the strain energy function ψ_e into a tensile and a compressive part and to allow only the tensile part to be influenced by the damage. Several types of decomposition have been introduced in the literature, for instance, volumetric–deviatoric split [54], or the no-tension model [55]. In this work, we employ the spectral decomposition, originally proposed in [5]. The strain energy functional is decomposed as $\psi_e(\mathbf{u}) := \psi_e^+(\mathbf{u}) + \psi_e^-(\mathbf{u})$, where $\psi_e^\pm(\mathbf{u}) := \frac{1}{2} \lambda [\text{tr}^\pm(\boldsymbol{\varepsilon})]^2 + \mu \boldsymbol{\varepsilon}^\pm \cdot \boldsymbol{\varepsilon}^\pm$. Here, the positive and negative parts of the strain tensor are obtained as $\boldsymbol{\varepsilon}^\pm := \sum_i \langle \varepsilon_i \rangle_\pm \mathbf{n}_i \otimes \mathbf{n}_i$, where $\{\varepsilon_i\}_i$ and $\{\mathbf{n}_i\}_i$ denote the eigenvalues and the eigenvectors of $\boldsymbol{\varepsilon}$, respectively. Moreover, the ramp functions are defined as $\langle x \rangle_+ = \max(x, 0)$ and $\langle x \rangle_- = \min(x, 0)$. Employing this additive decomposition of the strain energy, and the particular choices of $g(\cdot)$, $\omega(\cdot)$ and c_ω , we can reformulate (3) as

$$\mathcal{E}(\mathbf{u}, c) := \underbrace{\int_{\Omega} \left((1 - c)^2 \, \psi_e^+(\mathbf{u}) + \psi_e^-(\mathbf{u}) \right) \, d\Omega}_{=: \mathcal{E}_e} + \underbrace{\frac{\mathcal{G}_c}{2} \int_{\Omega} \left(\frac{c^2}{l_s} + l_s |\nabla c|^2 \right) \, d\Omega}_{=: \mathcal{E}_f}. \tag{4}$$

Please note that the degradation function $g(c) = (1 - c)^2$ in (4) affects only the positive part of the elastic energy.

2.2. Minimization problem

Using the phase-field fracture energy functional (4), the quasi-static brittle fracture problem is expressed as

$$(\mathbf{u}^t, c^t) = \underset{\substack{\mathbf{u}=\mathbf{u}_D \text{ on } \partial\Omega_D \\ \partial_t c \geq 0}}{\arg \min} \mathcal{E}(\mathbf{u}, c). \tag{5}$$

where the constraint $\partial_t c \geq 0$ imposes the positive evolution of the phase-field, which in turn ensures the fracture irreversibility. From the numerical point of view, the irreversibility condition can be enforced directly by means of inequality constraints. Consequently, the constrained optimization algorithms, e.g., the active-set methods [18], or the interior-point methods [31] need to be employed. Alternatively, the irreversibility condition can be modeled indirectly, for example by introducing a history field [6], using penalization approaches [56,57], or by considering relaxed equality condition [4].

In this work, we opt for an indirect approach and impose the irreversibility condition using the quadratic penalty method [57]. This allows us to reformulate the constrained minimization problem (5) as the following unconstrained minimization problem:

$$(\mathbf{u}^t, c^t) = \arg \min_{\mathbf{u}=\mathbf{u}_D^t \text{ on } \partial\Omega_D} \Psi(\mathbf{u}, c) := \mathcal{E}(\mathbf{u}, c) + \frac{\gamma}{2} \int_{\Omega} ((c - c^{t-1})_-)^2 d\Omega, \tag{6}$$

where $\gamma \in \mathbb{R}^+$ denotes the penalty parameter. The solution of (6) converges to the solution of (5) as $\gamma \rightarrow \infty$. However, very large values of the penalty parameter γ give rise to ill-conditioning. Therefore, it is beneficial to employ the smallest possible γ , which ensures that the irreversibility condition is satisfied with sufficient accuracy. Following [57], we choose γ as

$$\gamma = \frac{\mathcal{G}_c}{l_s} \left(\frac{1}{\tau_{\text{irr}}^2} - 1 \right), \tag{7}$$

where τ_{irr} denotes a tolerance, with which the irreversibility condition should be satisfied. This explicit formulation for the parameter γ has been derived by exploiting the notion of the optimal phase-field profile as well as the Γ -convergence in [57].

2.3. Euler–Lagrange equations and finite-element discretization

We define function spaces $\mathcal{V}^t = \{\mathbf{v} \in [H^1(\Omega)]^d : \mathbf{v}|_{\partial\Omega_D} = \mathbf{u}_D^t\}$ and $\mathcal{V}_0 = \{\mathbf{v} \in [H^1(\Omega)]^d : \mathbf{v}|_{\partial\Omega_D} = \mathbf{0}\}$, where H^1 denotes the Sobolev space that consists of all square-integrable functions and have square-integrable first weak derivatives. At each time t , the first order necessary optimality conditions of the minimization problem (6) are obtained by differentiating the energy functional Ψ , which gives rise to the following Euler–Lagrange equations: Find a pair $(\mathbf{u}, c) \in \mathcal{V}^t \times H^1(\Omega)$, such that

$$\nabla_{\mathbf{u}} \Psi(\mathbf{u}, c; \mathbf{v}) = 0, \quad \forall \mathbf{v} \in \mathcal{V}_0, \quad \text{and} \quad \nabla_c \Psi(\mathbf{u}, c; w) = 0, \quad \forall w \in H^1(\Omega), \tag{8}$$

where

$$\begin{aligned} \nabla_{\mathbf{u}} \Psi(\mathbf{u}, c; \mathbf{v}) &= ((1 - c)^2 \boldsymbol{\sigma}^+(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_{\Omega} + (\boldsymbol{\sigma}^-(\mathbf{u}), \boldsymbol{\varepsilon}(\mathbf{v}))_{\Omega}, \\ \nabla_c \Psi(\mathbf{u}, c; w) &= (2(c - 1)\psi_e^+(\mathbf{u}), w)_{\Omega} + \frac{\mathcal{G}_c}{l_s} [(c, w)_{\Omega} + l_s^2 (\nabla c, \nabla w)_{\Omega}] + \gamma ((c - c^{t-1})_-, w)_{\Omega}. \end{aligned} \tag{9}$$

Here, the symbol $(\cdot, \cdot)_{\Omega}$ denotes L^2 inner product over the domain Ω .

We discretize (8) using a Galerkin finite element (FE) method. To this aim, we consider a shape regular, quasi-uniform, conforming quadrilateral mesh \mathcal{Q}_h of the domain Ω . Moreover, we define the continuous FE spaces $\mathcal{V}_h^t = \{\mathbf{v}_h \in [H^1(\Omega)]^d : \mathbf{v}_h|_{\mathcal{Q}} \in \mathcal{Q}_h \text{ is bilinear}, \mathbf{v}_h|_{(\partial\mathcal{Q}_h)_D} = \mathbf{u}_D^t\}$, and $\mathcal{W}_h = \{w_h \in H^1(\Omega) : w_h|_{\mathcal{Q}} \text{ is bilinear}\}$. Both FE spaces, \mathcal{V}_h and \mathcal{W}_h , are spanned by nodal basis functions $\{\phi_i\}_{i=1}^n$, which we use to approximate the displacement and the phase-field as

$$\mathbf{u} \approx \mathbf{u}_h(\mathbf{x}) = \sum_{i=1}^n \sum_{k=1}^d \mathbf{e}_k \phi_i(\mathbf{x}) U_i, \quad \text{and} \quad c \approx c_h(\mathbf{x}) = \sum_{i=1}^n \phi_i(\mathbf{x}) C_i, \tag{10}$$

where $\{\mathbf{e}_k\}_{k=1, \dots, d}$ denotes the Euclidean basis of \mathbb{R}^d . The vectors $\mathbf{U} \in \mathbb{R}^{dn}$ and $\mathbf{C} \in \mathbb{R}^n$ contain the nodal coefficients for the displacement and the phase-field, respectively.

Using the finite element approximations \mathbf{u}_h and c_h , the discretized Euler–Lagrange equations (8) can be expressed by means of residuals $F_u : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{dn}$ and $F_c : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, components of which are given as follows:

$$\begin{aligned} (F_u(\mathbf{U}, \mathbf{C}))_i &= ((1 - c_h)^2 \boldsymbol{\sigma}^+(\mathbf{u}_h), \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_i))_{\mathcal{Q}_h} + (\boldsymbol{\sigma}^-(\mathbf{u}_h), \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_i))_{\mathcal{Q}_h}, \quad \text{for } k = 1, \dots, d, \\ (F_c(\mathbf{U}, \mathbf{C}))_i &= (2(c_h - 1)\psi_e^+(\mathbf{u}_h), \phi_i)_{\mathcal{Q}_h} + \frac{\mathcal{G}_c}{l_s} [(c_h, \phi_i)_{\mathcal{Q}_h} + l_s^2 (\nabla c_h, \nabla \phi_i)_{\mathcal{Q}_h}] + \gamma ((c_h - c_h^{t-1})_-, \phi_i)_{\mathcal{Q}_h}. \end{aligned} \tag{11}$$

Here, the subscript i stands for the nodal index. Please note, that the component $(F_u(\mathbf{U}, \mathbf{C}))_i$ is a vector of dimension d . Using F_u and F_c , the residual of the coupled problem $F : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n}$ is defined as $F(\mathbf{U}, \mathbf{C}) := \begin{bmatrix} F_u(\mathbf{U}, \mathbf{C}) \\ F_c(\mathbf{U}, \mathbf{C}) \end{bmatrix}$.

Algorithm 1: Alternate Minimization (AM)**Data:** $F : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n}$, $F_u : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{dn}$, $F_c : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\mathbf{U}^{(0)} \in \mathbb{R}^{dn}$, $\mathbf{C}^{(0)} \in \mathbb{R}^n$ **Result:** $\mathbf{U}^{(k)}$, $\mathbf{C}^{(k)}$ $k \leftarrow 0$ **while** $\|F(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})\| > \epsilon_{\text{rel, glob, nonl}} \|F(\mathbf{U}^{(0)}, \mathbf{C}^{(0)})\|$ **do** For fixed $\mathbf{C}^{(k)}$, find $\mathbf{U}^{(k+1)}$ by solving $F_u(\mathbf{U}^{(k+1)}, \mathbf{C}^{(k)}) = 0$ \triangleright use ND/NK/INK method For fixed $\mathbf{U}^{(k+1)}$, find $\mathbf{C}^{(k+1)}$ by solving $F_c(\mathbf{U}^{(k+1)}, \mathbf{C}^{(k+1)}) = 0$ \triangleright use ND/NK/INK method $k \leftarrow k + 1$

In this work, we solve the arising discrete coupled problems using nonlinearly preconditioned inexact Newton method, which requires knowledge about the Jacobian $F' : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n \times (d+1)n}$. Similarly to the residual F , the Jacobian $\mathbf{J} = F'(\mathbf{U}, \mathbf{C})$ has also block structure, i.e., $\mathbf{J} = \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{uc} \\ \mathbf{J}_{cu} & \mathbf{J}_{cc} \end{bmatrix}$, where its components are given as

$$\begin{aligned}
(\mathbf{J}_{uu})_{i,j} &= \frac{\partial (F_u)_i}{\partial U_j} = ((1 - c_h)^2 \boldsymbol{\sigma}^+(\mathbf{e}_l \phi_j), \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_i))_{\mathcal{Q}_h} + (\boldsymbol{\sigma}^-(\mathbf{e}_l \phi_j), \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_i))_{\mathcal{Q}_h}, & \text{for } k, l = 1, \dots, d, \\
(\mathbf{J}_{uc})_{i,j} &= \frac{\partial (F_u)_i}{\partial C_j} = (2(c_h - 1) \phi_j \boldsymbol{\sigma}^+(\mathbf{u}_h), \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_i))_{\mathcal{Q}_h}, & \text{for } k = 1, \dots, d, \\
(\mathbf{J}_{cu})_{i,j} &= \frac{\partial (F_c)_i}{\partial U_j} = (2(c_h - 1) \boldsymbol{\sigma}^+(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{e}_k \phi_j), \phi_i)_{\mathcal{Q}_h}, & \text{for } k = 1, \dots, d, \\
(\mathbf{J}_{cc})_{i,j} &= \frac{\partial (F_c)_i}{\partial C_j} = (2 \phi_j \psi_e^+(\mathbf{u}_h), \phi_i)_{\mathcal{Q}_h} + \frac{\mathcal{G}_c}{l_s} [(\phi_j, \phi_i)_{\mathcal{Q}_h} + l_s^2 (\nabla \phi_j, \nabla \phi_i)_{\mathcal{Q}_h}] + \gamma H^-(c_h - c_h^{t-1}) (\phi_j, \phi_i)_{\mathcal{Q}_h}.
\end{aligned} \tag{12}$$

Here, the symbol $H^-(\cdot)$ denotes the Heaviside function, defined as

$$H^-(x) = \begin{cases} 1, & \text{if } x < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

3. Standard solution strategies

The solution of the energy minimization problem (6) can be found by solving the nonlinear system of equations associated with the following first-order necessary optimality conditions: Find $\mathbf{U}^* \in \mathbb{R}^{dn}$ and $\mathbf{C}^* \in \mathbb{R}^n$ such that

$$F(\mathbf{U}, \mathbf{C}) = \begin{bmatrix} F_u(\mathbf{U}, \mathbf{C}) \\ F_c(\mathbf{U}, \mathbf{C}) \end{bmatrix} = \mathbf{0}. \tag{14}$$

The nonlinear system (14) can be solved using a monolithic solution scheme or by employing the alternate minimization (AM) method. The monolithic solution scheme requires a solution of the coupled problem, which is computationally challenging due to the non-convexity of energy functional induced by the coupling between the displacement and the phase-field. In contrast, the AM method decouples the nonlinear problem into two subproblems $F_u(\mathbf{U}, \mathbf{C}) = 0$ and $F_c(\mathbf{U}, \mathbf{C}) = 0$ associated with the displacement and the phase-field, respectively. The subproblems are then solved alternatively in order to update the displacement \mathbf{U} and the phase-field \mathbf{C} , see Algorithm 1. The biggest advantage of the AM method, and also the main reason for its popularity, is the fact that both subproblems are convex [4]. Hence, the standard solution strategies can be readily applied. Here, we point out that subproblems might not necessarily be convex if a different phase-field fracture model is employed. For instance, if the anisotropy [23], or nonlinear constitutive laws [10] are considered.

3.1. Newton's method

The monolithic solution scheme requires a solution of a coupled nonlinear system of equations at each (pseudo-)time-step, while AM method requires a solution of two nonlinear systems of equations at each iteration. A standard approach to solve such a system of nonlinear equations is to employ Newton's method, as it is simple and locally

quadratically convergent. However, the convergence of the method relies on a good initial guess. Global convergence, convergence to a stationary point irrespectively of the choice of the initial guess, can be ensured by employing some globalization strategy, e.g., line-search or trust-region method [58]. In this work, we employ a cubic backtracking line-search algorithm [59, Algorithm A6.3.1, pages 325–327] with strong Wolfe conditions, see Appendix B for more details.

Let us consider a system of nonlinear equations of the following form: $R(\mathbf{x}) = 0$, where $R : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$. An iteration of Newton's method within a line-search framework is given as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}, \quad (15)$$

where $\mathbf{p}^{(k)}$ denotes a search-direction and $\alpha^{(k)} \in \mathbb{R}^+$ is step size obtained by the backtracking algorithm. At each iteration, the search-direction $\mathbf{p}^{(k)}$ is acquired by solving the linear system of equations

$$R'(\mathbf{x}^{(k)}) \mathbf{p}^{(k)} = -R(\mathbf{x}^{(k)}). \quad (16)$$

A solution of this linear system can be found by employing a direct or an iterative method, giving rise to different variants of Newton's method. In this work, we consider three particular variants of Newton's method, see also Algorithm 2.

