



# Truncated nonsmooth Newton multigrid for phase-field brittle-fracture problems, with analysis

Carsten Gräser<sup>1</sup> · Daniel Kienle<sup>2</sup> · Oliver Sander<sup>3</sup>

Received: 2 December 2022 / Accepted: 24 March 2023 / Published online: 20 May 2023  
© The Author(s) 2023

## Abstract

We propose the truncated nonsmooth Newton multigrid method (TNNMG) as a solver for the spatial problems of the small-strain brittle-fracture phase-field equations. TNNMG is a nonsmooth multigrid method that can solve biconvex, block-separably nonsmooth minimization problems with linear time complexity. It exploits the variational structure inherent in the problem, and handles the pointwise irreversibility constraint on the damage variable directly, without regularization or the introduction of a local history field. In the paper we introduce the method and show how it can be applied to several established models of phase-field brittle fracture. We then prove convergence of the solver to a solution of the nonsmooth Euler–Lagrange equations of the spatial problem for any load and initial iterate. On the way, we show several crucial convexity and regularity properties of the models considered here. Numerical comparisons to an operator-splitting algorithm show a considerable speed increase, without loss of robustness.

**Keywords** Phase-field · Brittle fracture · Spectral strain decomposition · Convex analysis · Nonsmooth multigrid · Global convergence

## 1 Introduction

The equations of phase-field models of brittle fracture present a number of challenges to the designers of numerical solution algorithms [2]. Even in the small-strain case the equations are nonlinear, due to the multiplicative coupling of the mechani-

cal stresses to the degradation function. At the same time the non-healing condition introduces an inequality constraint. Finally, eigenvalue-based splittings of the energy density as in [38] make the equations nondifferentiable.

In this paper we focus on the spatial problems of small-strain brittle-fracture phase-field models obtained by a suitable time discretization. The standard approach to solving these spatial problems is based on operator splitting. Algorithms based on this approach, also known as staggered schemes, alternate between solving a displacement problem with fixed damage and a damage problem with fixed displacement. Both subproblems are elliptic and well-understood, and such methods are therefore straightforward to implement. The method can be interpreted as a nonlinear Gauß–Seidel method [16], which provides a natural framework for convergence proofs. Applications of the operator splitting scheme and its extensions appear, e.g., in [8, 10, 11]. With particular semi-implicit time discretizations it is also possible to solve the spatial problems by solving only one damage problem and one displacement problem [37], without iterating. This is very fast, but works only if the load steps are small enough.

In contrast, other works propose monolithic solution schemes based on Newton’s method [16, 17, 52, 54, 55]. For

---

The authors gratefully acknowledge the financial support by the German Federal Ministry of Education and Research through the ParaPhase project within the framework “IKT 2020 – Forschung für Innovationen” (Project Numbers 01-H15005C, 01-15005D, 01-15005E).

✉ Oliver Sander  
oliver.sander@tu-dresden.de

Carsten Gräser  
graeser@math.fau.de

Daniel Kienle  
kienle@mechbau.uni-stuttgart.de

<sup>1</sup> Department Mathematik, Friedrich-Alexander-Universität Erlangen–Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

<sup>2</sup> Institut für Angewandte Mechanik, Universität Stuttgart, Pfaffenwaldring 7, 70569 Stuttgart, Germany

<sup>3</sup> Institut für Numerische Mathematik, Technische Universität Dresden, Zellescher Weg 12–14, 01069 Dresden, Germany

the unmodified Newton method only local convergence can be shown, and failure to converge for large load steps is readily observed in practice [57]. Therefore, various authors have proposed extensions or modifications of the Newton idea to stabilize the method. In [17] a line search strategy is applied to enlarge the domain of convergence of Newton's method. [57] and [29] propose to use the BFGS quasi-Newton algorithm, claiming that it is more stable than Newton's method and more efficient than operator splitting. [24] proposed a modified Newton scheme which was later improved by [55] with an adaptive transition from Newton's method to the modified Newton scheme. Recent results in [26] suggest that nonlinear preconditioning can speed up the solution process tremendously. Finally, the authors of [36, 48] suggest an arc-length method based on the fracture surface, and an adaptive time stepping scheme to enhance the robustness. In summary, while monolithic Newton-type methods are reported to be faster than operator-splitting algorithms, the latter ones are more robust [2, 57].

Various approaches are used in the literature to deal with the damage irreversibility. A natural approach is to regularize the constraint, as investigated in [26, 38, 53]. This leads to an additional parameter, and to ill-conditioned tangent matrices [18]. Interior-point solvers implement an automatic control of the new parameter; they are investigated in [52]. An alternative approach considers the thermodynamic driving force of the fracture phase-field as a global unknown yielding a three-field formulation which results in a saddle-point principle [38]. A third formulation considers the Karush–Kuhn–Tucker conditions and shifts the thermodynamic driving force of the fracture phase-field into a local history field representing the maximum over time of the elastic energy [37]. This approach, frequently known as the  $\mathcal{H}$ -field technique, therefore trades the inequality constraint for the nondifferentiable maximum function. Alternatively, the time discrete problems can be reformulated as semismooth systems by means of so-called complementarity functions. This strategy can be combined with monolithic semismooth Newton techniques [34] or nested Newton and active set methods [24]. Unfortunately, these approaches spoil the variational structure of the spatial problems. Augmented Lagrangian solvers as in [53] introduce extra variables. Closest in spirit to the present manuscript is the use of bound-constrained optimization solvers, used, e.g., in [4, 11, 16, 56]. None of these approaches are fully satisfactory.

The effect of the nondifferentiable terms caused by anisotropic splittings of the mechanical energy density as in [4, 38, 49] is rarely discussed in the literature. Hybrid formulations like the one proposed in [2] try to overcome the additional computational difficulties of these splittings by further changes to the model, again at the cost of sacrificing the variational structure.