3.1.1. Newton-direct (ND) method

In the context of phase-field fracture simulations, direct methods, such as LU or Cholesky factorization, are typically used to solve the linear systems arising at each Newton's iteration [4,32]. The popularity of direct solvers results from their remarkable robustness as well as the open-source availability of highly efficient implementations. However, the computational cost and memory requirements of direct methods grow with increasing problem size. In particular, modern sparse direct solvers minimize the amount of required floating-point operations as well as storage by exploiting the sparsity pattern of the matrices. In this way, the factorization of the system with n unknowns can be performed using $\mathcal{O}(n^{3/2})$ flops in two spatial dimensions and $\mathcal{O}(n^2)$ flops in three spatial dimensions. Thus, when n becomes too large ($>500,000$ dofs), the computational cost and memory demands induced by the fill-in become computationally intractable. Moreover, direct methods cannot be easily parallelized, which in turn limits their scalability.

3.1.2. Newton-Krylov (NK) method

An alternative to the ND method is the Newton–Krylov (NK) method, which employs a Krylov subspace method to solve the linear system of equations arising at each Newton's iteration. The Krylov subspace methods are the iterative methods, which find the solution of the linear system of equations by projecting a sequence of iterates onto the Krylov subspaces. Krylov methods have smaller memory requirements compared to direct methods and are well-suited for parallel processing environments, as their main building blocks e.g., matrix–vector multiplication, or inner product, allow for hybrid parallelization.

Based on the problem at hand, a specific Krylov method might be suitable to solve the problem efficiently. In the context of phase-field fracture problems considered in this work, the arising coupled linear systems can be solved efficiently using the MINRES method. This is due to the fact that the Jacobian matrices of the coupled nonlinear system are symmetric but not necessarily positive definite due to the non-convex nature of the problem. For the AM method, the conjugate gradient (CG) method represents an ideal choice, as the Jacobians that arise while solving displacement and the phase-field subproblems are symmetric positive definite, due to the convexity of F_u and F_c .

A major drawback of Krylov methods is that their convergence speed deteriorates with increasing problem size and increasing condition number. As a consequence, preconditioning strategies have to be employed in order to accelerate convergence. In this work, we precondition the coupled linear systems using a nonlinear additive and multiplicative field-split preconditioner, described in Section 4. In order to precondition the linear systems, which arise while solving the displacement and the phase-field subproblems, we employ an algebraic multigrid (AMG) method.

3.1.3. Inexact Newton-Krylov (INK) method

Although NK methods are an improvement upon the ND methods, solving the arising linear systems accurately is computationally expensive. Moreover, if the current iterate $\mathbf{x}^{(k)}$ is far from a solution, the local linear model

Algorithm 2: Newton/Inexact Newton (ND/NK/INK)**Data:** $R : \mathbb{R}^n \rightarrow \mathbb{R}^n, \mathbf{x}^{(0)} \in \mathbb{R}^n$ **Result:** $\mathbf{x}^{(k)}$ $k \leftarrow 0$ **while** $\|R(\mathbf{x}^{(k)})\| > \epsilon_{\text{rel.sub.nonl}} \|R(\mathbf{x}^{(0)})\|$ **do**Find $\mathbf{p}^{(k)}$ by solving $R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -R(\mathbf{x}^{(k)})$ \triangleright solve linear system**if** ND **then**

solve the system using a direct solver

if NK **then**

solve the system using a Krylov method, such that

$$\|R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} + R(\mathbf{x}^{(k)})\| \leq \max(\epsilon_{\text{rel.lin}} \|R(\mathbf{x}^{(k)})\|, \epsilon_{\text{abs.lin}})$$

if INK **then**

solve the system using a Krylov method, such that

$$\|R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} + R(\mathbf{x}^{(k)})\| \leq \max(\eta \|R(\mathbf{x}^{(k)})\|, \epsilon_{\text{abs.lin}})$$

Find $\alpha^{(k)}$ using a backtracking algorithm \triangleright use line-search $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}$ \triangleright update iterate $k \leftarrow k + 1$

$R(\mathbf{x}^{(k)}) + R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)}$ might not approximate the functional R well enough. Since $R(\mathbf{x}^{(k)}) + R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)}$ is also the residual of the linear system (16), obtaining $\mathbf{p}^{(k)}$ by solving $R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -R(\mathbf{x}^{(k)})$ exactly might cause over-solving, which in turn leads to little or no progress towards a solution. As a result, far from the solution, it might be cheaper as well as more effective to obtain less accurate Newton's steps.

Inexact Newton (IN) methods [60] incorporate the aforementioned observations and determine how accurately the linear systems should be solved at each Newton's iteration. More precisely, the search-direction $\mathbf{p}^{(k)}$ is obtained by solving (16), such that

$$\|R'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} + R(\mathbf{x}^{(k)})\| \leq \eta \|R(\mathbf{x}^{(k)})\|. \quad (17)$$

Following [61], we employ a simple, but effective, choice of the parameter η , i.e., $\eta = 10^{-4}$. In this way, the induced tolerance $\eta \|R(\mathbf{x}^{(k)})\|$, with which (16) is solved, is updated at each iteration by taking into account the information about the current residual $R(\mathbf{x}^{(k)})$. This tolerance adapts proportionally to $\|R(\mathbf{x}^{(k)})\|$, which avoids over-solving in the initial phases of the solution process and ensures the quadratic convergence of IN method close to a solution [60]. In this work, we solve the arising linear systems inexactly, using preconditioned Krylov methods, which gives rise to the inexact Newton–Krylov (INK) method.

4. Field-split Schwarz preconditioned inexact Newton (SPIN) method

Standard Newton's method converges quadratically to a solution if a good initial guess is provided. However, if a good initial guess is not known, Newton's methods might exhibit very slow convergence until a local neighborhood of a solution is approached. Slow convergence is typically associated with unbalanced and highly localized nonlinearities. In the context of the phase-field fracture problems considered in this work, the unbalanced nonlinearities occur due to coupling between the displacement and the phase-field, and due to locally varying material stiffness and steep gradients of the phase-field function.

Nonlinear preconditioning strategies, such as SPIN [46,47], RASPEN [62], or nonlinear elimination [63] can enhance the convergence of Newton's method by rebalancing the nonlinearities, or by transforming the basis of the solution space. These nonlinear preconditioning strategies utilize the decomposition of the solution space into multiple subspaces, for instance, related to different parts of the computational domain or the different fields.

In this work, we adopt the SPIN methodology and instead of solving the original coupled nonlinear system (14), we solve an equivalent nonlinearly preconditioned system. This nonlinearly preconditioned system is constructed such that it has the same root as the original nonlinear system, but its nonlinearities are more balanced. Motivated by the robustness of the AM method, we construct the nonlinearly preconditioned system of equations using the

field-split approach, thus by decomposing the original problem into two subproblems that are associated with the displacement and the phase-field. From the algorithmic point of view, the SPIN strategy can be performed in an additive or multiplicative manner, which gives rise to two distinct variants of the SPIN method, named ASPIN and MSPIN, respectively.

The additive approach is more suitable for parallel computing environments, as nonlinear subproblems associated with different subspaces can be solved simultaneously. In contrast, the multiplicative approach is inherently sequential, but it usually exhibits faster convergence than the additive approach.

4.1. Field-split ASPIN and MSPIN methods

Let $G : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n}$ be a nonlinear preconditioner. Using the field-split approach, the preconditioner G is defined as follows

$$G(\mathbf{U}, \mathbf{C}) = \begin{bmatrix} G_u(\mathbf{U}, \mathbf{C}) \\ G_c(\mathbf{U}, \mathbf{C}) \end{bmatrix}, \quad (18)$$

where $G_u : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{dn}$ and $G_c : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ are solution operators associated with displacement and phase-field, respectively. Ideally, the functions G_u and G_c approximate the inverse of nonlinear operators F_u and F_c , i.e., $G_u \approx F_u^{-1}$, and $G_c \approx F_c^{-1}$. We, therefore, expect that at the fixed point, i.e., at the solution of $F(\mathbf{U}, \mathbf{C}) = 0$, the following relation holds:

$$[\mathbf{U}, \mathbf{C}]^\top = G(\mathbf{U}, \mathbf{C}). \quad (19)$$

Using this fixed point iteration, we can construct nonlinearly preconditioned residual $\mathcal{F} : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n}$, as follows

$$\mathcal{F}(\mathbf{U}, \mathbf{C}) := [\mathbf{U}, \mathbf{C}]^\top - G(\mathbf{U}, \mathbf{C}) = 0. \quad (20)$$

The nonlinearly preconditioned system of equations (20) has the same solution as the original nonlinear system associated with the coupled phase-field fracture problem (14). However, the preconditioned system (20) should be easier to solve, as the nonlinearities arising due to the coupling between displacement and phase-field should be more balanced.

An exact form of the preconditioner G is not known apriori. Therefore, we construct the preconditioned residual \mathcal{F} by solving the nonlinear subproblems associated with the displacement and the phase-field. In this case, $G_u(\mathbf{U}, \mathbf{C})$ can be obtained as \mathbf{U}_{prec} , which solves $F_u(\mathbf{U}_{\text{prec}}, \mathbf{C}) = 0$, for a fixed \mathbf{C} . Similarly, $G_c(\mathbf{U}, \mathbf{C})$ can be obtained as \mathbf{C}_{prec} , which solves $F_c(\mathbf{U}, \mathbf{C}_{\text{prec}}) = 0$, for a fixed \mathbf{U} . In practice, the nonlinear preconditioner \mathcal{F} can be constructed in an additive or a multiplicative manner, which gives rise to two different variants of the SPIN method, namely ASPIN and MSPIN. In the additive case, the preconditioned nonlinear system \mathcal{F}^{add} is defined as follows

$$\mathcal{F}^{\text{add}}(\mathbf{U}, \mathbf{C}) := \begin{bmatrix} \mathbf{U} - G_u(\mathbf{U}, \mathbf{C}) \\ \mathbf{C} - G_c(\mathbf{U}, \mathbf{C}) \end{bmatrix} = \mathbf{0}. \quad (21)$$

Here, both subproblems can be solved simultaneously, as their solution depends only on the values of displacement \mathbf{U} and phase-field \mathbf{C} . In the multiplicative case, the preconditioned residual $\mathcal{F}^{\text{mult}}$ has the following form:

$$\mathcal{F}^{\text{mult}}(\mathbf{U}, \mathbf{C}) := \begin{bmatrix} \mathbf{U} - G_u(\mathbf{U}, \mathbf{C}) \\ \mathbf{C} - G_c(\mathbf{U} - G_u(\mathbf{U}, \mathbf{C}), \mathbf{C}) \end{bmatrix} = \mathbf{0}. \quad (22)$$

Here, the subproblems have to be solved sequentially, as the solution of the phase-field subproblem relies on the solution of the displacement subproblem, namely $G_u(\mathbf{U}, \mathbf{C})$.

Once, the nonlinearly preconditioned system \mathcal{F} is formed, we can find its root using the inexact-Newton method. Thus, on each k th iteration, the search direction $\mathbf{p}^{(k)}$ is found by solving the following linear system of equations:

$$\mathcal{F}'(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})\mathbf{p}^{(k)} = -\mathcal{F}(\mathbf{U}^{(k)}, \mathbf{C}^{(k)}), \quad (23)$$

where \mathcal{F}' denotes the Jacobian of \mathcal{F} . However, it is not straightforward to construct the Jacobian \mathcal{F}' explicitly, or to evaluate its action on a vector, as the preconditioned residual \mathcal{F} is defined implicitly. Following [47], we can approximate the Jacobian \mathcal{F}' utilizing the Jacobian F' of the original residual F . In the additive case, the

Algorithm 3: ASPIN / MSPIN

Data: $F : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1)n}$, $F_u : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^{dn}$, $F_c : \mathbb{R}^{dn} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\mathbf{U}^{(0)} \in \mathbb{R}^{dn}$, $\mathbf{C}^{(0)} \in \mathbb{R}^n$

Result: $\mathbf{U}^{(k)}$, $\mathbf{C}^{(k)}$

$k \leftarrow 0$

while $\|F(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})\| \geq \epsilon_{\text{rel.glob.nonl}} \|F(\mathbf{U}^{(0)}, \mathbf{C}^{(0)})\|$ **do**

if additive SPIN then

For fixed $\mathbf{C}^{(k)}$ find \mathbf{U}_{prec} by solving $F_u(\mathbf{U}_{\text{prec}}, \mathbf{C}^{(k)}) = 0$ ▷ use INK method

For fixed $\mathbf{U}^{(k)}$ find \mathbf{C}_{prec} by solving $F_c(\mathbf{U}^{(k)}, \mathbf{C}_{\text{prec}}) = 0$ ▷ use INK method

Form $\mathbf{P}_{\text{add}}^{(k)}$ by means of (28) ▷ construct ASPIN preconditioner

if multiplicative SPIN then

For fixed $\mathbf{C}^{(k)}$ find \mathbf{U}_{prec} by solving $F_u(\mathbf{U}_{\text{prec}}, \mathbf{C}^{(k)}) = 0$ ▷ use INK method

For fixed \mathbf{U}_{prec} find \mathbf{C}_{prec} by solving $F_c(\mathbf{U}_{\text{prec}}, \mathbf{C}_{\text{prec}}) = 0$ ▷ use INK method

Form $\mathbf{P}_{\text{mult}}^{(k)}$ by means of (29) ▷ construct MSPIN preconditioner

$\mathbf{s}_{\text{add/mult}}^{(k)} \leftarrow [\mathbf{U}^{(k)} - \mathbf{U}_{\text{prec}}, \mathbf{C}^{(k)} - \mathbf{C}_{\text{prec}}]^\top$ ▷ evaluate nonlinearly preconditioned residual

$\mathbf{J}^{(k)} \leftarrow F'(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})$ ▷ assemble monolithic Jacobian

Find $\mathbf{p}^{(k)}$ by approximately solving $\mathbf{P}_{\text{add/mult}}^{(k)} \mathbf{J}^{(k)} \mathbf{p}^{(k)} = -\mathbf{s}_{\text{add/mult}}^{(k)}$ ▷ use a Krylov method

$$\text{such that } \left\| \mathbf{P}_{\text{add/mult}}^{(k)} \mathbf{J}^{(k)} \mathbf{p}^{(k)} + \mathbf{s}_{\text{add/mult}}^{(k)} \right\| \leq \eta \left\| \mathbf{s}_{\text{add/mult}}^{(k)} \right\|$$

Find $\alpha^{(k)}$ using a backtracking algorithm ▷ use line-search

$[\mathbf{U}^{(k+1)}, \mathbf{C}^{(k+1)}]^\top \leftarrow [\mathbf{U}^{(k)}, \mathbf{C}^{(k)}]^\top + \alpha^{(k)} [\mathbf{p}_u^{(k)}, \mathbf{p}_c^{(k)}]^\top$ ▷ update the iterates

$k \leftarrow k + 1$

approximation of the Jacobian $(\mathcal{F}^{\text{add}})'$ evaluated at iterate k is given as

$$(\mathcal{F}^{\text{add}})'(\mathbf{U}^{(k)}, \mathbf{C}^{(k)}) \approx \underbrace{\begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \\ & \mathbf{J}_{cc}^{(k)} \end{bmatrix}^{-1}}_{=\mathbf{P}_{\text{add}}^{(k)}} \underbrace{\begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \mathbf{J}_{uc}^{(k)} \\ \mathbf{J}_{cu}^{(k)} & \mathbf{J}_{cc}^{(k)} \end{bmatrix}}_{=\mathbf{J}^{(k)}}, \tag{24}$$

where $\mathbf{P}_{\text{add}}^{(k)}$ denotes the additive preconditioner and $\mathbf{J}^{(k)} = F'(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})$. In the multiplicative case, the approximation of the Jacobian $(\mathcal{F}^{\text{mult}})'$ evaluated at iterate k is given as

$$(\mathcal{F}^{\text{mult}})'(\mathbf{U}^{(k)}, \mathbf{C}^{(k)}) \approx \underbrace{\begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \\ & \mathbf{J}_{cc}^{(k)} \end{bmatrix}^{-1}}_{=\mathbf{P}_{\text{mult}}^{(k)}} \underbrace{\begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \mathbf{J}_{uc}^{(k)} \\ \mathbf{J}_{cu}^{(k)} & \mathbf{J}_{cc}^{(k)} \end{bmatrix}}_{=\mathbf{J}^{(k)}}, \tag{25}$$

where $\mathbf{P}_{\text{mult}}^{(k)}$ denotes the multiplicative preconditioner.

Employing (24) or (25), the linear system of equation (23) can be reformulated as follows:

$$\mathbf{P}^{(k)} \mathbf{J}^{(k)} \mathbf{p}^{(k)} = -\mathcal{F}(\mathbf{U}^{(k)}, \mathbf{C}^{(k)}), \tag{26}$$

where $\mathbf{P}^{(k)}$ denotes additive or multiplicative preconditioner matrix. In view of using the inexact-Newton method, the linear system (26) is solved inexactly as discussed in Section 3.

4.2. Algorithmic description of SPIN method

In this section, we briefly discuss the algorithmic components and the practical implementation of the additive and the multiplicative SPIN methods. In general, the SPIN methods, summarized in Algorithm 3, consist of two steps. The first step (highlighted by yellow color) is associated with the construction of the additively/multiplicatively preconditioned nonlinear residual, while the second step (highlighted by gray color) is associated with the solution of a global nonlinearly preconditioned linear system of equations.

4.2.1. Subproblem step — construction of the preconditioned residual

On k th iteration of the SPIN method, the nonlinearly preconditioned residual \mathcal{F} is constructed by solving the subproblems associated with the displacement and the phase-field using the inexact Newton method (Algorithm 2). Thus, we seek the solution \mathbf{U}_{prec} of the displacement subproblems $F_u = 0$ and the solution \mathbf{C}_{prec} of the phase-field subproblem $F_c = 0$. In case of ASPIN method, the subproblem $F_u = 0$ is solved with a fixed value of the phase-field $\mathbf{C}^{(k)}$ and the subproblem $F_c = 0$ is solved with a fixed value of displacement $\mathbf{U}^{(k)}$. The most appealing feature of the ASPIN method is that both subproblems can be solved simultaneously. In the case of the MSPIN method, the nonlinear subproblem $F_u = 0$ is solved first, while the phase-field variable $\mathbf{C}^{(k)}$ is held fixed. Afterward, the subproblem $F_c = 0$ is solved with the fixed value of the displacement \mathbf{U}_{prec} , where \mathbf{U}_{prec} is the solution of the displacement subproblem $F_u = 0$, obtained at the previous step. Here, we highlight the fact that the construction of multiplicatively preconditioned residual coincides with a single iteration of the AM method (Algorithm 1).

After both subproblems are solved, we can use the obtained solutions \mathbf{U}_{prec} and \mathbf{C}_{prec} to construct the nonlinearly preconditioned residual. This is achieved by evaluating the corrections computed during the preconditioning step as

$$\mathbf{s}_u = \mathbf{U}^{(k)} - \mathbf{U}_{\text{prec}}, \quad \mathbf{s}_c = \mathbf{C}^{(k)} - \mathbf{C}_{\text{prec}}. \quad (27)$$

Here, \mathbf{s}_u and \mathbf{s}_c denote the algebraic representation of the preconditioned residuals $\mathcal{F}_u(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})$ and $\mathcal{F}_c(\mathbf{U}^{(k)}, \mathbf{C}^{(k)})$, respectively.

4.2.2. Global step — construction and solution of the preconditioned monolithic linear system

Once the preconditioned residual is formed, we can obtain a search direction $\mathbf{p}^{(k)}$ by solving the linear system of equations $(\mathbf{P}^{(k)} \mathbf{J}^{(k)}) \mathbf{p}^{(k)} = -\mathbf{s}^{(k)}$. To form this linear system, we have to assemble the Jacobian $\mathbf{J}^{(k)}$ associated with the coupled nonlinear residual F and the preconditioning matrix $\mathbf{P}^{(k)}$. Unfortunately, the explicit assembly of the matrix $\mathbf{P}^{(k)}$ is computationally and memory exhaustive, as it requires an inverse of the block-diagonal, or lower-triangular part of the Jacobian $\mathbf{J}^{(k)}$, recall (24) and (25). To overcome the difficulty, we solve the arising linear systems using Krylov subspace methods, which do not require an explicit representation of the preconditioning matrix $\mathbf{P}^{(k)}$. Instead, it is only necessary to evaluate an application of a matrix $\mathbf{P}^{(k)} \mathbf{J}^{(k)}$ to a vector \mathbf{v} , i.e., $\mathbf{y} = \mathbf{P}^{(k)} \mathbf{J}^{(k)} \mathbf{v}$. This evaluation can be carried out sequentially in two steps. Firstly, we multiply the Jacobian matrix $\mathbf{J}^{(k)}$ with a vector \mathbf{v} , i.e., $\mathbf{w} = \mathbf{J}^{(k)} \mathbf{v}$. Secondly, we multiply the preconditioning matrix $\mathbf{P}^{(k)}$ with the vector \mathbf{w} , i.e., $\mathbf{y} = \mathbf{P}^{(k)} \mathbf{w}$. In order to perform this second multiplication efficiently, matrices $\mathbf{P}_{\text{add}}^{(k)}$ and $\mathbf{P}_{\text{mult}}^{(k)}$ defined in (24) and (25) have to be recast to computationally more suitable form. In particular, we reformulate the additive preconditioner $\mathbf{P}_{\text{add}}^{(k)}$ as

$$\mathbf{P}_{\text{add}}^{(k)} := \begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \\ & \mathbf{J}_{cc}^{(k)} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{J}_{uu}^{(k)})^{-1} & \\ & (\mathbf{J}_{cc}^{(k)})^{-1} \end{bmatrix}, \quad (28)$$

while the multiplicative preconditioner $\mathbf{P}_{\text{mult}}^{(k)}$ is reformulated as

$$\mathbf{P}_{\text{mult}}^{(k)} := \begin{bmatrix} \mathbf{J}_{uu}^{(k)} & \\ \mathbf{J}_{cu}^{(k)} & \mathbf{J}_{cc}^{(k)} \end{bmatrix}^{-1} := \begin{bmatrix} \mathbf{I}_{uu} & \\ & (\mathbf{J}_{cc}^{(k)})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{uu} & \\ -\mathbf{J}_{cu}^{(k)} & \mathbf{I}_{cc} \end{bmatrix} \begin{bmatrix} (\mathbf{J}_{uu}^{(k)})^{-1} & \\ & \mathbf{I}_{cc} \end{bmatrix}, \quad (29)$$

where $\mathbf{I}_{uu} \in \mathbb{R}^{dn \times dn}$ and $\mathbf{I}_{cc} \in \mathbb{R}^{n \times n}$ denote identity matrices.

By examining the structure of the matrices $\mathbf{P}_{\text{add}}^{(k)}$ and $\mathbf{P}_{\text{mult}}^{(k)}$, it is now clear that the multiplication of the matrix $\mathbf{P}^{(k)}$ with a vector requires the solution of two linear systems, associated with blocks $\mathbf{J}_{uu}^{(k)}$ and $\mathbf{J}_{cc}^{(k)}$. Here, we solve these linear systems using preconditioned Krylov methods. Since the solution of these linear systems is only used to approximate the matrix–vector product, therefore, it is sufficient to solve these linear systems inexactly. The process of applying the nonlinearly preconditioned operator $\mathbf{P}^{(k)} \mathbf{J}^{(k)}$ to a vector \mathbf{v} is summarized in Algorithms 4 and 5.

From the structure of operators $\mathbf{P}_{\text{add}}^{(k)} \mathbf{J}^{(k)}$ and $\mathbf{P}_{\text{mult}}^{(k)} \mathbf{J}^{(k)}$, we can also see that the additively preconditioned operator $\mathbf{P}_{\text{add}}^{(k)} \mathbf{J}^{(k)}$ is symmetric, while multiplicatively preconditioned operator $\mathbf{P}_{\text{mult}}^{(k)} \mathbf{J}^{(k)}$ is non-symmetric. As a consequence, a different Krylov method is suitable to solve the additively and multiplicatively preconditioned linear system. For instance, the generalized minimal residual (GMRES) method can be employed to efficiently solve the non-symmetric linear systems arising at each MSPIN iteration, while the minimal residual (MINRES) method can be used at each ASPIN iteration. We also point out that the preconditioned Jacobian $\mathbf{P}^{(k)} \mathbf{J}^{(k)}$ has a significantly smaller condition number than the Jacobian of the original system $\mathbf{J}^{(k)}$. This is due to the fact that the SPIN preconditioner implicitly provides robust linear preconditioning by means of the preconditioning matrix $\mathbf{P}^{(k)}$. As a consequence, solving (26) typically requires only very few iterations of a Krylov method.

Algorithm 4: Evaluation of $\mathbf{P}_{\text{add}}^{(k)} \mathbf{J}^{(k)} \mathbf{v}$ **Data:** $\mathbf{J}^{(k)} \in \mathbb{R}^{(d+1)n \times (d+1)n}$, $\mathbf{v}_u \in \mathbb{R}^{dn}$, $\mathbf{v}_c \in \mathbb{R}^n$ **Result:** $\mathbf{y}_u, \mathbf{y}_c$

$[\mathbf{w}_u, \mathbf{w}_c]^\top \leftarrow \mathbf{J}^{(k)}[\mathbf{v}_u, \mathbf{v}_c]^\top$ ▷ apply Jacobian to vector
 Obtain $\mathbf{J}_{uu}^{(k)}, \mathbf{J}_{cc}^{(k)}$ by decomposing $\mathbf{J}^{(k)}$ ▷ get block components of Jacobian
 Find \mathbf{y}_u by approximately solving $\mathbf{J}_{uu}^{(k)} \mathbf{y}_u = \mathbf{w}_u$ ▷ use a Krylov method
 such that $\|\mathbf{J}_{uu}^{(k)} \mathbf{y}_u - \mathbf{w}_u\| \leq \epsilon_{\text{app_lin}} \|\mathbf{w}_u\|$
 Find \mathbf{y}_c by approximately solving $\mathbf{J}_{cc}^{(k)} \mathbf{y}_c = \mathbf{w}_c$ ▷ use a Krylov method
 such that $\|\mathbf{J}_{cc}^{(k)} \mathbf{y}_c - \mathbf{w}_c\| \leq \epsilon_{\text{app_lin}} \|\mathbf{w}_c\|$

Algorithm 5: Evaluation of $\mathbf{P}_{\text{mult}}^{(k)} \mathbf{J}^{(k)} \mathbf{v}$ **Data:** $\mathbf{J}^{(k)} \in \mathbb{R}^{(d+1)n \times (d+1)n}$, $\mathbf{v}_u \in \mathbb{R}^{dn}$, $\mathbf{v}_c \in \mathbb{R}^n$ **Result:** $\mathbf{y}_u, \mathbf{y}_c$

$[\mathbf{w}_u, \mathbf{w}_c]^\top \leftarrow \mathbf{J}^{(k)}[\mathbf{v}_u, \mathbf{v}_c]^\top$ ▷ apply Jacobian to vector
 Obtain $\mathbf{J}_{uu}^{(k)}, \mathbf{J}_{cc}^{(k)}, \mathbf{J}_{cu}^{(k)}$ by decomposing $\mathbf{J}^{(k)}$ ▷ get block components of Jacobian
 Find \mathbf{y}_u by approximately solving $\mathbf{J}_{uu}^{(k)} \mathbf{y}_u = \mathbf{w}_u$ ▷ use a Krylov method
 such that $\|\mathbf{J}_{uu}^{(k)} \mathbf{y}_u - \mathbf{w}_u\| \leq \epsilon_{\text{app_lin}} \|\mathbf{w}_u\|$
 $\mathbf{z}_c \leftarrow \mathbf{w}_c - \mathbf{J}_{cu}^{(k)} \mathbf{y}_u$ ▷ apply off-diagonal block
 Find \mathbf{y}_c by approximately solving $\mathbf{J}_{cc}^{(k)} \mathbf{y}_c = \mathbf{z}_c$ ▷ use a Krylov method
 such that $\|\mathbf{J}_{cc}^{(k)} \mathbf{y}_c - \mathbf{z}_c\| \leq \epsilon_{\text{app_lin}} \|\mathbf{z}_c\|$

After the search direction $\mathbf{p}^{(k)}$ is obtained by solving the nonlinearly preconditioned system of equations (26), the line-search method is used to determine the appropriate step size $\alpha^{(k)}$. This is followed by the update of variables $\mathbf{U}^{(k)}$ and $\mathbf{C}^{(k)}$, performed as $\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \alpha^{(k)} \mathbf{p}_u^{(k)}$ and $\mathbf{C}^{(k+1)} = \mathbf{C}^{(k)} + \alpha^{(k)} \mathbf{p}_c^{(k)}$, where $\mathbf{p}_u^{(k)}$ and $\mathbf{p}_c^{(k)}$ denote the components of $\mathbf{p}^{(k)}$ associated with the displacement and the phase-field, respectively.

Compared to the AM method, one iteration of the SPIN algorithm is computationally more expensive. The additional computational cost is associated with the global step of the SPIN algorithm. More precisely, we have to assemble monolithic Jacobian $\mathbf{J}^{(k)}$, which can be performed efficiently by reusing the block \mathbf{J}_{uu} constructed during the first iteration of Newton's method, while solving the subproblem $F_u = 0$. In the case of the ASPIN method, block \mathbf{J}_{cc} can be also reused. In contrast, for the MSPIN method, the block \mathbf{J}_{cc} has to be re-evaluated at $\mathbf{U}^{(k)}$ and $\mathbf{C}^{(k)}$. Moreover, the off-diagonal coupling blocks \mathbf{J}_{uc} and \mathbf{J}_{cu} have to be assembled for both SPIN methods. Besides the assembly cost, the preconditioned coupled system of equations has to be solved using the Krylov method, which requires the solution of two linear systems, associated with \mathbf{J}_{uu} and \mathbf{J}_{cc} , at each iteration. Even though one iteration of the SPIN method is more expensive than an iteration of the AM method, its superior convergence properties outweigh the additional computational cost. Hence, the overall simulation time of the phase-field fracture problems is greatly reduced, as shown in Section 6.

5. Benchmark problems

In this section, we present five benchmark problems, which we use to demonstrate the convergence properties and the performance of the proposed SPIN methods. For all test problems, we prepare a finite element mesh by locally refining the expected crack path. The length scale parameter l_s is then prescribed as $l_s = 2h$, where h denotes the size of the largest element from the locally pre-refined zone. Table 1 contains a summary of the material parameters, the tolerance used for enforcing the irreversibility condition, and the number of degrees of freedom (dofs) for all numerical examples.

All considered benchmark problems are solved using both, additive and multiplicative, SPIN methods as well as the AM method. During all numerical experiments, the SPIN methods solve the phase-field and displacement subproblems using the INK method with the biconjugate gradient stabilized (BiCGSTAB) method preconditioned

Table 1

Values of material parameters and the number of dofs for the presented numerical experiments.

Parameters	Test problems				
	Tension	Shear	Three-point bending	L-shaped	Asymmetrically notched beam
\mathcal{G}_c (kN/mm)	$2.7 \cdot 10^{-3}$	$2.7 \cdot 10^{-3}$	$5.4 \cdot 10^{-4}$	$8.9 \cdot 10^{-5}$	10^{-3}
λ (kN/mm ²)	121.15	121.15	12.00	6.16	12.00
μ (kN/mm ²)	80.77	80.77	8.00	10.95	8.00
l_s	0.003	0.006	0.01	2.0	0.06
τ_{irr}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
γ	$9 \cdot 10^3$	$4.5 \cdot 10^3$	540	0.45	166.67
# dofs	180,117	92,004	99,135	141,849	139,020

with the AMG method, as described in Section 3. The solution of the global preconditioned linear system is obtained using the GMRES method for both SPIN methods. As discussed in Section 4.2, applying the SPIN preconditioner requires a solution of two linear systems. Since these linear systems have the same structure as the linear systems arising while solving the subproblems, we also solve them using the BiCGSTAB method, preconditioned with the AMG method.

In this work, we consider four variants of AM method, which employ different subproblem solvers and stopping criteria. More precisely, the AM-ND and AM-ST methods make use of Newton's method with a direct linear solver. The AM-NK employs Newton's method with the BiCGSTAB linear solver, which is preconditioned with the AMG method. We also consider the AM-INK method, which utilizes the inexact Newton method with the AMG preconditioned BiCGSTAB method.