All these approaches are slow in the sense that they have to solve global partial differential equations at each Newton or operator-splitting iteration. This is expensive, even if efficient multigrid methods are used for the linear subproblems (as, e.g., in [25]). When the methods use direct sparse solvers for the linear tangent problems, memory consumption can become problematic, too. At the same time, the problem of small-strain phase-field brittle fracture has a lot of elegant variational structure; in particular, it fits directly into the rate-independent framework of [39]. As a consequence, implicit time discretization leads to a sequence of coercive minimization problems for the displacement and damage fields together. These problems are not convex, but they are biconvex, i.e., convex (even strongly convex) in each variable separately. Pointwise inequality restrictions  $\dot{d} \geq 0$  to handle the irreversibility of the fracture process as proposed in [38] reduce the smoothness of the objective functional, but do not influence its convexity or coercivity properties. The same holds for anisotropic energy splittings based on linear quantities or the eigenvalues of the mechanical strain.

Recent years have shown that nonsmooth multigrid methods are able to solve variational nonsmooth problems from mechanics efficiently without the need for solving global linear systems of equations. In works like [20, 22, 23, 27, 28, 47], such multigrid methods have shown to be vastly more efficient than operator-splitting or Newton-based methods. As there are no sparse matrix factorizations, memory consumption remains linear in the number of unknowns. In addition, these multigrid methods can be shown to converge globally (i.e., from any initial iterate and for any load step) to a stationary point of the objective functional. The proof exploits the above-mentioned variational structure, together with certain separability properties. As one such method, the Truncated Nonsmooth Newton Multigrid method (TNNMG) can treat the pointwise constraints of the increment problems directly, i.e., without artificial regularization or tricks like the  $\mathcal{H}$ -field technique [37]. The idea is that TNNMG only needs to handle these constraints in a series of low-dimensional subproblems, each of which is easy to solve by itself. As a consequence, solving the problems with constraints is not appreciably slower than solving the corresponding unconstrained problem.

In this paper we show how the TNNMG method can be used to solve small-strain brittle-fracture problems. This involves in particular verifying that the increment functionals have the required convexity and smoothness properties. We do this for a range of different degradation functions and local crack surface densities (including the standard Ambrosio–Tortorelli functionals of type 1 and 2), closely related to the family of models considered in [12]. We cover elastic energies with various types of anisotropic splittings, including the splitting based on strain eigenvalues of [38]. For the proofs we use results from the convex analysis of spectral functions [5,

44]. Extension to the slightly more general damage models of [35] is straightforward, provided the stored elastic energy has the required convexity and smoothness properties. In contrast to the multilevel trust region method proposed in [27], the TNNMG method presented here relies on nonsmooth Newton techniques leading to linear subproblems and thus gives more flexibility in the selection of coarse grid solvers.

The paper is organized as follows: Sect. 2 discusses a framework of small-strain phase-field models for brittle fracture, and shows the range of applicability of the TNNMG solver. Section 3 introduces the natural fully implicit time discretization, and proves existence of solutions for the spatial problems. In both sections we pay particular attention to the mathematical properties of the energy functionals. In Sect. 4, finally, we introduce the TNNMG method. We explain its construction, discuss various algorithmic options, and prove that it converges globally to stationary points of the increment energy functional. The numerical efficiency is then demonstrated in Sect. 5. Our reference for comparison, briefly revisited in Sect. 5.1, is the operator-splitting iteration proposed in [12], which uses a projected Newton method [7] for the constrained damage problems. We compare the solvers for two- and three-dimensional example problems with different forms of the local crack surface density, and with and without spectral splittings. We observe a noticeable performance increase, without loss of robustness.

## 2 Phase-field models of brittle fracture

This section presents a range of phase-field models for brittle fracture, and discusses its smoothness and convexity properties.

Consider a deformable  $m$ -dimensional object represented by a domain  $\Omega \in \mathbb{R}^m$ . The deformation of such an object is characterized by a displacement field  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^m$ . The object is supposed to exhibit small-strain deformations and elastic material behavior only, and we therefore introduce the linearized strain tensor  $\boldsymbol{\varepsilon}(\mathbf{u}) := \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ . Following [38], we model the fracturing by a scalar damage field  $d : \Omega \rightarrow [0, 1]$ , where  $d = 0$  signifies intact material, and  $d = 1$  a fully broken one. Dirichlet boundary conditions can be posed both for the displacement and for the damage field. For this we select two not necessarily equal subsets  $\Gamma_{D,\mathbf{u}}, \Gamma_{D,d} \subset \partial\Omega$  of the domain boundary, and require

$$\mathbf{u} = \mathbf{u}_0 \quad \text{on } \Gamma_{D,\mathbf{u}}, \quad d = d_0 \quad \text{on } \Gamma_{D,d},$$

where  $\mathbf{u}_0$  and  $d_0$  are two given functions.

Displacement and damage field evolve together, governed by a system of coupled nonsmooth partial differential equations. Disregarding inertia effects, we obtain a rate-

independent system in the sense of [39]. Such a system can be written using the Biot equation

$$D_{(\mathbf{u},d)}\mathcal{E}(t, \mathbf{u}, d) + \partial_{\dot{d}}\mathcal{R}(d, \dot{d}) \ni 0, \tag{1}$$

where  $D_{(\mathbf{u},d)}\mathcal{E}(t, \mathbf{u}, d)$  means the Gâteaux derivative with respect to the second and third arguments of  $\mathcal{E}$ , and  $\partial_{\dot{d}}\mathcal{R}(d, \dot{d})$  is the convex subdifferential with respect to the second argument of the dissipation potential  $\mathcal{R}$ .

In this equation,  $\mathcal{E}$  is a potential energy, which we assume to be of the form

$$\begin{aligned} \mathcal{E}(t, \mathbf{u}, d) = & \int_{\Omega} \psi(\boldsymbol{\varepsilon}(\mathbf{u}), d) dV + \int_{\Omega} g_c \gamma(d, \nabla d) dV \\ & + P_{\text{ext}}(t, \mathbf{u}) \\ & + \int_{\Omega} I_{[0,1]}(d) dV. \end{aligned} \tag{2}$$

The term  $\psi$  is a degraded elastic energy density, and will be discussed in detail in Sect. 2.1. The term  $\gamma$  models the local crack surface density, and will be discussed in Sect. 2.2. The number  $g_c$  is Griffith’s critical energy release rate, a material parameter.  $P_{\text{ext}}$  represents time-dependent volume and surface forces, which drive the evolution. We assume that  $P_{\text{ext}}$  is linear and  $H^1(\Omega)$ -continuous in  $\mathbf{u}$ , and differentiable in  $t$  with bounded time derivative.