Table 2 provides a summary of all tolerances used for the termination of all employed solution strategies. We point out that the AM-NK, AM-INK, and AM-ND methods use the same stopping criteria as the SPIN methods, in order to ensure a fair comparison. In contrast, AM-ST method employs a termination criterium commonly used in the phase-field fracture literature [4], i.e., the method terminates as soon as the change in the phase-field drops below a tolerance $\epsilon_{c,diff}$, i.e., $\|\mathbf{C}^{(k+1)} - \mathbf{C}^{(k)}\|_\infty \leq \epsilon_{c,diff}$.

Remark 2. The stopping criteria used in this work employ standard Euclidean norms and therefore they are mesh-dependent. Alternatively, the mesh-independent stopping criteria could be employed, for example by using the correction in the energy norm [64] given as $\|[\mathbf{U}^{(k+1)}, \mathbf{C}^{(k+1)}]^\top - [\mathbf{U}^{(k)}, \mathbf{C}^{(k)}]^\top\|_{F'} \leq \epsilon$, or the relative correction in the L^2 -norm [38] given as $\max\left(\frac{\|\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}\|_{L^2}}{\|\mathbf{U}^{(k+1)}\|_{L^2}}, \frac{\|\mathbf{C}^{(k+1)} - \mathbf{C}^{(k)}\|_{L^2}}{\|\mathbf{C}^{(k+1)}\|_{L^2}}\right) \leq \epsilon$.

The convergence study and exact configuration of all solution strategies are discussed in detail in Section 6. Here, we only report simulation results, in terms of obtained crack patterns. If all solution strategies converge to the identical solution, only a single snapshot of the result is reported. Moreover, the evolution of elastic and fracture energy is monitored for all simulations in order to assess the quality of all employed solution strategies. To increase the readability of our results, we do not report the energy evolution for AM-NK/INK methods, as their global convergence behavior and stopping criterium is identical to those of the AM-ND method.

5.1. Single-edge notched plate under tension

As the first benchmark problem, we investigate the single-edge notched plate under uniaxial tension. We consider plate of size 1×1 mm with 0.5 mm initial notch, subjected to boundary conditions as depicted in Fig. 2 on the left. In particular, we impose Dirichlet boundary conditions $(\mathbf{u}_D^t)_y := t\bar{u}$, where $\bar{u} = 1$ mm/s on the top of the boundary. During this experiment, the pseudo time-step is chosen as $\delta t = 5 \cdot 10^{-5}$. For this particular boundary value problem setup, the initial crack is known to extend across the plate within a single loading step. Therefore, the experiment is widely employed in the literature [18,21,32] to investigate the robustness of the solution strategies with respect to the ‘‘brutal’’ crack propagation.

Fig. 2 in the middle depicts the simulation result, while Fig. 2 on the right illustrates the evolution of elastic and fracture energy as a function of time. As we can see, the fracture energy remains zero and elastic energy is

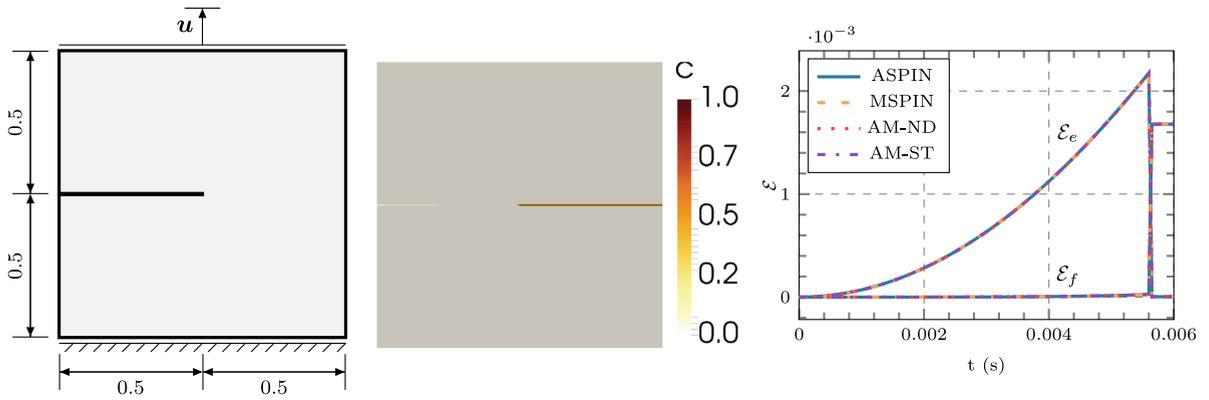


Fig. 2. Tension test. Left: Geometry (units are in mm) and boundary value problem setup. Middle: Simulation result. Right: Evolution of elastic (\mathcal{E}_e) and fracture (\mathcal{E}_f) energy over (pseudo-)time for ASPIN, MSPIN, AM-ND and AM-ST solution strategies.

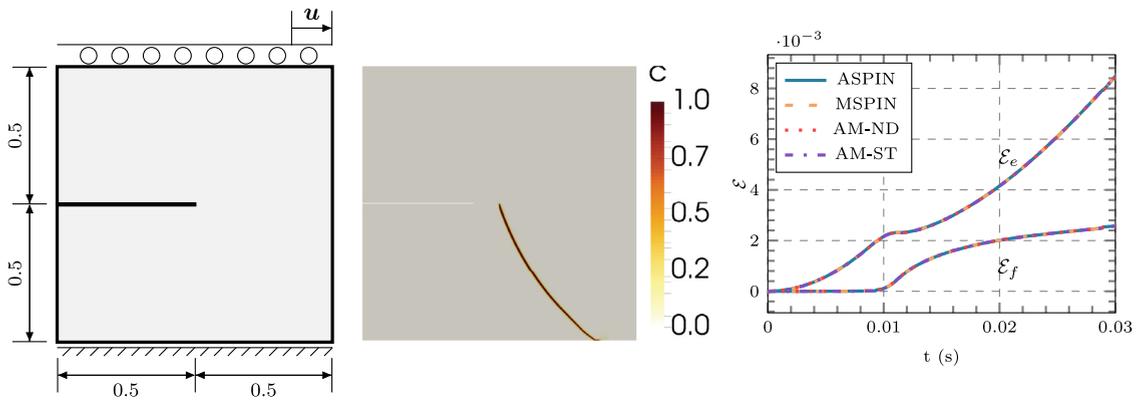


Fig. 3. Shear test. Left: Geometry (units are in mm) and boundary value problem setup. Middle: Simulation result. Right: Evolution of elastic (\mathcal{E}_e) and fracture (\mathcal{E}_f) energy over (pseudo-)time for ASPIN, MSPIN, AM-ND and AM-ST solution strategies.

increasing until the critical time $t_c = 5.65 \cdot 10^{-3}$ second is reached. Once t_c is reached, the “brutal” crack propagation takes place. The formation of the crack causes an increase in fracture energy and the degradation in elastic energy. As we can also observe from Fig. 2, the evolution of the elastic and fracture energy is comparable for all four solution strategies.

5.2. Single-edge notched plate under shear

As our second test, we investigate the single-edge notched plate under shear loading. Here, we consider the same domain as for the tension test, but different boundary conditions are imposed. In particular, the bottom of the specimen is held fixed, while the incremental horizontal displacement is prescribed, see also Fig. 3 on the left. During this experiment, we prescribe the Dirichlet boundary conditions $(u^t_D)_x := t\bar{u}$, where $\bar{u} = 1$ mm/s, on the top of the boundary. Moreover, we employ $\delta t = 10^{-3}$ for the first 8 steps and $\delta t = 7.5 \cdot 10^{-5}$ subsequently until the complete failure occurs.

This example is often considered a canonical one in the phase-field fracture literature [4,5,18,54], as it enables to test the capability of the model to ensure the asymmetric nucleation in tension and compression. As described in Section 2, we employ the phase-field model with the spectral decomposition of the strain tensor. In this case, the initial crack is extended only in the bottom-right part of the specimen and forms a curved crack, see Fig. 3 in the

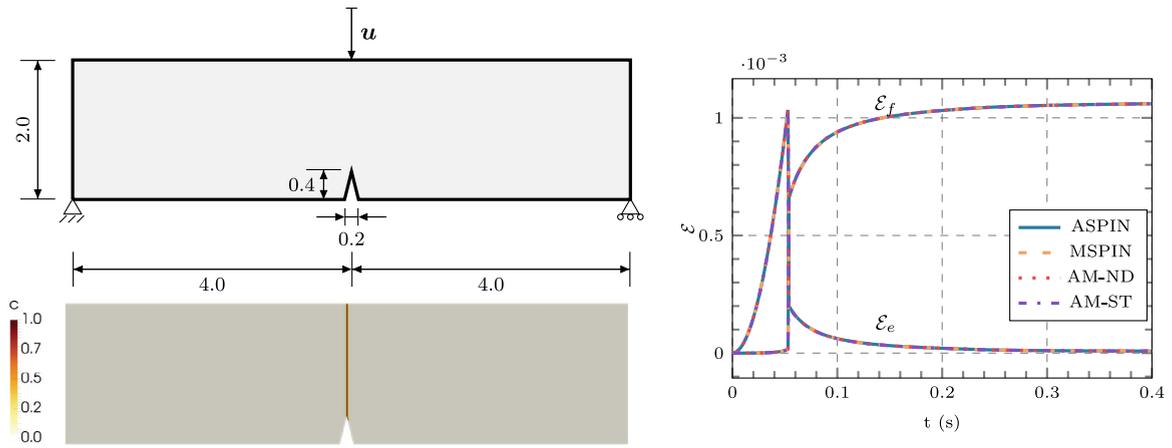


Fig. 4. Three-point bending test. Left top: Geometry (units are in mm) and boundary value problem setup. Left bottom: Simulation result. Right: Evolution of elastic (\mathcal{E}_e) and fracture (\mathcal{E}_f) energy over (pseudo-)time for ASPIN, MSPIN, AM-ND and AM-ST solution strategies.

middle. In contrast to the tension test, the crack propagation is stable and occurs consistently over multiple loading steps. This enables us to assess the convergence properties and the efficiency of the solution strategies over time.

Fig. 3 on the right depicts the evolution of the elastic and fracture energies as a function of time. As we can see, the behavior of the elastic and fracture energies coincides for all three solution strategies. We can also observe that complete failure is never attained. After the cracked specimen reaches a stage, where no further evolution of the phase-field in the lower-right corner is possible, the behavior corresponds to the linearly elastic response of the cracked specimen clamped at the undamaged bottom-right portion of the boundary. Comparable numerical results have also been reported in [5,65].

5.3. Three-point bending test

In this experiment, we model the three-point bending of a simply supported beam, as presented in [6,65]. Fig. 4 on the left illustrates the geometry and the boundary value problem setup. The time-dependent boundary conditions are prescribed on the top as $(\mathbf{u}'_D)_y := t\bar{u}$, where $\bar{u} = -1$ mm/s. During this experiment, we use constant pseudo time-step $\delta t = 10^{-3}$. For this particular loading scenario, the crack propagates from the notch in a straight line towards the upper boundary, see also Fig. 4 on the left. The crack propagation exhibits two phases, which can be observed from the curves illustrating the energy evolution over time (Fig. 4 on the right). The first phase consists of the crack propagation in the tension mode, thus we can observe “brutal” crack propagation at time $t = 0.058$ seconds. This is followed by the second phase, where the crack propagates gradually. As for the previous examples, the same energy evolution is obtained with all three solution strategies under consideration.

Since this experiment contains both “brutal” and gradual crack propagation, we also decided to use it in order to assess our implementation of the phase-field fracture model. To this aim, we study the converge behavior of the phase-field fracture model under refinement. We consider a hierarchy of three meshes, obtained by uniformly refining the adaptively refined finite element mesh considered in the experiments above. With each refinement step, the mesh size decreases by a factor of two, which allows us to also reduce the associated value of the length-scale parameter l_s . We can observe from Fig. 15 on the left, that the mesh resolution and the value of length-scale parameter influence the time of crack propagation, as well as the values of the elastic and fracture energy. As expected from the Γ -convergence theory [1], as the value of the length-scale parameter l_s approaches zero, the fracture energy converges towards $\mathcal{E}_f = \mathcal{G}_c W$, where W symbolizes the area of the fracture surface.

5.4. L-shaped panel test

Our next experiment considers crack propagation in an L-shaped panel. Here, we adopt the geometry and material parameters from [65], which mimics the experimental setup presented in [66]. The details about geometry and the

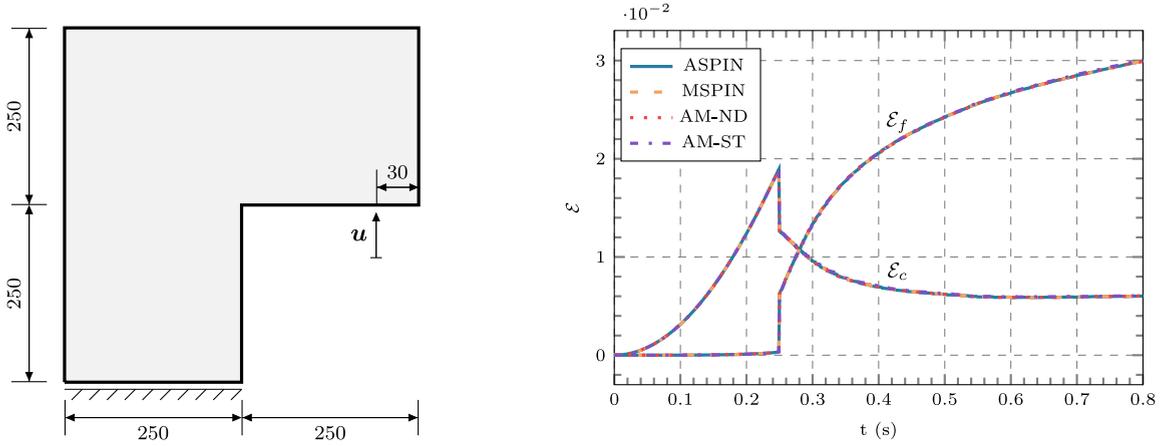


Fig. 5. The L-shaped panel test. Left: Geometry (units are in mm) and boundary value problem setup. Right: Evolution of elastic (\mathcal{E}_e) and fracture (\mathcal{E}_f) energy over (pseudo)-time for ASPIN, MSPIN, AM-ND and AM-ST solution strategies.

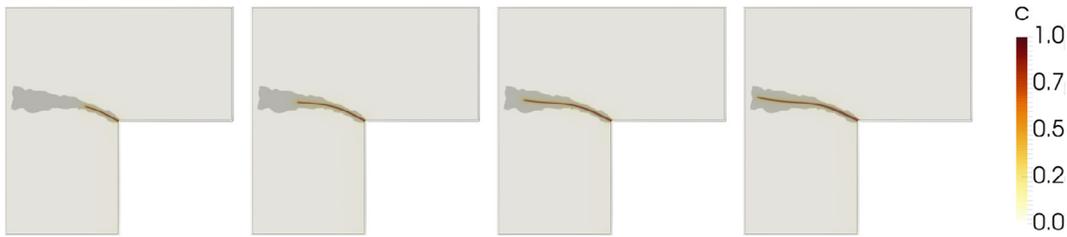


Fig. 6. Simulation result for L-shaped panel test. The simulation result (the value of the phase-field) overlays the experimentally obtained results reported in [66]. The crack evolution depicted at time $t \in \{0.27, 0.365, 0.515, 0.792\}$ (s), from left to the right.

problem setup can be found in Fig. 5 on the left. The simulation is performed by imposing the Dirichlet boundary conditions of the following form: $(\mathbf{u}'_D)_y := t\bar{u}$, where $\bar{u} = 1$ mm/s. The pseudo time-step δt is set to 10^{-2} for the first 20 steps and to 10^{-3} afterwards. Fig. 6 presents the obtained results, i.e., the simulated crack path at different loading steps. As we can see, the simulated crack path is in agreement with the experimental results reported in [66]. Furthermore, Fig. 5 on the right demonstrates that the identical evolution of the energies is obtained irrespectively of the choice of the solution strategy at the hand.