The last term of (2) implements the restriction that the damage field can only assume values between 0 and 1. For a set  $\mathcal{K} \subset \mathbb{R}$  we define the indicator functional

$$I_{\mathcal{K}} : \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}, \quad I_{\mathcal{K}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{K}, \\ \infty & \text{otherwise.} \end{cases}$$

For a closed, convex, nonempty set  $\mathcal{K}$ , the functional  $I_{\mathcal{K}}$  is convex, lower semicontinuous, and proper. Adding the constraint  $d \in [0, 1]$  explicitly is not always necessary, as some fracture models lead to evolutions that satisfy the constraints implicitly. However, as pointwise bounds come with practically no cost when using the TNNMG solver, we do include them to extend our range of models.

To make the potential energy  $\mathcal{E}$  well defined, we will in general consider it on the first-order Sobolev space  $H^1(\Omega, \mathbb{R}^m) \times H^1(\Omega, \mathbb{R})$ . Incorporating the boundary conditions leads to the affine subspace

$$\begin{aligned} \mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1 := & \left\{ \mathbf{v} \in H^1(\Omega, \mathbb{R}^m) \mid \mathbf{v}|_{\Gamma_{D,\mathbf{u}}} = \mathbf{u}_0 \right\} \\ & \times \left\{ v \in H^1(\Omega) \mid v|_{\Gamma_{D,d}} = d_0 \right\}. \end{aligned}$$

The second term of the Biot equation (1) is  $\partial_{\dot{d}}\mathcal{R}(d, \dot{d})$ , where  $\mathcal{R}$  is the dissipation potential

$$\mathcal{R}(d, \dot{d}) = \int_{\Omega} I_{[0,\infty)}(\dot{d}) dV. \tag{3}$$

It implements the pointwise non-healing condition  $\dot{d} \geq 0$  as proposed by [38]. Note that  $\mathcal{R}(d, \cdot) : H^1(\Omega) \rightarrow [0, \infty]$  is convex and lower semicontinuous, and  $\mathcal{R}(d, 0) = 0$ . The fact that  $\mathcal{R}$  is positively 1-homogeneous in  $\dot{d}$  implies the rate-independence of the system. Since the particular functional (3) only depends on  $\dot{d}$  but not on  $d$ , we will also write  $\mathcal{R}(\dot{d}) := \mathcal{R}(d, \dot{d})$  and  $\partial \mathcal{R}(\dot{d}) := \partial_{\dot{d}} \mathcal{R}(d, \dot{d})$ .

**Remark 2.1** Using the definition of the subdifferential  $\partial \mathcal{R}$  it is straightforward to see that the Biot equation (1) is equivalent to the variational inequality

$$\langle D_{(u,d)} \mathcal{E}(t, \mathbf{u}, d), (\mathbf{v}, e) - (\mathbf{u}, \dot{d}) \rangle + \mathcal{R}(e) \geq \mathcal{R}(\dot{d}) \quad \forall (\mathbf{v}, e) \in \mathbf{H}_{u_0}^1 \times H_0^1. \quad (4)$$

Likewise, it is equivalent to the coupled system

$$\begin{aligned} \langle D_u \mathcal{E}(t, \mathbf{u}, d), \mathbf{v} \rangle &= 0 & \forall \mathbf{v} \in \mathbf{H}_0^1, \\ \langle D_d \mathcal{E}(t, \mathbf{u}, d), e - \dot{d} \rangle + \mathcal{R}(e) &\geq \mathcal{R}(\dot{d}), & \forall e \in H_0^1, \end{aligned}$$

where we have denoted by  $\mathbf{H}_0^1 \times H_0^1 := \mathbf{H}_{u_0}^1 \times H_{d_0}^1 - (\mathbf{u}_0, d_0)$  the homogeneous space corresponding to  $\mathbf{H}_{u_0}^1 \times H_{d_0}^1$ . Since these two notions of solutions are based on first-order derivatives of  $\mathcal{E}$  and  $\mathcal{R}$ , they may lead to local minimizers or saddle points of the functional  $\mathcal{E}$ . To overcome this, there are alternative so-called *energetic formulations* of the problem. In general energetic solutions are solutions of (4), while the converse is only true for convex  $\mathcal{E}$ . We will not discuss such solutions here, but refer to [39, 50], where they are discussed for damage-related and more general rate-independent processes.

**Remark 2.2** In the engineering literature, the same problem is sometimes formulated as

$$\partial_{(\dot{\mathbf{u}}, \dot{d})} \Pi(\dot{\mathbf{u}}, \dot{d}; \mathbf{u}, d) \ni 0,$$

with the rate potential

$$\Pi(\dot{\mathbf{u}}, \dot{d}; \mathbf{u}, d) := \frac{d}{dt} \mathcal{E}_0(\mathbf{u}, d) + P_{\text{ext}}(t, \dot{\mathbf{u}}) + \mathcal{R}(d, \dot{d}),$$

(see, e.g., [38, Section 4.2]), where  $\mathcal{E}_0(\mathbf{u}, d)$  denotes the parts of the energy  $\mathcal{E}(t, \mathbf{u}, d)$  that do not explicitly depend on  $t$ . If  $P_{\text{ext}}(t, \mathbf{u})$  is linear in  $\mathbf{u}$ , and if the problem is sufficiently smooth, then this formulation is equivalent to the Biot equation (1). Indeed, we then have

$$\Pi(\dot{\mathbf{u}}, \dot{d}; \mathbf{u}, d) := D_{(u,d)} \mathcal{E}_0(\mathbf{u}, d)(\dot{\mathbf{u}}, \dot{d}) + P_{\text{ext}}(t, \dot{\mathbf{u}}) + \mathcal{R}(d, \dot{d}),$$

and

$$\begin{aligned} \partial_{(\dot{\mathbf{u}}, \dot{d})} \Pi(\dot{\mathbf{u}}, \dot{d}; \mathbf{u}, d) &= D_{(u,d)} \mathcal{E}_0(\mathbf{u}, d) + P_{\text{ext}}(t, \cdot) \\ &\quad + \partial_{\dot{d}} \mathcal{R}(d, \dot{d}) \\ &= D_{(u,d)} [\mathcal{E}_0(\mathbf{u}, d) + P_{\text{ext}}(t, \mathbf{u})] \\ &\quad + \partial_{\dot{d}} \mathcal{R}(d, \dot{d}). \end{aligned}$$

Requiring this to contain 0 is the Biot equation (1).