5.5. Asymmetrically notched beam test

In the last benchmark problem, we consider an asymmetrically notched beam, with the notch and three holes in the left part, as illustrated in Fig. 7. The beam is subjected to displacement loading in the center of the top boundary while being supported at two points at the bottom boundary. Thus, we prescribe the Dirichlet boundary conditions as $(\mathbf{u}'_D)_y := t\bar{u}$, where $\bar{u} = -1$ mm/s. The time-step is chosen as $\delta t = 10^{-3}$ for the first 160 steps and as $\delta t = 10^{-4}$ for the remaining part of the simulation. The material parameters are selected as in [36,37,67] in order to mimic the experimental setup from [68].

For a given setup, the experimental results have shown that crack should propagate towards the second hole, which we can also observe in our numerical experiments, as depicted in Fig. 8. However, we obtain three distinct numerical solutions. This is not surprising, as the energy functional is non-convex and therefore the minimization admits multiple solutions. Interestingly, all variants of the AM method converge to the same solution, while the ASPIN and MSPIN method find two different crack patterns. Close examination of the energies suggests that the MSPIN method converges to the solution with the smallest value of the total potential energy \mathcal{E} . In particular, the value of the total potential energy is $\mathcal{E} \approx 1.379 \cdot 10^{-2}$ for the MSPIN method, $\mathcal{E} \approx 1.402 \cdot 10^{-2}$ for the AM methods and, $\mathcal{E} \approx 1.435 \cdot 10^{-2}$ for the ASPIN method. However, we also point out that additional solutions might exist.

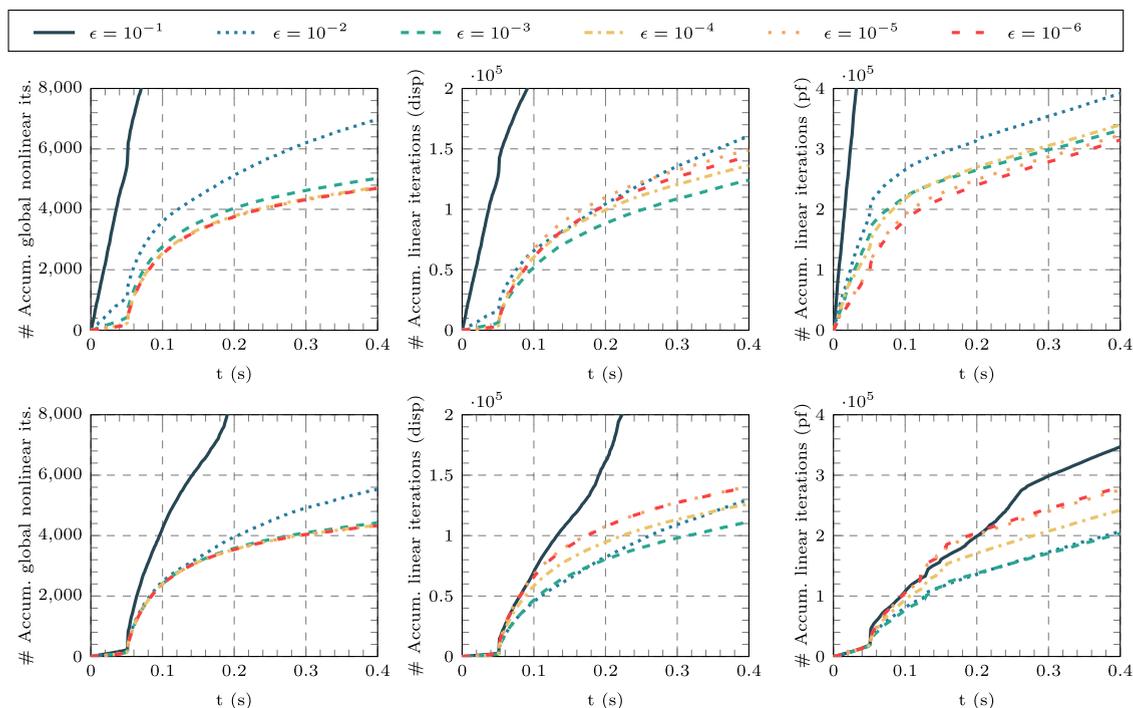


Fig. 9. Three-point bending test. The convergence behavior of the ASPIN (top row) and MSPIN (bottom row) methods with respect to different values of ϵ_{app_lin} . Left: Accumulated number of global nonlinear iterations. Middle: The accumulated number of linear iterations required for solving the linear systems associated with the displacement field. Right: The accumulated number of linear iterations required for solving the linear systems associated with the phase-field.

6.2. Sensitivity of the SPIN methods to choice of ϵ_{app_lin}

As discussed in Section 4.2, applying the SPIN preconditioners requires a solution of two linear systems. However, it is not necessary to solve these linear systems exactly, but only with some relative tolerance ϵ_{app_lin} . Fig. 9 demonstrates the convergence behavior of SPIN methods with respect to different values of ϵ_{app_lin} . For this study, the experiments are performed using the three-point bending test with 6417 dofs, where the mesh size is approximately 4 times larger than the mesh size reported in Table 2. We also remark that the length-scale parameter is adjusted accordingly for the given mesh size. As we can observe from the obtained results, it is necessary to choose $\epsilon_{app_lin} \leq 10^{-2}$, as larger values of ϵ_{app_lin} can cause the stagnation or even divergence of SPIN method. We can also see that for $\epsilon_{app_lin} \leq 10^{-3}$, the number of required global ASPIN/MSPIN iterations remains constant. However, as the tolerance ϵ_{app_lin} decreases, the number of required linear iterations varies, which causes an increase in the computational cost of the overall method. Identical behavior can be also observed in terms of execution time, reported in Table 3. As we can see, the MSPIN method configured with $\epsilon_{app_lin} = 10^{-4}$ is the fastest. While for the ASPIN method the fastest performance can be observed with $\epsilon_{app_lin} = 10^{-5}$. As a compromise, all numerical results presented in this work employ $\epsilon_{app_lin} = 10^{-4}$.

6.3. Convergence properties of solution strategies

In this section, we study the convergence and the performance of the AM and SPIN methods for all benchmark problems. Fig. 10 depicts the number of iterations required for reaching the desirable stopping criterion at each loading step. In particular, we plot a number of global nonlinear iterations, the total number of linear iterations required for solving linear systems associated with displacement and phase-field subproblems, at each loading step. Furthermore, Table 4 reports an accumulated number of nonlinear and linear iterations required for solving arising coupled problems, as well as the displacement and phase-field subproblems. Here, we point out that the

Table 2

Summary of the numerical tolerances used for the termination of all solution strategies.

Symbol	Description of the termination criteria	Value
$\epsilon_{\text{abs_glob_nonl}}$	Absolute residual norm of global algorithms (AM/ASPIN/MSPIN).	10^{-7}
$\epsilon_{\text{rel_glob_nonl}}$	Relative residual norm of global algorithms (AM/ASPIN/MSPIN).	10^{-6}
$\epsilon_{\text{abs_sub_nonl}}$	Absolute residual norm of ND/NK/INK methods, used for solving subproblems.	10^{-7}
$\epsilon_{\text{rel_sub_nonl}}$	Relative residual norm of ND/NK/INK methods, used for solving subproblems.	10^{-6}
$\epsilon_{\text{abs_lin}}$	Absolute residual norm of Krylov methods, used inside of NK method.	10^{-9}
$\epsilon_{\text{rel_lin}}$	Relative residual norm of Krylov methods, used inside of NK method.	10^{-9}
$\epsilon_{\text{app_lin}}$	Relative residual norm of Krylov methods, used for applying SPIN preconditioner.	10^{-4}
$\epsilon_{\text{c_diff}}$	Absolute difference in change of phase-field measured in infinity norm.	10^{-4}

Table 3The execution time (mins) of the ASPIN and MSPIN method required for the simulation of the three-point bending test. The time is reported for different values of $\epsilon_{\text{app_lin}}$.

$\epsilon_{\text{app_lin}}$	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
ASPIN	206.05	24.26	16.13	15.02	14.74	15.80
MSPIN	73.79	18.65	15.60	14.26	14.98	14.78

AM method requires a solution of linear systems only during the local (subproblem) phase. In contrast, the SPIN methods require a solution of the preconditioned coupled linear system at each global nonlinear iteration. This is achieved using Krylov methods, which rely on an efficient evaluation of the matrix–vector product $(\mathbf{P}\mathbf{J})\mathbf{v}$, which in turn necessitates approximation of \mathbf{J}_{uu}^{-1} and \mathbf{J}_{cc}^{-1} , recall Algorithms 4 and 5. As specified in Section 4.2, we approximate both inverses by solving associated linear systems using the BiCGSTAB method preconditioned with the AMG method. Therefore, the reported number of linear iterations for displacement and phase-field subproblems also includes the iterations required while solving the preconditioned coupled system.

As we can see from Fig. 10, the number of linear and nonlinear iterations increases rapidly as soon as the crack propagation starts. For the benchmark problems, which exhibit only the “brutal” crack-propagation, i.e., tension and asymmetrically notched beam, a rapid increase in iteration count typically occurs only once per simulation, at the critical time t_c . By taking a closer look at the number of required global nonlinear iterations, we can see that all variants of the AM method except the AM-ST method exhibit almost identical convergence behavior. This is due to the fact that the AM-ST method satisfies its stopping criterion earlier than the other three AM methods. Thus, the choice of linear solvers does not impact the overall convergence speed of the AM method, only its performance in terms of the execution time, as we see in Section 6.4. We also point out that the SPIN methods require a lower number of nonlinear global iterations than all variants of AM method for almost all benchmark problems. The difference is more prevalent for the problems with the gradual crack propagation (Shear, Three-point bending, L-shaped panel), as the obtained savings accumulate over several loading steps. For instance, for the shear test, an accumulated number of global nonlinear iterations are around 40,000 and 82,000 for AM-ST and AM-INK/NK/ND methods, respectively. In contrast, the SPIN methods require only around 20,000 global nonlinear iterations, see Table 4.

The displacement and phase-field subproblems are solved once per each global nonlinear iteration. As a consequence, the accumulated number of nonlinear iterations required to solve both subproblems is significantly higher for AM methods than for SPIN solution strategies. For instance, for the L-shaped panel benchmark problem, the number of accumulated nonlinear iterations required for solving all arising displacement subproblems is approximately 170,000 for AM-ND/NK/INK methods, 80,000 for the AM-ST methods while SPIN methods require approximately 70,000 iterations. If we examine the iteration count more closely, we also notice that the benchmark problems with gradual crack propagation require fewer nonlinear subproblem iterations per global step than benchmark problems that exhibit brutal crack propagation. For example, the AM-ND method requires on average approximately 1.11 nonlinear iterations for displacement subproblem per global step for the shear test, while approximately 2.16 iterations are needed for the tension test.

SPIN methods require a solution of preconditioned coupled linear system (26) at each global nonlinear iteration. Results reported in Table 4 suggest that solving these preconditioned systems is fairly cheap as the Krylov methods

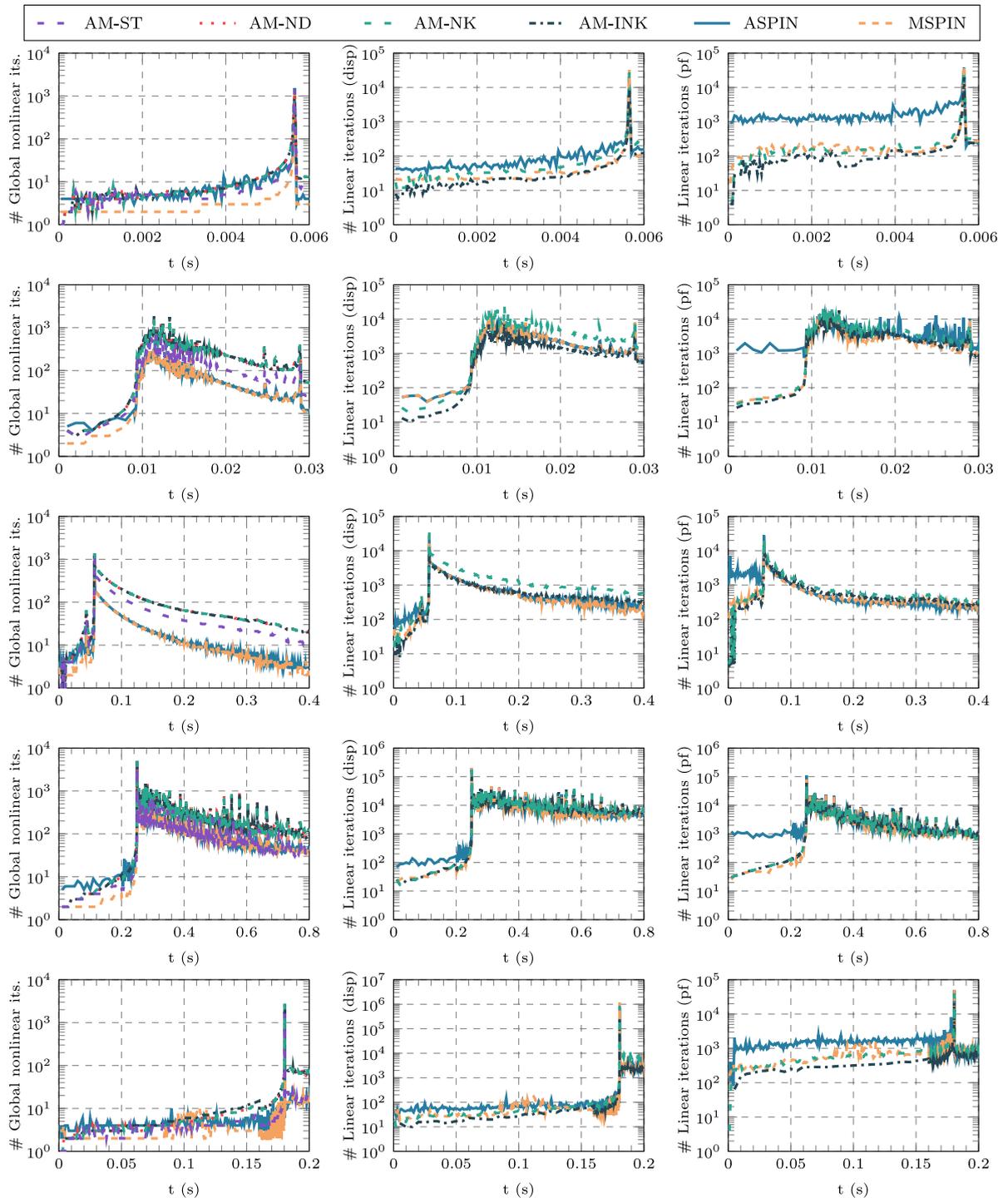


Fig. 10. Convergence history of ASPIN, MSPIN, AM-ST, AM-ND, AM-NK, and AM-INK solution strategies. The experiments are performed for tension, shear, three-point bending, L-shaped panel, and asymmetrically notched beam tests (from top to bottom). Left: The number of global nonlinear iterations over time. Middle: The number of linear iterations performed for solving linear systems related to the displacement subproblem. Right: The number of linear iterations performed for solving linear systems related to the phase-field subproblem.

Table 4

Summary of computational cost for ASPIN, MSPIN, AM-ST, AM-ND, AM-NK, and AM-INK methods. Summary information is expressed in terms of the accumulated number of linear and nonlinear iterations required to solve the coupled problem as well as the internal displacement and phase-field subproblems. The symbols *, †, + denote convergence to a different solution.

Example	Solver	Global		Disp. subproblem		PF subproblem	
		Nonlinear	Linear	Nonlinear	Linear	Nonlinear	Linear
Tension	AM-ND	2,507	–	5,421	–	4,422	–
	AM-NK	2,507	–	5,421	34,670	4,422	57,516
	AM-INK	2,526	–	6,373	12,336	4,422	34,330
	AM-ST	2,135	–	5,052	–	4,033	–
	ASPIN	2,375	4,925	5,490	31,580	11,322	222,009
	MSPIN	1,823	5,336	5,125	37,034	6,328	63,237
Shear	AM-ND	81,940	–	91,066	–	86,908	–
	AM-NK	81,940	–	91,066	1,437,712	86,909	1,430,913
	AM-INK	82,106	–	115,785	459,834	87,088	980,977
	AM-ST	40,626	–	50,023	–	45,710	–
	ASPIN	20,250	59,584	26,237	711,607	45,312	1,212,635
	MSPIN	19,824	58,766	25,318	697,591	32,732	849,583
Three-point bending	AM-ND	36,037	–	42,249	–	39,232	–
	AM-NK	36,037	–	42,250	600,923	39,232	380,221
	AM-INK	36,045	–	42,622	290,456	39,243	291,308
	AM-ST	22,407	–	28,716	–	25,639	–
	ASPIN	8,875	26,021	12,968	304,249	22,096	362,201
	MSPIN	8,409	25,452	12,168	287,991	17,299	244,672
L-shaped panel	AM-ND	152,121	–	168,475	–	166,997	–
	AM-NK	152,121	–	168,475	5,549,803	166,998	1,774,117
	AM-INK	158,205	–	175,214	1,989,176	167,005	948,725
	AM-ST	64,833	–	81,656	–	79,357	–
	ASPIN	67,653	201,748	71,651	4,433,556	111,782	1,981,303
	MSPIN	66,326	198,405	70,099	4,370,314	97,412	1,719,563
Asymmetrically notched beam	AM-ND*	21,961	–	29,959	–	29,989	–
	AM-NK*	21,961	–	29,515	2,262,266	29,958	559,184
	AM-INK*	21,376	–	29,543	707,202	28,725	287,935
	AM-ST*	7,200	–	13,957	–	15,391	–
	ASPIN†	6,868	21,918	37,882	1,343,071	36,408	784,603
	MSPIN+	5,978	22,367	48,558	1,710,703	27,193	520,496

require only a few iterations to converge. In particular, for the tension test, the GMRES method requires on average approximately 2.07 and 2.92 iterations to solve the preconditioned coupled linear system, when invoked by the ASPIN method and the MSPIN method, respectively.

In the end, we note that the number of linear iterations for both, displacement and phase-field, subproblems mimics the trend observed for the global nonlinear iterations. Thus, AM-NK and AM-INK methods require a substantially higher amount of linear iterations than SPIN methods. For instance, for the three-point bending test, AM-NK requires approximately 1.55 times more linear phase-field subproblem iterations and 2.09 times more linear displacement subproblem iterations than the MSPIN method. Here, we also point out that the AM-INK method requires fewer iterations than the AM-NK method, due to the use of the adaptive stopping criterion induced by the inexact Newton method, recall Section 3.1.3

6.3.1. Analysis of convergence history

In this section, we analyze the convergence behavior of the AM-ST and SPIN methods by examining the norm of residual and the value of energy during the solution process. For the tension test, three-point bending test, and asymmetrically notched beam test, we focus our attention on the loading steps where the “brutal” crack propagation occurs. For the shear test, the crack propagation takes place in a gradual manner, so we pick a random loading step. In the case of the L-shaped panel test, a crack is first propagated brutally, which is followed by gradual crack propagation. Therefore, we consider the loading step where the crack propagation initiates.

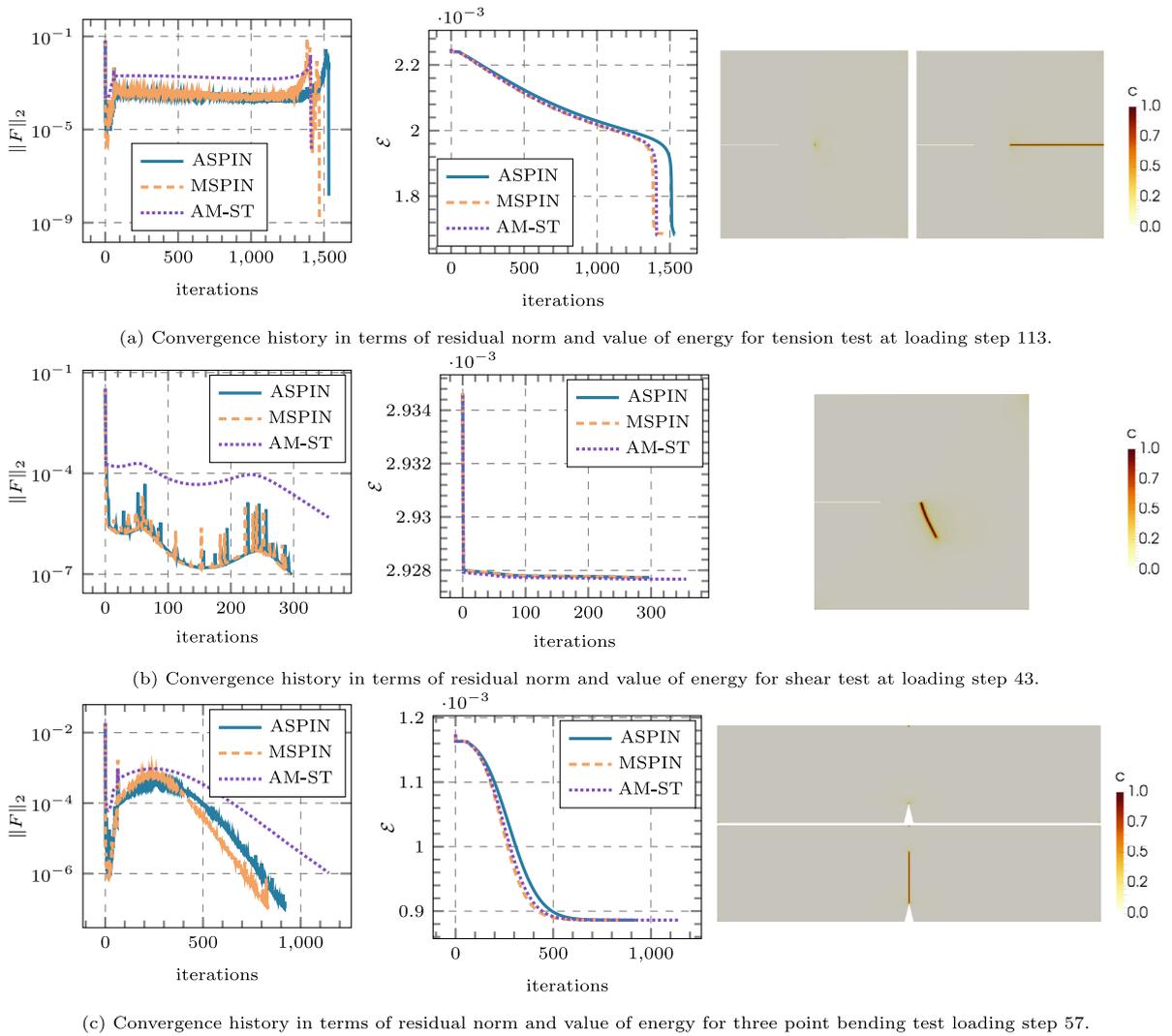


Fig. 11. Convergence history and snapshots of crack evolution for a given loading step.

Fig. 11 illustrates the norm of residual and energy over iterations as well as the crack pattern before and after the specific time step for the tension test, the shear test, and the three-point bending tests. Fig. 12 depicts the same quantities for the L-shaped panel test and the asymmetrically notched beam tests. From Figs. 11 and 12, we observe that the AM-ST method terminates sooner than SPIN methods. We can notice that the energy is monotonically decreasing as we have employed the backtracking line-search method. In contrast, the norm of residual is allowed to oscillate. For the considered test cases, we see that the residual is oscillating more for the SPIN methods than for the AM-ST method.

For the tension test (Fig. 11(a)), we see that the norm of residual reduces rapidly in the first few iterations, but then it increases and oscillates before actually achieving the quadratic convergence after more than 1000 iterations. In the last few iterations, we also observe that the energy reduces dramatically. This is quite contrary to the shear test (Fig. 11(b)), where the energy reduces rapidly only in the first iteration. The three-point bending test also shows a different behavior (Fig. 11(c)) as the energy is reduced in the first 500 iterations and then a flat region is encountered. Although, we notice that the norm of residual is reducing at a higher rate.

The norm of residual is more oscillatory for the L-shaped panel test than for other benchmark problems, as can be seen from Fig. 12(a). Here, we notice that the AM-ST method requires fewer iterations to converge than the

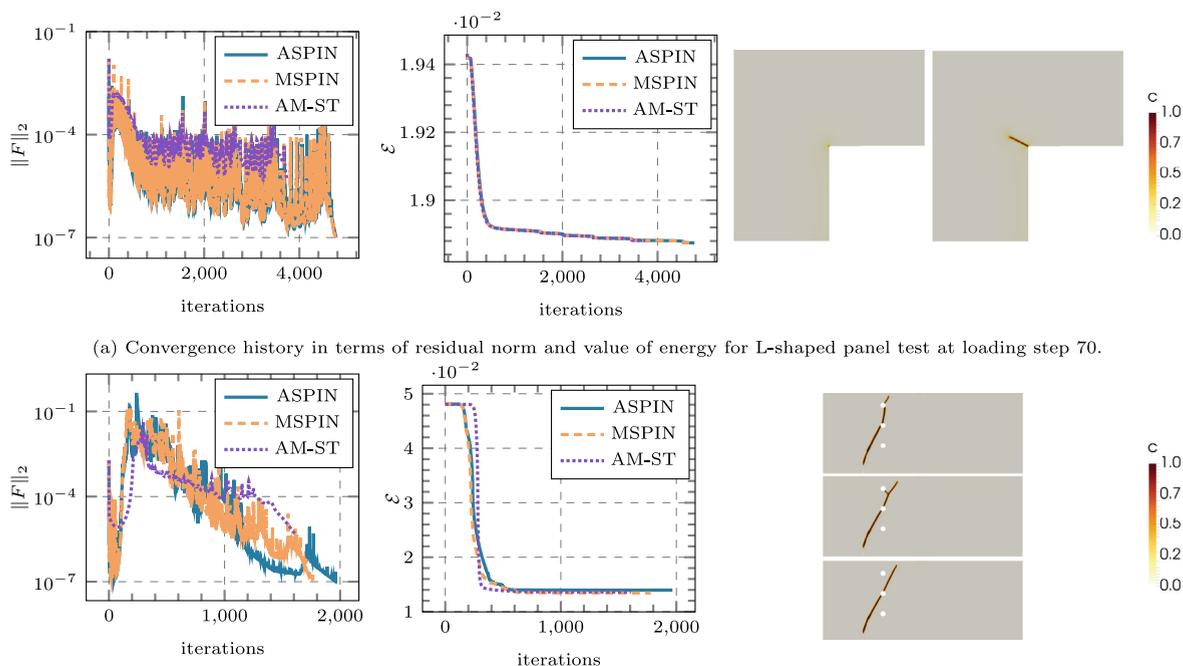


Fig. 12. Convergence history and snapshots of crack evolution for a given loading step.

SPIN method as it employs weaker termination criteria. More interesting results can be observed in Fig. 12(b), as the obtained crack pattern for the asymmetrically notched beam differs for different solution strategies. For this test case, we can see that the MSPIN method achieves a solution with the smallest value of energy in comparison with the other methods.

6.4. Execution time

In this section, we compare the execution time required by the AM and SPIN methods for all benchmark problems. The reported execution times stand for the total execution time necessary to run the entire simulation, which includes, call to nonlinear assembly routines, setup and execution of nonlinear and linear solvers, I/O, and post-processing. The obtained results, summarized in Table 5, demonstrate that the SPIN methods are faster than the AM methods for all benchmark problems. One exception in these tests is the asymmetrically notched beam test. Since the AM, the ASPIN, and the MSPIN methods converge to different solutions for this particular example, we cannot compare the iteration numbers and the computation time, in a fair manner. The obtained speedup varies across different examples, depending on the type of crack propagation, particular loading, and the problem size. As expected, a larger speedup is obtained for problems with the gradual crack propagation. As a consequence, we observe the speedup up to a factor of 1.8 for the problems with “brutal” crack propagation, and up to a factor of 5.57 for the problems with gradual crack propagation. Moreover, as shown in Section 6.6, the speedup of SPIN methods increases with the problem size.

The obtained numerical results also suggest that the alternate minimization configured with Krylov methods (AM-NK, AM-INK) is always faster than the configuration with the direct solver (AM-ND). This is not surprising, as the LU factorization of the sparse linear system with n unknowns requires $\mathcal{O}(n^{3/2})$ flops in $2D$, recall Section 3.1. In contrast, the multigrid preconditioner requires approximately $\mathcal{O}(n)$ flops and it is also optimal in terms of memory usage. As a consequence, the AM-ND and AM-ST methods can be very effective for small problems, but their use becomes prohibitive for larger problems. We also observe that the AM-INK is slightly faster than AM-NK, in particular by a factor of 1.1 on average. The execution time of the ASPIN and the MSPIN methods is comparable,

Table 5

The execution time required to simulate all benchmark problems using ASPIN, MSPIN, AM-ST, AM-ND, AM-NK, and AM-INK methods. The symbols *, †, + denote convergence to a different solution.

Example	Solver	Time (min)	Speedup with respect to				
			AM-ND	AM-NK	AM-INK	AM-ST	ASPIN
Tension	AM-ND	247.72	–	–	–	–	–
	AM-NK	168.92	1.47	–	–	–	–
	AM-INK	157.02	1.58	1.08	–	–	–
	AM-ST	254.10	0.97	0.66	0.62	–	–
	ASPIN	183.07	1.35	0.92	0.86	1.39	–
	MSPIN	131.55	1.88	1.28	1.19	1.93	1.39
Shear	AM-ND	2526.34	–	–	–	–	–
	AM-NK	1,716.82	1.47	–	–	–	–
	AM-INK	1,757.77	1.44	0.98	–	–	–
	AM-ST	1,318.44	1.92	1.30	1.33	–	–
	ASPIN	498.36	5.07	3.44	3.53	2.65	–
	MSPIN	453.78	5.57	3.75	3.87	2.91	1.10
Three-point bending	AM-ND	1,378.35	–	–	–	–	–
	AM-NK	988.34	1.39	–	–	–	–
	AM-INK	855.32	1.61	1.16	–	–	–
	AM-ST	810.00	1.70	1.22	1.06	–	–
	ASPIN	291.01	4.74	3.40	2.94	2.78	–
	MSPIN	260.63	5.29	3.79	3.28	3.11	1.12
L-shaped panel	AM-ND	8,662.82	–	–	–	–	–
	AM-NK	5,748.19	1.51	–	–	–	–
	AM-INK	5,862.43	1.48	0.98	–	–	–
	AM-ST	3,539.46	2.45	1.62	1.66	–	–
	ASPIN	2,674.32	3.24	2.15	2.19	1.32	–
	MSPIN	2,564.15	3.38	2.24	2.29	1.38	1.04
Asymmetrically notched beam	AM-ND*	1,263.54	–	–	–	–	–
	AM-NK*	1,060.75	1.19	–	–	–	–
	AM-INK*	952.49	1.33	1.11	–	–	–
	AM-ST*	563.77	2.24	1.88	1.69	–	–
	ASPIN†	667.23	1.89	1.59	1.43	0.84	–
	MSPIN+	710.64	1.78	1.49	1.34	0.79	0.94

favoring MSPIN by a factor of 1.13 on average. We note that the behavior observed in terms of execution time corresponds to the results reported in terms of iteration count in [Table 4](#).

6.4.1. Analysis of a nonlinear iteration cost

In this section, we analyze the computational time of a single iteration of the AM and the SPIN methods. We consider a three-point bending test with two different mesh resolutions, corresponding to coupled problems with 99,135 and 1,572,147 dofs. We report the time required for solving the displacement subproblems (blue color), the phase-field subproblems (brown color), and the global/coupled linear problem (red color). Furthermore, we also distinguish between the time required by the linear solvers, and the assembly routines, which include an evaluation of the energy, residuals, and Jacobians. The results presented in this section are based on serial executions.