## 2.1 Degraded elastic energy density

We consider models that behave linearly elastic and isotropic if the material is in an undamaged state. That is, for the undamaged stored energy density we use the St. Venant–Kirchhoff material law, whose energy density is given by

$$\psi_0(\boldsymbol{\varepsilon}) = \frac{\lambda}{2} \text{tr}[\boldsymbol{\varepsilon}]^2 + \mu \text{tr}[\boldsymbol{\varepsilon}^2],$$

with Lamé parameters  $\mu > 0$  and  $\lambda > -\frac{2}{3}\mu$ . With this choice of parameters the quadratic functional  $\psi_0$  is strongly convex on  $\mathbb{S}^m$ , the set of real symmetric  $m \times m$  matrices.

The undamaged energy density is split as

$$\psi_0(\boldsymbol{\varepsilon}) = \psi_0^+(\boldsymbol{\varepsilon}) + \psi_0^-(\boldsymbol{\varepsilon})$$

into a part  $\psi_0^+$  that produces damage and another part  $\psi_0^-$  that does not. The damage-producing part is then scaled by a so-called degradation function

$$g : [0, 1] \rightarrow [0, 1],$$

and the energy density  $\psi : \mathbb{S}^m \times [0, 1] \rightarrow \mathbb{R}$  takes the form

$$\psi(\boldsymbol{\varepsilon}, d) = [g(d) + k] \psi_0^+(\boldsymbol{\varepsilon}) + (1 + k) \psi_0^-(\boldsymbol{\varepsilon}). \quad (5)$$

The residual stiffness  $k > 0$  guarantees a well-posed problem in case of fracture.

Various different degradation functions have appeared in the literature [12, 30, 49]. While the details vary, there appears to be agreement on the following properties:

**Assumption 2.3** The degradation function  $g : [0, 1] \rightarrow [0, 1]$  is differentiable, monotone decreasing, and fulfills  $g(0) = 1$  and  $g(1) = 0$ .

**Remark 2.4** As an alternative to this assumption, one could consider degradation functions with  $g(0) = 1 - k$  such that the original energy density  $\psi_0$  is exactly recovered in the undamaged case (see, e.g., [54]). However, in the typical regime of  $k \ll 1$  both will essentially yield the same result. Thus we will follow the more common approach  $g(0) = 1$  here, although the proposed methods could equally be applied to the alternative one.

Note that several authors assume  $g'(1) = 0$  in order to ensure that the evolution does not lead to values of  $d$  larger than 1 (cf., e.g., [42]). We do not need this assumption on  $g'$  here, because the pointwise constraint  $d \leq 1$  is enforced explicitly by the energy term (2).

The following specific degradation functions all fulfill Assumption 2.3:

$$g_a(d) = (1 - d)^2 \tag{from [9]}$$

$$g_b(d) = (1 - d)^2 \cdot (2d + 1) \tag{from [30]}$$

$$g_c(d) = (1 - d)^3 \cdot (3d + 1) \tag{from [30]}$$

$$g_d(d) = \frac{\exp(bd) - (b(d - 1) + 1)\exp(b)}{(b - 1)\exp(b) + 1}, \quad b > 0 \tag{from [49)].}$$

Note that the functions  $g_a$  and  $g_d$  are strictly convex, but  $g_b$  and  $g_c$  are not even convex. For the rest of the paper we will restrict our considerations to convex twice continuously differentiable degradation functions  $g$ .

Various splittings of  $\psi_0$  have been proposed in the literature. We cover four common strain-based splittings taking the form (5).<sup>1</sup> All those splittings have the property that  $\psi_0 = \psi_0^+ + \psi_0^-$ , and we will show that all have the following essential properties:

- (P1)  $\psi(\boldsymbol{\varepsilon}, \cdot) \in C^2$  for all  $\boldsymbol{\varepsilon} \in \mathbb{S}^m$  and  $\psi(\cdot, d) \in C^{1,1}$  for all  $d \in [0, 1]$ , i.e.,  $\psi(\cdot, d)$  is differentiable with locally Lipschitz continuous derivative.
- (P2) The gradient  $\nabla\psi(\cdot, d)$  is semismooth for all  $d \in [0, 1]$ .
- (P3) The gradient  $\nabla\psi(\cdot, d)$  is globally Lipschitz continuous uniformly in  $d$ , i.e., there exists  $L \geq 0$  independent of  $d$  such that for all matrices  $A, B \in \mathbb{S}^m$  we have

$$|\nabla\psi(A, d) - \nabla\psi(B, d)| \leq L|A - B|_F,$$

where  $|\cdot|_F$  denotes the Frobenius norm.<sup>2</sup>

- (P4)  $\psi(\cdot, d) : \mathbb{S}^m \rightarrow \mathbb{R}$  is strongly convex uniformly in  $d$ , i.e., there exists  $\eta > 0$  independent of  $d$  such that for all matrices  $A, B \in \mathbb{S}^m$  we have

$$\psi(tA + (1 - t)B, d) \leq t\psi(A, d) + (1 - t)\psi(B, d) - \frac{1}{2}\eta t(1 - t)|A - B|_F^2.$$

<sup>1</sup> The stress-based splitting of [49] is left for future work.

<sup>2</sup> While we only require  $L \geq 0$  in (P3), the strong convexity (P4) in fact implies  $L > 0$ .

- (P5)  $\psi(\cdot, d)$  is coercive uniformly in  $d$  in the sense that there exists  $C > 0$  independent of  $d$  such that  $\psi(\boldsymbol{\varepsilon}, d) \geq C|\boldsymbol{\varepsilon}|_F^2$ .<sup>3</sup>

We remind the readers that the gradient  $\nabla\psi(\cdot, d)$  is called semismooth if for any point  $A \in \mathbb{S}^m$  and any direction  $V \in \mathbb{S}^m$  the limit

$$\lim_{n \rightarrow \infty} G_n V_n$$

exists and is unique for all sequences  $V_n$  and  $G_n$  with  $V_n \rightarrow V$  and  $G_n \in \partial(\nabla\psi(\cdot, d))(A + t_n V_n)$  for a sequence  $t_n \searrow 0$ . The set  $\partial(\nabla\psi(\cdot, d))(A)$  denotes Clarke’s generalized Jacobian of the locally Lipschitz continuous map  $\nabla\psi(\cdot, d) : \mathbb{S}^m \rightarrow \mathbb{S}^m$  at  $A \in \mathbb{S}^m$  (cf. [45]). Notice that the strong convexity (P4) implies strong monotonicity of  $\nabla\psi$ , i.e.,

$$\langle \nabla\psi(A) - \nabla\psi(B), A - B \rangle \geq \eta|A - B|_F^2.$$

Here and in the following we denote by  $\langle \cdot, \cdot \rangle : \mathcal{V}^* \times \mathcal{V} \rightarrow \mathbb{R}$  the duality pairing of a vector space  $\mathcal{V}$ .

For the splittings considered in the following we will only prove (P1) and (P2) directly, and show that the simplified assumptions of the following lemma hold true. This then implies (P3), (P4), and (P5).

**Lemma 2.5** *Let  $\psi_0^+$  and  $\psi_0^-$  be convex, non-negative, and differentiable with Lipschitz continuous gradients  $\nabla\psi_0^+$  and  $\nabla\psi_0^-$ . Then  $\psi$  satisfies (P3), (P4), and (P5).*

**Proof** Let  $L^+$  and  $L^-$  be the Lipschitz constants of  $\nabla\psi_0^+$  and  $\nabla\psi_0^-$ , respectively. Then  $\nabla\psi(\cdot, d)$  is Lipschitz continuous with uniform Lipschitz constant  $(1 + k)(L^+ + L^-)$ , because  $g(d) + k \leq 1 + k$ .

To show strong convexity, we first note that  $\psi_0 = \psi_0^+ + \psi_0^-$  is strongly convex on  $\mathbb{S}^m$  with a modulus  $\eta > 0$  independent of  $d$ . Now consider the function

$$\boldsymbol{\varepsilon} \mapsto \psi(\boldsymbol{\varepsilon}, d) - k\psi_0(\boldsymbol{\varepsilon}) = g(d)\psi_0^+(\boldsymbol{\varepsilon}) + \psi_0^-(\boldsymbol{\varepsilon}).$$

Since this is a weighted sum of two convex functions  $\psi_0^+$  and  $\psi_0^-$  with non-negative weights  $g(d) \geq 0$  and 1, it is itself convex. Thus, as a sum of this convex function and the strongly convex functions  $C\psi_0$ , the function  $\psi(\cdot, d)$  is itself strongly convex and inherits the convexity modulus  $k\eta$  of  $k\psi_0$ . Finally, we note that with the same  $\eta$  we have

$$\psi(\boldsymbol{\varepsilon}, d) \geq k\psi_0(\boldsymbol{\varepsilon}) \geq \frac{\eta}{2}|\boldsymbol{\varepsilon}|_F^2. \quad \square$$

Despite those strong properties of  $\psi(\cdot, d)$  we note that  $\psi(\boldsymbol{\varepsilon}, d)$  is not convex in  $d$  and  $\boldsymbol{\varepsilon}$  together for any of the splittings considered below.

<sup>3</sup> Under the additional assumption that  $0 = \psi(0, d) \leq \psi(\boldsymbol{\varepsilon}, d)$  holds for all  $\boldsymbol{\varepsilon}$  and  $d$  one can show that (P4) implies (P5).

### 2.1.1 Isotropic splitting

In this model, any strain will lead to damage. The splitting is therefore

$$\psi_0^+(\boldsymbol{\varepsilon}) = \psi_0(\boldsymbol{\varepsilon}), \quad \psi_0^-(\boldsymbol{\varepsilon}) = 0. \tag{6}$$

Without proof, we note the following simple properties of the energy density  $\psi$  defined by (5) and this splitting:

**Lemma 2.6** *The energy density  $\psi$  defined in (5) with the isotropic splitting (6) has the properties (P1)–(P5). Furthermore  $\psi(\cdot, d)$  has the stronger property that it is in  $C^\infty$  and quadratic for all  $d \in [0, 1]$ .*

### 2.1.2 Volumetric decompositions

The isotropic model is of only limited use, because it produces fracturing for all kinds of strain. In [31], Lancioni and Royer-Carfagn obtained better results by letting only the deviatoric strain contribute to the degradation. They introduced the split

$$\psi_0^+(\boldsymbol{\varepsilon}) = \psi_0(\text{dev } \boldsymbol{\varepsilon}), \quad \psi_0^-(\boldsymbol{\varepsilon}) = \psi_0(\text{vol } \boldsymbol{\varepsilon}),$$

with the deviatoric–volumetric strain splitting

$$\text{vol } \boldsymbol{\varepsilon} := \frac{\text{tr } \boldsymbol{\varepsilon}}{m} I, \quad \text{dev } \boldsymbol{\varepsilon} := \boldsymbol{\varepsilon} - \text{vol } \boldsymbol{\varepsilon}.$$

With these definitions, the energies are

$$\begin{aligned} \psi_0^+(\boldsymbol{\varepsilon}) &= \left(\frac{\mu}{m} + \frac{\lambda}{2}\right)(\text{tr } \boldsymbol{\varepsilon})^2, \\ \psi_0^-(\boldsymbol{\varepsilon}) &= \mu \left(\boldsymbol{\varepsilon}^2 - \frac{1}{m}(\text{tr } \boldsymbol{\varepsilon})^2\right) = \mu \text{ dev } \boldsymbol{\varepsilon} : \text{ dev } \boldsymbol{\varepsilon}. \end{aligned} \tag{7}$$

**Lemma 2.7** *The energy density  $\psi$  defined in (5) with the isotropic volumetric splitting (7) has the properties (P1)–(P5). Furthermore  $\psi(\cdot, d)$  has the stronger property that it is in  $C^\infty$  and quadratic for all  $d \in [0, 1]$ .*