As we can observe from [Fig. 13](#), the execution time of a single nonlinear iteration for SPIN methods is higher than for the AM-NK/INK methods. This is due to the fact that SPIN methods require the solution of a nonlinearly preconditioned linear system of equations, as well as the assembly of coupled residuals and Jacobians. Moreover, SPIN methods also employ a line-search backtracking strategy to ensure that the energy of the coupled problem is decreasing monotonically. In contrast, AM methods only require the solution of displacement and phase-field subproblems, while the time attributed to the global problem is only associated with the transfer of data between different subproblems and convergence monitoring. However, if a direct solver is used then an iteration of AM method is more time-consuming than an iteration of the SPIN method. Moreover, this difference grows with an

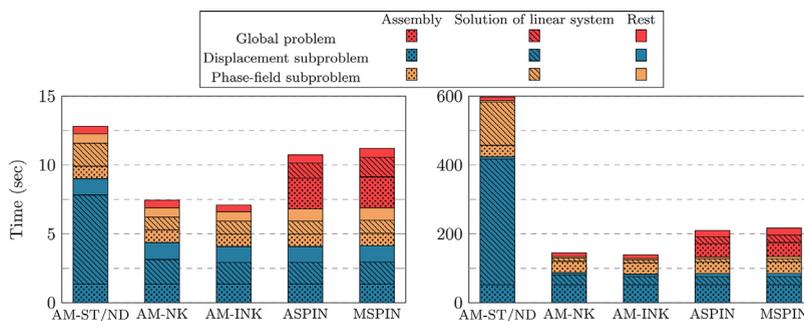


Fig. 13. Analysis of execution time for a nonlinear iteration of alternate minimization and SPIN methods. Time spent solving the displacement subproblem is denoted by blue, the phase-field subproblem by brown, and the global problem by red. The time spent by the assembly of energy, residuals, and Jacobians is depicted by the dotted pattern, while time attributed to the solution of a linear system is depicted by the slanted line pattern. No pattern denotes the time required by all other routines. The experiment was performed in serial using a three-point bending test with 99,135 dofs (left) and 1,572,147 dofs (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

increasing number of dofs. In the end, we also point out that all solution strategies which employ iterative linear solvers spent a larger amount of computational time for assembly than for solutions of linear systems. This is due to the fact that in order to evaluate energy functional and its derivatives on each quadrature point, we have to perform the eigen-decomposition of the strain tensor.

6.5. Memory requirements

In this section, we investigate the memory requirements of the employed solution strategies. The analysis is performed using a three-point bending test with increasing mesh refinements. During the presented study, we distinguish between the memory requirements associated with the assembly of quantities, such as residuals, and Jacobians, and the memory requirements associated with the solution of arising linear systems, see Fig. 14 on the top left and top right, respectively. As we can observe, the memory requirements associated with assembly routines are higher for SPIN methods, as they require the knowledge of coupled residuals and Jacobians. In contrast, AM methods require to store only quantities associated with the phase-field and displacement subproblems. Regarding the memory requirements associated with the solution of linear systems, it is important to distinguish if a direct or iterative linear solver is employed. As we can see, the memory requirements of AM-ST/ND methods grow significantly with an increasing number of dofs. This is due to the fact that the LU factorization of the Jacobians contains a large amount of non-zero elements, giving rise to a so-called memory fill-in. We note that a number of non-zeros grow with $\mathcal{O}(n^{2/3})$ in two dimensions and with $\mathcal{O}(n^2)$ in three dimensions. In contrast, the memory requirements of Krylov methods preconditioned with the multigrid method grow linearly with the problem size. As a consequence, the overall memory requirements of AM-ST/ND are significantly higher for problems with a large number of dofs, see Fig. 14 on the left bottom.

6.6. Efficiency of the solution strategies with increasing mesh resolution

In this section, we investigate the convergence properties of the solution strategies with respect to increasing mesh resolution. During this experiment, we use a three-point bending test, described in Section 5.3. We consider a hierarchy of three meshes, which were obtained by uniformly refining an initial - adaptively refined finite element mesh. The problem on the coarsest mesh (h_0) consists of the 6417 dofs and the problem on the finest level ($h_0/4$) has 99,135 dofs. With each refinement step, the mesh size h and the value of the length-scale parameter l_s decrease by a factor of two. As a consequence, we can approximate the fracture surface more accurately. However, this comes at a higher computational cost for two main reasons. Firstly, the non-linearity of the underlying energy minimization problem tends to strengthen, which causes an increase in the number of nonlinear iterations. Secondly, the computational cost per nonlinear iteration grows due to the increasing number of dofs. Moreover,

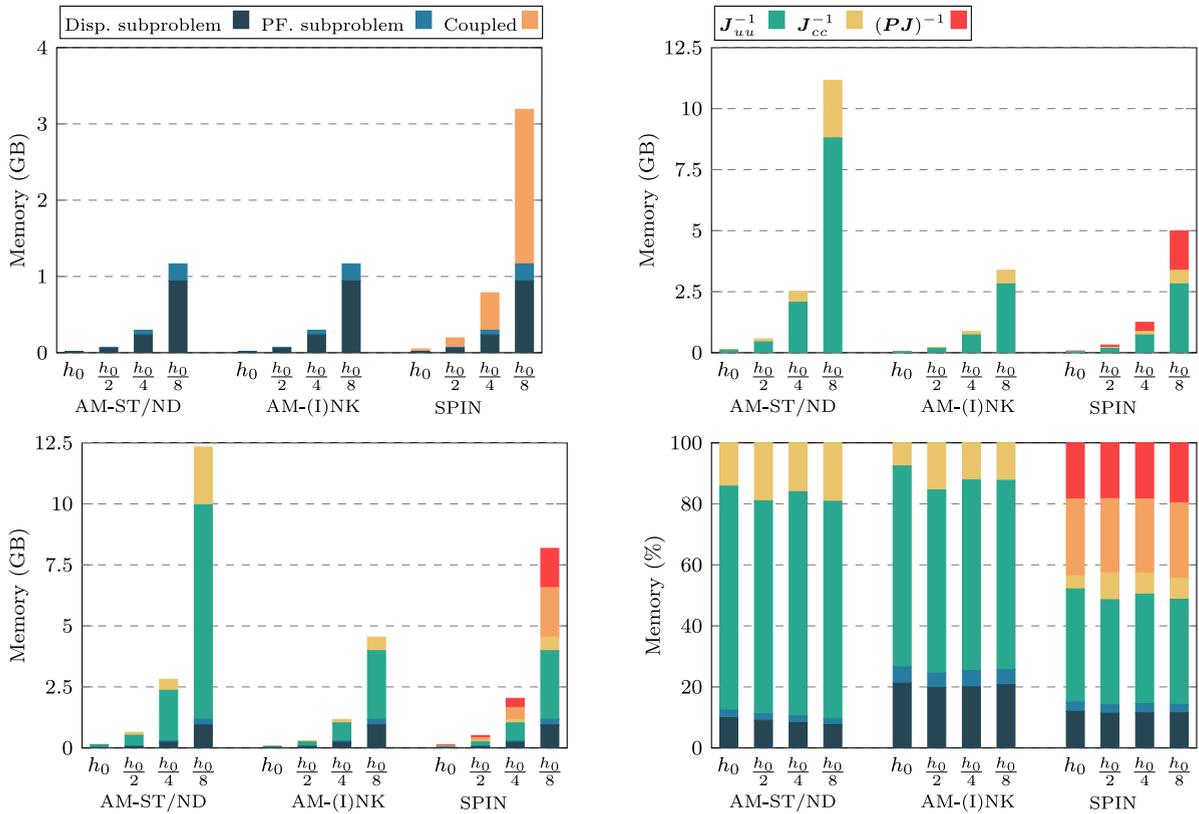


Fig. 14. Memory requirements of different variants of alternate minimization and SPIN methods. Top left: Memory requirement associated with storing Jacobians for phase-field/displacement subproblems and the fully coupled problem. Top right: Memory requirement associated with solution process of each arising linear systems Bottom left: Combined memory requirement for operators and the solution process. Bottom right: The percentage weight of the memory requirement associated with overall system.

the condition number of the arising linear systems grows with each refinement level, which in turn influences the convergence speed of Krylov methods. For instance, the number of required iterations for the BiCGSTAB method grows proportionally with the condition number [39]. In this work, we aim to eliminate this dependence on the condition number by employing the AMG preconditioner, which is known to be mesh independent, in the sense, that the number of iterations should stay constant for an increasing number of dofs. But in practice, this is not easy to achieve as the penalty parameter (γ) employed to ensure the irreversibility condition also increases with each refinement. Here, we remark that even though the SPIN method by itself is a nonlinear preconditioner for the global/coupled problem, a scalable solution strategy is required to solve linear problems arising during the minimization of the phase-field and displacement subproblems (recall Section 4).

Fig. 15 on the left demonstrates that energy can be approximated more accurately as we employ finer meshes. Also, Fig. 15 on the right illustrates how the number of accumulated global nonlinear iterations increases with increasing mesh resolution. Interestingly, an increase in the number of iterations is more prominent for the alternate minimization than for the SPIN methods. This suggests, that using SPIN methods becomes even more beneficial for the larger problems. Indeed, the results reported in Table 6 demonstrate that the SPIN methods achieve a larger speedup compared to the AM method, as the problem size grows. For instance, the MSPIN method achieves a speedup of 2.14 compared to AM-ND, when the mesh with mesh size h_0 is used. In contrast, the speedup of 5.29 is observed for the problem discretized using the twice refined mesh, i.e., with the mesh size $h_0/4$. While in comparison with the AM-ST method, the MSPIN method achieves a speedup of 1.18 for the mesh size h_0 , but this speedup grows almost linearly with increasing problem size, i.e. with mesh sizes $h_0/2, h_0/4$, we observe a speedup of 2.16 and 3.11, respectively.

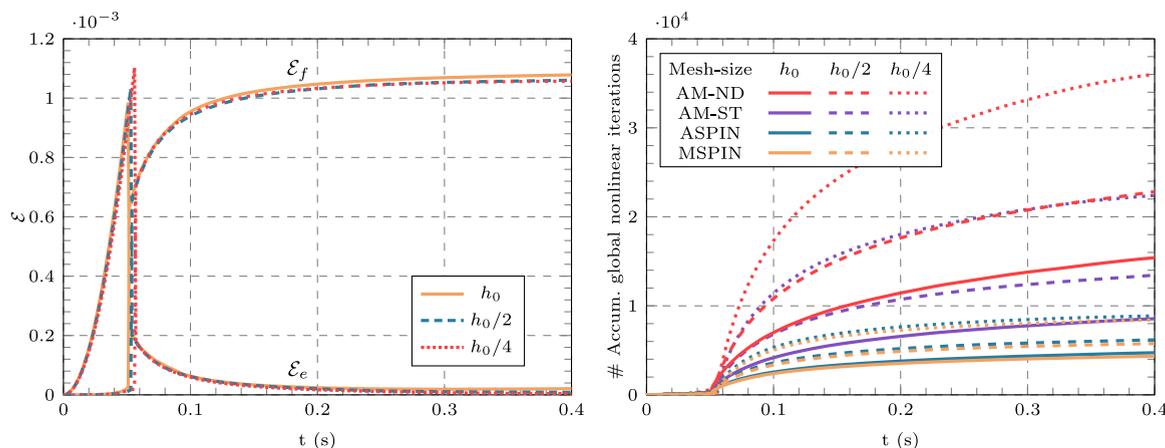


Fig. 15. Three-point bending test. Left: The evolution of elastic and fracture energy over displacement with respect to increasing refinement level. The experiment is performed using the MSPIN method. Right: A number of accumulated global nonlinear iterations as a function of refinement level for AM-ND, AM-ST, ASPIN and MSPIN methods. The symbol h_0 denotes the mesh size of the initial mesh.

Table 6

The execution time for the three-point bending test with respect to varying mesh resolution.

Mesh size (h)	Solver	Time (min)	Speedup with respect to				
			AM-ND	AM-NK	AM-INK	AM-ST	ASPIN
h_0	AM-ND	34.65	–	–	–	–	–
	AM-NK	34.77	1.00	–	–	–	–
	AM-INK	29.88	1.16	1.16	–	–	–
	AM-ST	19.15	1.81	1.82	1.56	–	–
	ASPIN	17.34	2.00	2.01	1.72	1.10	–
	MSPIN	16.21	2.14	2.15	1.84	1.18	1.07
$\frac{h_0}{2}$	AM-ND	192.35	–	–	–	–	–
	AM-NK	167.22	1.15	–	–	–	–
	AM-INK	163.87	1.17	1.02	–	–	–
	AM-ST	121.61	1.58	1.38	1.35	–	–
	ASPIN	61.75	3.11	2.71	2.65	1.97	–
	MSPIN	56.26	3.42	2.97	2.91	2.16	1.10
$\frac{h_0}{4}$	AM-ND	1378.35	–	–	–	–	–
	AM-NK	988.34	1.39	–	–	–	–
	AM-INK	855.32	1.61	1.16	–	–	–
	AM-ST	810.00	1.70	1.22	1.06	–	–
	ASPIN	291.01	4.74	3.40	2.94	2.78	–
	MSPIN	260.63	5.29	3.79	3.28	3.11	1.12

7. Conclusion

In this work, we have proposed a variant of the nonlinear Schwarz Preconditioned Inexact Newton (SPIN) method to solve the nonlinear systems arising in the phase-field fracture simulations in a monolithic manner. Motivated by the robustness of the alternate minimization scheme, we have constructed the nonlinear preconditioner for the coupled problem utilizing the field split approach. Thus, the SPIN preconditioner has been built by minimizing the underlying energy functional separately with respect to the displacement and the phase-field. This minimization has been performed in an additive or multiplicative manner using the inexact Newton method. The numerical performance and the robustness of the newly proposed SPIN methods have been demonstrated using five standard benchmark problems. A comparison with a widely used alternate minimization scheme has been made and it has shown a significant reduction in the execution time, especially for the problems with the gradual crack propagation.

Table A.7

The command line arguments used to configure AM/SPIN methods.

Option	Default value	Description
-snes_atol	10^{-7}	Absolute tolerance (coupled residual).
-snes_rtol	10^{-6}	Relative tolerance (coupled residual).
-snes_stol	10^{-8}	Correction tolerance (coupled correction).
-snes_max_it	50000	A maximum number of iterations.
-snes_am_c_diff_tol	10^{-4}	Absolute tolerance (change in the phase-field variable).
-snes_am_disp_diff_tol	10^{-12}	Absolute tolerance (change in the displacement variable).
-snes_am_exact_solve	true	Usage of the inexact Newton method.
-snes_am_direct_solver	false	Usage of the direct linear solver.
-snes_spin_additive	true	Usage of additive variant of the SPIN method.
-snes_spin_action_rtol	10^{-4}	Relative tolerance (action of SPIN operator).

Moreover, we have also demonstrated that the speedup factor of the SPIN method grows with increasing mesh resolution.

The SPIN algorithms for solving phase-field fracture problems can be extended in several ways. For example, we plan to investigate the convergence behavior of the SPIN method and its efficiency using more complex phase-field fracture models, such as ones, which take into account anisotropy, or nonlinear constitutive laws. We also aim to extend the proposed field-split SPIN method using the nesting domain decomposition approach. In this particular case, a variant of the additive SPIN method, which would employ decomposition of the computational domain into smaller subdomains, could be employed to solve the phase-field and displacement subproblems. This would enable an additional level of parallelism, which could in turn provide a further reduction of computational time. Moreover, we plan to port the implementation of the presented nonlinear field-split preconditioner to GPU-based computing architectures, which would deem an extension of the proposed algorithmic framework into matrix-free settings, such as ones considered in [41,76].

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 authors would like to thank the Swiss National Science Foundation and the Deutsche Forschungsgemeinschaft, Germany (DFG) for their support through the project SPP 1962 “Stress-Based Methods for Variational Inequalities in Solid Mechanics: Finite Element Discretization and Solution by Hierarchical Optimization [186407]”. Additionally, we would also like to gratefully acknowledge the support of Platform for Advanced Scientific Computing (PASC) through projects FraNetG: Fracture Network Growth and FASTER: Forecasting and Assessing Seismicity and Thermal Evolution in geothermal Reservoirs. We also thank the anonymous reviewers for their insightful comments and suggestions which helped us to improve this manuscript.

Appendix A. Command-line options

The implementation of the phase-field fracture model, benchmark problems, and the solution strategies presented in this manuscript is freely available at <https://bitbucket.org/alena.kopanicakova/pf.frac.spin>. Following PETSc methodology, the developed solutions strategies can be configured using command-line options, which are summarized in Table A.7.

Appendix B. Strong-Wolfe conditions

Consider the following minimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad (\text{B.1})$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is an objective function. The k th iteration of an iterative method embedded inside of a line-search framework is given as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}, \quad (\text{B.2})$$

where $\mathbf{p}^{(k)}$ denotes a descent search-direction. In this work, the step size $\alpha^{(k)} \in \mathbb{R}^+$ is obtained by a cubic backtracking line-search method [59, Algorithm A6.3.1, pages 325–327], such that it satisfies the following strong Wolfe conditions [58]:

$$\begin{aligned} f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}) &\leq f(\mathbf{x}_k^{(k)}) + c_1 \alpha^{(k)} \langle \nabla f(\mathbf{x}^{(k)}), \mathbf{p}^{(k)} \rangle \\ |\langle \nabla f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}), \mathbf{p}^{(k)} \rangle| &\leq c_2 |\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{p}^{(k)} \rangle|, \end{aligned} \quad (\text{B.3})$$

where $0 < c_1 < c_2 < 1$. The numerical results reported in Section 6 employ $c_1 = 10^{-4}$ and $c_2 = 0.9$.

Appendix C. Avoiding negative curvature

The line-search methods require that the search direction $\mathbf{p}^{(k)}$ is a descent search-direction [58], i.e., $\langle \mathbf{p}^{(k)}, \nabla f(\mathbf{x}^{(k)}) \rangle < 0$. As the underlying energy functional of the coupled phase-field fracture problem is non-convex, we cannot guarantee that the search directions provided by the SPIN method are directions of descent. Motivated by the trust-region strategies [77], we employ a simple heuristic to ensure that the obtained search directions are always descent directions. More precisely, if the search direction provided by the SPIN algorithm is a direction of ascent, then it is replaced with an inexact Newton direction. If also inexact Newton direction is a direction of ascent, then we use negative gradient as a search-direction instead.

Based on our empirical experience, this situation occurs rather rarely. We encountered negative curvature only once for MSPIN and twice for ASPIN while simulating crack propagation of the asymmetrically notched beam. For all other examples, the search directions provided by SPIN methods were directions of descent.

References

- [1] L. Ambrosio, V.M. Tortorelli, Approximation of functional depending on jumps by elliptic functional via Γ -convergence, *Comm. Pure Appl. Math.* 43 (8) (1990) 999–1036.
- [2] G.A. Francfort, J.-J. Marigo, Revisiting brittle fracture as an energy minimization problem, *J. Mech. Phys. Solids* 46 (8) (1998) 1319–1342.
- [3] A.A. Griffith, The phenomena of rupture and flow in solids, *Philos. Trans. R. Soc. Lond. Ser. A* 221 (1921) 163–198.
- [4] B. Bourdin, G.A. Francfort, J.-J. Marigo, Numerical experiments in revisited brittle fracture, *J. Mech. Phys. Solids* 48 (4) (2000) 797–826.
- [5] C. Miehe, F. Welschinger, M. Hofacker, Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations, *Internat. J. Numer. Methods Engrg.* 83 (10) (2010) 1273–1311.
- [6] 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. Engrg.* 199 (45) (2010) 2765–2778.
- [7] A. Schluter, C. Kuhn, R. Muller, Phase field approximation of dynamic brittle fracture, *PAMM* 14 (1) (2014) 143–144.
- [8] B. Bourdin, C.J. Larsen, C.L. Richardson, A time-discrete model for dynamic fracture based on crack regularization, *Int. J. Fract.* 168 (2) (2011) 133–143.
- [9] G. Del Piero, G. Lancioni, R. March, A variational model for fracture mechanics: numerical experiments, *J. Mech. Phys. Solids* 55 (12) (2007) 2513–2537.
- [10] C. Hesch, A. Gil, R. Ortigosa, M. Dittmann, C. Bilgen, P. Betsch, M. Franke, A. Janz, K. Weinberg, A framework for polyconvex large strain phase-field methods to fracture, *Comput. Methods Appl. Mech. Engrg.* (2017).
- [11] C. Bilgen, A. Kopaničáková, R. Krause, K. Weinberg, A phase-field approach to pneumatic fracture, *PAMM* 17 (1) (2017) 71–74.
- [12] T. Wick, *Multiphysics Phase-Field Fracture: Modeling, Adaptive Discretizations, and Solvers*, Vol. 28, Walter de Gruyter GmbH & Co KG, 2020.
- [13] Y. Heider, A review on phase-field modeling of hydraulic fracturing, *Eng. Fract. Mech.* 253 (2021) 107881.
- [14] J.-Y. Wu, V.P. Nguyen, C.T. Nguyen, D. Sutula, S. Sinaie, S.P. Bordas, Phase-field modeling of fracture, in: *Advances in Applied Mechanics*, Vol. 53, Elsevier, 2020, pp. 1–183.
- [15] X. Zhuang, S. Zhou, G. Huynh, P. Aerias, T. Rabczuk, Phase field modelling and computer implementation: A review, *Eng. Fract. Mech.* (2022) 108234.

- [16] R.U. Patil, B.K. Mishra, Singh, An adaptive multiscale phase field method for brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 329 (2018) 254–288.
- [17] N. Ferro, S. Micheletti, S. Perotto, Anisotropic mesh adaptation for crack propagation induced by a thermal shock in 2D, *Comput. Methods Appl. Mech. Engrg.* 331 (2018) 138–158.
- [18] T. Heister, M.F. Wheeler, T. Wick, A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach, *Comput. Methods Appl. Mech. Engrg.* 290 (2015) 466–495.
- [19] K. Mang, M. Walloth, T. Wick, W. Wollner, Mesh adaptivity for quasi-static phase-field fractures based on a residual-type a posteriori error estimator, *GAMM-Mitt.* 43 (1) (2020) e202000003.
- [20] L.H. Nguyen, D. Schilling, The multiscale finite element method for nonlinear continuum localization problems at full fine-scale fidelity, illustrated through phase-field fracture and plasticity, *J. Comput. Phys.* 396 (2019) 129–160.
- [21] T. Gerasimov, L.D. Lorenzis, A line search assisted monolithic approach for phase-field computing of brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 312 (2016) 276–303.
- [22] A. Kopaničáková, R. Krause, A recursive multilevel trust region method with application to fully monolithic phase-field models of brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112720.
- [23] T. Gerasimov, L. De Lorenzis, Second-order phase-field formulations for anisotropic brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 389 (2022) 114403.
- [24] S. May, J. Vignollet, R. de Borst, A new arc-length control method based on the rates of the internal and the dissipated energy, *Eng. Comput.* (2016).
- [25] N. Singh, C.V. Verhoosel, R. De Borst, E.H. Van Brummelen, A fracture-controlled path-following technique for phase-field modeling of brittle fracture, *Finite Elem. Anal. Des.* 113 (2016) 14–29.
- [26] N. Noii, M. Fan, T. Wick, Y. Jin, A quasi-monolithic phase-field description for orthotropic anisotropic fracture with adaptive mesh refinement and primal-dual active set method, *Eng. Fract. Mech.* (2021) 108060.
- [27] T. Wick, An error-oriented Newton/inexact augmented Lagrangian approach for fully monolithic phase-field fracture propagation, *SIAM J. Sci. Comput.* 39 (4) (2017) B589–B617.
- [28] T. Wick, Modified Newton methods for solving fully monolithic phase-field quasi-static brittle fracture propagation, *Comput. Methods Appl. Mech. Engrg.* 325 (2017) 577–611.
- [29] J.-Y. Wu, Y. Huang, V.P. Nguyen, On the BFGS monolithic algorithm for the unified phase field damage theory, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112704.
- [30] P.K. Kristensen, E. Martínez-Pañeda, Phase field fracture modelling using quasi-Newton methods and a new adaptive step scheme, *Theor. Appl. Fract. Mech.* 107 (2020) 102446.
- [31] J. Wambacq, J. Ulloa, G. Lombaert, S. François, Interior-point methods for the phase-field approach to brittle and ductile fracture, *Comput. Methods Appl. Mech. Engrg.* 375 (2021) 113612.
- [32] O. Lampron, D. Theriault, M. Lévesque, An efficient and robust monolithic approach to phase-field quasi-static brittle fracture using a modified Newton method, *Comput. Methods Appl. Mech. Engrg.* 386 (2021) 114091.
- [33] B. Bourdin, Numerical implementation of the variational formulation for quasi-static brittle fracture, *Interfaces Free Bound.* 9 (2007) 411–430.
- [34] C. Bilgen, A. Kopaničáková, R. Krause, K. Weinberg, A detailed investigation of the model influencing parameters of the phase-field fracture approach, *GAMM-Mitt.* (2019) e202000005.
- [35] P. Farrell, C. Maurini, Linear and nonlinear solvers for variational phase-field models of brittle fracture, *Internat. J. Numer. Methods Engrg.* 109 (5) (2017) 648–667.
- [36] E. Storvik, J.W. Both, J.M. Sargado, J.M. Nordbotten, F.A. Radu, An accelerated staggered scheme for variational phase-field models of brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 381 (2021) 113822.
- [37] M.K. Brun, T. Wick, I. Berre, J.M. Nordbotten, F.A. Radu, An iterative staggered scheme for phase field brittle fracture propagation with stabilizing parameters, *Comput. Methods Appl. Mech. Engrg.* 361 (2020) 112752.
- [38] A. Gupta, U.M. Krishnan, R. Chowdhury, A. Chakrabarti, An auto-adaptive sub-stepping algorithm for phase-field modeling of brittle fracture, *Theor. Appl. Fract. Mech.* 108 (2020) 102622.
- [39] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Siam, 2003.
- [40] C. Bilgen, A. Kopaničáková, R. Krause, K. Weinberg, A phase-field approach to conchoidal fracture, *Meccanica* 53 (2018).
- [41] D. Jodlbauer, U. Langer, T. Wick, Matrix-free multigrid solvers for phase-field fracture problems, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113431.
- [42] C. Gräser, D. Kienle, O. Sander, Truncated nonsmooth Newton multigrid for phase-field brittle-fracture problems, 2020, arXiv preprint arXiv:2007.12290.
- [43] P. Zulian, A. Kopaničáková, M.G.C. Nestola, A. Fink, N.A. Fadel, J. VandeVondele, R. Krause, Large scale simulation of pressure induced phase-field fracture propagation using Utopia, *CCF Trans. High Perform. Comput.* (2021) 1–20.
- [44] M. Badri, G. Rastiello, E. Foerster, Preconditioning strategies for vectorial finite element linear systems arising from phase-field models for fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 373 (2021) 113472.
- [45] L. Svolos, L. Berger-Vergiat, H. Waisman, Updating strategy of a domain decomposition preconditioner for parallel solution of dynamic fracture problems, *J. Comput. Phys.* 422 (2020) 109746.
- [46] X.-C. Cai, D.E. Keyes, Nonlinearly preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* 24 (1) (2002) 183–200.
- [47] L. Liu, D.E. Keyes, Field-split preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* 37 (3) (2015) A1388–A1409.
- [48] L. Liu, D.E. Keyes, R. Krause, A note on adaptive nonlinear preconditioning techniques, *SIAM J. Sci. Comput.* 40 (2) (2018) A1171–A1186.

- [49] C. Groß, R. Krause, On the globalization of ASPIN employing trust-region control strategies—convergence analysis and numerical examples, 2021, arXiv preprint arXiv:2104.05672.
- [50] C. Kuhn, A. Schlüter, R. Müller, On degradation functions in phase field fracture models, *Comput. Mater. Sci.* 108 (2015) 374–384.
- [51] J.M. Sargado, E. Keilegavlen, I. Berre, J.M. Nordbotten, High-accuracy phase-field models for brittle fracture based on a new family of degradation functions, *J. Mech. Phys. Solids* 111 (2018) 458–489.
- [52] A. Giacomini, Ambrosio-tortorelli approximation of quasi-static evolution of brittle fractures, *Calc. Var. Partial Differential Equations* 22 (2) (2005) 129–172.
- [53] L. De Lorenzis, C. Maurini, Nucleation under multi-axial loading in variational phase-field models of brittle fracture, *Int. J. Fract.* (2021) 1–21.
- [54] H. Amor, J.-J. Marigo, C. Maurini, Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments, *J. Mech. Phys. Solids* 57 (8) (2009) 1209–1229.
- [55] F. Freddi, G. Royer-Carfagni, Regularized variational theories of fracture: a unified approach, *J. Mech. Phys. Solids* 58 (8) (2010) 1154–1174.
- [56] M.F. Wheeler, T. Wick, W. Wollner, An augmented-Lagrangian method for the phase-field approach for pressurized fractures, *Comput. Methods Appl. Mech. Engrg.* 271 (2014) 69–85.
- [57] T. Gerasimov, L. De Lorenzis, On penalization in variational phase-field models of brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 354 (2019) 990–1026.
- [58] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.
- [59] J. Dennis, R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996.
- [60] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, *SIAM J. Sci. Comput.* 17 (1) (1996) 16–32.
- [61] X.-C. Cai, W.D. Gropp, D.E. Keyes, M.D. Tidriri, Newton-Krylov-Schwarz methods in CFD, in: *Numerical Methods for the Navier-Stokes Equations*, Springer, 1994, pp. 17–30.
- [62] V. Dolean, M.J. Gander, W. Kheriji, F. Kwok, R. Masson, Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method, *SIAM J. Sci. Comput.* 38 (6) (2016) A3357–A3380.
- [63] P.J. Lanzkron, D.J. Rose, J.T. Wilkes, An analysis of approximate nonlinear elimination, *SIAM J. Sci. Comput.* 17 (2) (1996) 538–559.
- [64] P. Deufhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Vol. 35, Springer Science & Business Media, 2011.
- [65] M. Ambati, T. Gerasimov, L. De Lorenzis, A review on phase-field models of brittle fracture and a new fast hybrid formulation, *Comput. Mech.* 55 (2) (2015) 383–405.
- [66] B.J. Winkler, *Load Analysis of Unreinforced and Reinforced Concrete Structures on the Basis of an Objective Material Law for Concrete*, Innsbruck University Press, 2001.
- [67] C. Hesch, K. Weinberg, Thermodynamically consistent algorithms for a finite-deformation phase-field approach to fracture, *Internat. J. Numer. Methods Engrg.* 99 (12) (2014) 906–924.
- [68] T. Bittencourt, P. Wawrzynek, A. Ingraffea, J. Sousa, Quasi-automatic simulation of crack propagation for 2D LEFM problems, *Eng. Fract. Mech.* 55 (2) (1996) 321–334.
- [69] T. Gerasimov, U. Römer, J. Vondřejc, H.G. Matthies, L. De Lorenzis, Stochastic phase-field modeling of brittle fracture: computing multiple crack patterns and their probabilities, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113353.
- [70] D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandie, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, *Nucl. Eng. Des.* 239 (10) (2009) 1768–1778.
- [71] P. Chakraborty, Y. Zhang, M.R. Tonks, Multi-scale modeling of microstructure dependent intergranular brittle fracture using a quantitative phase-field based method, *Comput. Mater. Sci.* 113 (2016) 38–52.
- [72] P. Zulian, A. Kopaničáková, M.C.G. Nestola, A. Fink, N. Fadel, A. Rigazzi, V. Magri, T. Schneider, E. Botter, J. Mankau, R. Krause, Utopia: A C++ embedded domain specific language for scientific computing. Git repository, 2016, URL <https://bitbucket.org/zulianp/utopia>.
- [73] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, et al., *PETSc users manual revision 3.5*, Argonne National Lab. (ANL) (2014).
- [74] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, J. Koster, MUMPS: a general purpose distributed memory sparse solver, in: *International Workshop on Applied Parallel Computing*, Springer, 2000, pp. 121–130.
- [75] U.M. Yang, et al., Boomeramg: A parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.* 41 (1) (2002) 155–177.
- [76] H. Kothari, A. Kopaničáková, R. Krause, A multigrid preconditioner for Jacobian-free Newton-Krylov methods, 2021, arXiv preprint arXiv:2103.13890.
- [77] A.R. Conn, N.I.M. Gould, P.L. Toint, Trust Region Methods, in: *MOS-SIAM Series on Optimization*, SIAM, 2000.