**Proof**  $C^\infty$ -smoothness and thus (P1) and (P2) are straightforward. The fact that  $\psi_0^+$  and  $\psi_0^-$  are quadratic, convex, and non-negative allows to derive (P3), (P4), and (P5) from Lemma 2.5 and implies that  $\psi(\cdot, d)$  is also quadratic.  $\square$

The decomposition of Lancioni and Royer-Carfagni is still isotropic. [4] proposed to only degrade the expansive part of the volumetric strain. Using the ramp functions

$$\langle x \rangle_+ := \max\{0, x\}, \quad \langle x \rangle_- := \min\{0, x\}$$

that provide the decompositions  $x = \langle x \rangle_+ + \langle x \rangle_-$  and  $x^2 = \langle x \rangle_+^2 + \langle x \rangle_-^2$ , they proposed the energy split

$$\begin{aligned} \psi_0^+(\boldsymbol{\varepsilon}) &= \left(\frac{\mu}{m} + \frac{\lambda}{2}\right) \langle \text{tr } \boldsymbol{\varepsilon} \rangle_+^2, \\ \psi_0^-(\boldsymbol{\varepsilon}) &= \left(\frac{\mu}{m} + \frac{\lambda}{2}\right) \langle \text{tr } \boldsymbol{\varepsilon} \rangle_-^2 + \mu \text{ dev } \boldsymbol{\varepsilon} : \text{ dev } \boldsymbol{\varepsilon}, \end{aligned} \tag{8}$$

where only the tensile volumetric strain contributes to damage.

**Lemma 2.8** *The energy density  $\psi$  defined in (5) with the anisotropic volumetric splitting (8) has the properties (P1)–(P5). Furthermore  $\psi(\cdot, d)$  is not  $C^2$ , unless  $g(d) = 1$ .*

**Proof** We first note that the squared ramp functions  $\langle \cdot \rangle_\pm^2$  are convex,  $C^{1,1}$  with derivatives having a global Lipschitz constant 2, and piecewise  $C^2$  (in the sense of [51, Definition 2.19]). Hence the functions  $\psi_0^\pm$  are also  $C^{1,1}$  with globally Lipschitz gradients and piecewise  $C^2$ , which shows (P1) and (using Lemma 2.5) (P3). Being piecewise  $C^2$  implies semismoothness (P2) of  $\nabla \psi(\cdot, d)$  [51, Proposition 2.26]. Noting that  $\mu/m + \lambda/2 > 0$ , convexity of the squared ramp functions furthermore implies that the functions  $\psi_0^\pm$  are also convex and non-negative, which by Lemma 2.5 provides (P4) and (P5).

For  $g(d) + k = 1$  the functional  $\psi(\cdot, d)$  is quadratic and thus  $C^2$ . In the case  $g(d) \neq 1$ , if  $\psi(\cdot, d)$  would be  $C^2$ , then the function  $t \mapsto \psi(tI, d)$  would also be  $C^2$ . However, this function takes the form

$$\psi(tI, d) = \left(\frac{\mu}{m} + \frac{\lambda}{2}\right) m^2 t^2 \begin{cases} g(d) + k & \text{if } t \geq 0, \\ 1 + k & \text{if } t < 0 \end{cases}$$

and is thus piecewise quadratic but not  $C^2$  in  $t = 0$ .  $\square$

### 2.1.3 Spectral decomposition

A more elaborate nonlinear splitting separating the tensile and compressive parts of the elastic energy was introduced in [38]. To define this splitting it is convenient to introduce the ordered eigenvalue function  $\text{Eig} : \mathbb{S}^m \rightarrow \mathbb{R}^m$  on the space  $\mathbb{S}^m$  of symmetric  $m \times m$  matrices, mapping any symmetric matrix  $M$  to the vector  $\text{Eig}(M) \in \mathbb{R}^m$  containing its eigenvalues in ascending order. Using the ramp functions the tensile and compressive energies  $\psi_0^+$  and  $\psi_0^-$  are then defined as

$$\psi_0^\pm(\boldsymbol{\varepsilon}) := \frac{\lambda}{2} \left\langle \sum_{i=1}^m \text{Eig}(\boldsymbol{\varepsilon})_i \right\rangle_\pm^2 + \mu \sum_{i=1}^m \langle \text{Eig}(\boldsymbol{\varepsilon})_i \rangle_\pm^2. \tag{9}$$

Note that this indeed defines a splitting  $\psi_0 = \psi_0^+ + \psi_0^-$ . For this splitting we will make the additional assumption that  $\lambda \geq 0$ .

To quantify the properties of  $\psi(\boldsymbol{\varepsilon}, d)$  with respect to the strain tensor  $\boldsymbol{\varepsilon}$  we use the theory of spectral functions [32]. To this end we note that we can write  $\psi_0^\pm$  as

$$\psi_0^\pm = \widehat{\psi}_0^\pm \circ \text{Eig} : \mathbb{S}^m \rightarrow \mathbb{R}$$

with

$$\widehat{\psi}_0^\pm(\boldsymbol{\lambda}) := \frac{\lambda}{2} \left\langle \sum_{i=1}^m \lambda_i \right\rangle_\pm^2 + \mu \sum_{i=1}^m \langle \lambda_i \rangle_\pm^2.$$

The functions  $\widehat{\psi}_0^\pm$  are symmetric in the sense that  $\widehat{\psi}_0^\pm(\boldsymbol{\lambda})$  does not depend on the order of the entries of  $\boldsymbol{\lambda} \in \mathbb{R}^m$ . Having this form we can infer properties of the functions  $\psi_0^\pm = \widehat{\psi}_0^\pm \circ \text{Eig}$  from properties of the symmetric functions  $\widehat{\psi}_0^\pm$ .

**Lemma 2.9** *Let  $\lambda \geq 0$ . Then the energy density  $\psi$  defined in (5) with the spectral splitting (9) has the properties (P1)–(P5). Furthermore  $\psi(\cdot, d)$  is not  $C^2$ , unless  $g(d) = 1$ .*

**Proof** We will first show (P1)–(P5). An essential ingredient is that the squared ramp functions  $\langle \cdot \rangle_\pm^2$  are non-negative, piecewise quadratic, and convex.

(P1) The squared ramp functions  $\langle \cdot \rangle_\pm^2$  and thus  $\widehat{\psi}_0^\pm$  are  $C^{1,1}$ . Now [44, Proposition 4.3] shows that the spectral functions  $\psi_0^\pm = \widehat{\psi}_0^\pm \circ \text{Eig}$  are also  $C^{1,1}$ . Hence the same applies to  $\psi(\cdot, d)$ .

(P2) The squared ramp functions  $\langle \cdot \rangle_\pm^2$  are piecewise  $C^2$  functions. Hence the gradients  $\nabla \widehat{\psi}_0^\pm$  are piecewise  $C^1$  functions (in the sense of [51, Definition 2.19]) and thus semismooth [51, Proposition 2.26]. Now [44, Proposition 4.5] provides semismoothness of  $\nabla \psi_0^\pm$  and thus of  $\nabla \psi(\cdot, d)$ .

(P3) Since the functions  $\widehat{\psi}_0^\pm$  are piecewise quadratic and  $C^{1,1}$  the gradients  $\nabla \widehat{\psi}_0^\pm$  are globally Lipschitz continuous. Now Corollary 43 of [5] provides global Lipschitz continuity of the gradients  $\nabla \psi_0^\pm$  of the spectral functions  $\psi_0^\pm$  in the more general context of Euclidean Jordan algebras (which includes the special case of symmetric matrices). In fact, the Lipschitz constant of  $\nabla \widehat{\psi}_0^\pm$  equals the one for  $\nabla \psi_0^\pm$  if  $\mathbb{S}^m$  is equipped with the Frobenius norm. Using Lemma 2.5 this implies uniform Lipschitz continuity of  $\psi(\cdot, d)$ .

(P4),(P5) Since the functions  $\widehat{\psi}_0^\pm$  are weighted sums of convex, non-negative squared ramp functions with non-negative weights, they are convex and non-negative themselves. Convexity of the functions  $\psi_0^\pm$  then follows from [5, Theorem 41] while non-negativity of those functions is trivial. Now Lemma 2.5 provides (P4) and (P5).

To characterize second order differentiability of  $\psi(\cdot, d)$  we first consider  $g(d) = 1$ . Then  $\psi(\cdot, d)$  coincides with the quadratic function  $(1+k)\psi_0 = (1+k)\psi_0^+ + (1+k)\psi_0^-$  and is thus  $C^2$ . In the case  $g(d) \neq 1$ , if  $\psi(\cdot, d)$  would be  $C^2$ , then the function  $t \mapsto \psi(tE, d)$  for the fixed matrix  $E$  with  $E_{ij} = \delta_{1i}\delta_{1j}$  would also be  $C^2$ . However, this function takes

the form

$$\psi(tE, d) = \left(\frac{\lambda}{2} + 1\right)t^2 \begin{cases} g(d) + k & \text{if } t \geq 0, \\ 1 + k & \text{if } t < 0, \end{cases}$$

and is thus piecewise quadratic but not  $C^2$  in  $t = 0$ .  $\square$

**Remark 2.10** One can show that  $\mathbb{S}^m$  decomposes into finitely many disjoint subsets  $\mathcal{A}_i$  such that  $\psi(\cdot, d)$  is twice continuously differentiable in the interior of each of these sets. A matrix  $\boldsymbol{\varepsilon} \in \mathbb{S}^m$  is in the intersection of several  $\overline{\mathcal{A}}_i$  if it either has an eigenvalue  $\text{Eig}(\boldsymbol{\varepsilon})_i = 0$  or if  $\text{tr } \boldsymbol{\varepsilon} = 0$ . While  $\nabla \psi(\cdot, d)$  is not differentiable at those points, there are still generalized second-order derivatives. For example, the generalized Jacobian in the sense of Clarke contains the derivatives of  $\nabla \psi(\cdot, d)$  with respect to all the adjacent sets  $\mathcal{A}_i$ . Semismoothness essentially means that such generalized derivatives provide an approximation that can be exploited in a generalized Newton method.

**Remark 2.11** The additional assumption  $\lambda \geq 0$  is essential for convexity of  $\psi(\cdot, d)$ . To see this we consider for  $m = 2$  the line segment

$$\{D(t) = \text{diag}(-1, t) \mid t \in (0, 1)\} \subset \mathbb{S}^2.$$

Then, along this line segment,  $\psi(\cdot, 1)$  is quadratic and takes the form

$$\begin{aligned} \psi(D(t), 1) &= k\mu t^2 + (1+k) \left(\frac{\lambda}{2}(t-1)^2 + \mu\right) \\ &= \left(k\mu + (1+k)\frac{\lambda}{2}\right)t^2 + (1+k)\left(-\lambda t + \frac{\lambda}{2} + \mu\right) \end{aligned}$$

which is strictly concave for  $\lambda < 0$  and sufficiently small  $k \ll 1$ .

## 2.2 Crack surface density

The crack surface density function per unit volume of the solid is typically of the form [43]

$$\gamma(d, \nabla d) := \frac{1}{4c_w} \left( \frac{w(d)}{l} + l|\nabla d|^2 \right),$$

with parameters  $c_w$  and  $l$ , and a parameter function  $w : [0, 1] \rightarrow [0, 1]$ . Motivated by the seminal work of Modica and Mortola [40, 41], the associated crack surface functional

$$\mathcal{C}(d) := \int_{\Omega} \gamma(d, \nabla d) dx$$

is called a Modica–Mortola functional. The internal length scale parameter  $l$  controls the size of the diffusive zone

between a completely intact and a completely damaged material. For  $l \rightarrow 0$  the regularized crack surface yields a sharp crack topology in the sense of  $\Gamma$ -convergence [40, 41]. For a given function  $w$ , the normalization constant  $c_w$  must be chosen such that the integral of  $\gamma(d, \nabla d)$  over the fractured domain converges to the surface measure of the crack set as  $l \rightarrow 0$ .

The function  $w(d)$  models the local fracture energy. Two types of local crack density functions appear in the literature. Double-well potentials (as briefly reviewed in [2]) provide an energy barrier between broken and unbroken state, but will be disregarded here. Instead, we only consider single-well potentials  $w$ , which grow monotonically from the intact state  $w(0) = 0$  to the damaged state  $w(1) = 1$ . For such potentials the normalization constant is given by

$$c_w := \int_0^1 \sqrt{w(s)} ds. \tag{10}$$

While this follows from the  $\Gamma$ -convergence proof (see, e.g., [1]), the proper constant can also be computed by elementary means: Consider an open set  $\Gamma \subset \mathbb{R}^{m-1}$  of measure  $|\Gamma|$  embedded in a domain  $\Omega = \mathbb{R} \times \Gamma$ . We identify  $\Gamma$  with the set  $\{x \in \Omega : x_1 = 0\}$ , and interpret it as a crack in  $\Omega$ . To approximate  $\Gamma$  by a damage field  $d : \Omega \rightarrow [0, 1]$ , assume that  $d$  minimizes the crack surface energy

$$\mathcal{C}(d) := \int_{\Omega} \gamma(d, \nabla d) dx,$$

subject to the crack conditions  $d(0, \xi) = 1$  and  $d(\pm\infty, \xi) = 0$  for all  $\xi \in \Gamma$ . We find that  $d$  is given by  $d(x) = \hat{d}(|x_1|/l)$  where  $\hat{d} : [0, \infty] \rightarrow [0, 1]$  solves the normalized scalar Euler–Lagrange equation

$$w'(\hat{d}) - 2\hat{d}'' = 0 \quad \text{in } (0, \infty),$$

with boundary condition  $\hat{d}(0) = 1$ , or, equivalently, the initial value problem,

$$\hat{d}'(s) = -\sqrt{w(\hat{d}(s))} \quad s > 0, \quad \hat{d}(0) = 1. \tag{11}$$

Using (11) we find that the crack surface energy of the minimizer  $d$  is

$$\begin{aligned} \mathcal{C}(d) &= \frac{2|\Gamma|}{4c_w} \int_0^\infty w(\hat{d}) + |\hat{d}'|^2 ds \\ &= -\frac{2|\Gamma|}{4c_w} \int_0^\infty 2\sqrt{w(\hat{d})}\hat{d}' ds \\ &= \frac{|\Gamma|}{c_w} \int_0^1 \sqrt{w(s)} ds. \end{aligned}$$

Thus  $c_w$  has to be selected as in (10) to ensure that the crack surface energy scales like the limit crack surface measure  $|\Gamma|$ .

Note that the function  $\hat{d}$  is the  $w$ -dependent crack profile, rescaled by the length parameter  $l$ . In order to relate the length parameter to the actual crack width, we use the cone construction from [37] and define the crack width to be the average width of a tangential finite cone fitted to the crack profile and the zero function (Fig. 1). From the initial condition  $\hat{d}(0) = 1$ , the crack profile equation (11), and the normalization  $w(1) = 1$  it follows that  $\hat{d}'(0) = -1$ . Thus—for the normalized solution  $\hat{d}$ —the cone has width 2 at its base and average width 1. Hence the average crack cone width of the rescaled solution  $d$  is  $l$ .

We will focus on the two widely used potentials

$$w(d) = d, \quad c_w = \frac{2}{3}$$

and

$$w(d) = d^2, \quad c_w = \frac{1}{2}.$$

They are referred to in the literature as Ambrosio–Tortorelli (AT) functionals of type 1 and 2, respectively. The corresponding crack profiles are given by

$$\hat{d}_{\text{AT-1}}(s) = \begin{cases} (1 - \frac{s}{2})^2 & \text{if } s < 2, \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{d}_{\text{AT-2}}(s) = \exp(-s).$$

Some authors like [30] prefer  $w(d) = d^2$  because it has a local minimizer at  $d = 0$ . Thus, in the absence of mechanical strain, the unfractured solution  $d \equiv 0$  is a minimizer of the total energy. As a result, no additional constraints need to be applied to ensure that  $d \geq 0$ . However, this argument becomes void when solver technology is available that can handle the explicit constraints  $0 \leq d \leq 1$ . In contrast, for the AT-1 functional we have  $w' \neq 0$  in the intact state  $d = 0$ . Together with the constraint  $d \in [0, 1]$  this leads to a threshold, i.e., a minimum load required to cause damage [43]. A numerical comparison of the AT-1 and AT-2 functionals can be found in [12].

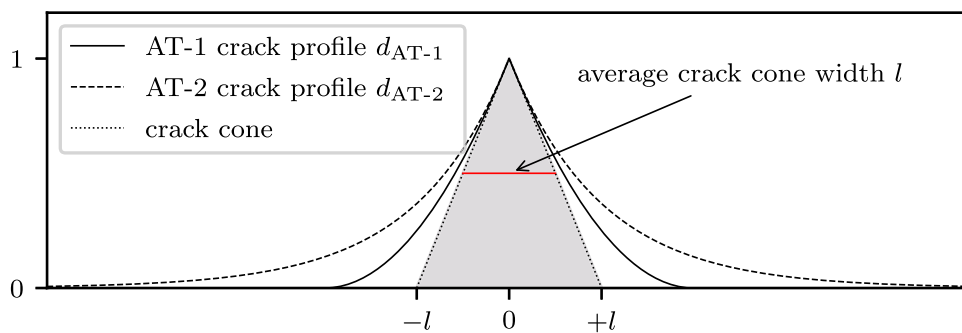
Kuhn et al. [30] proposed to regard the Ambrosio–Tortorelli functionals as special instances of the general family defined by

$$w(d) = (1 + \beta(1 - d))d, \tag{12}$$

with  $\beta \in [-1, 1]$ . The Ambrosio–Tortorelli functionals are obtained by setting  $\beta = 0$  for AT-1 and  $\beta = -1$  for AT-2. Fur-



**Fig. 1** Crack width definition for the AT functionals



ther choices of  $w$  are proposed in [43], which also contains a detailed stability analysis for one-dimensional problems.

We note the following properties of the functional  $w$  in (12):

**Lemma 2.12** *The function  $w$  given in (12) has the following properties:*

- (1) It fulfills  $w(0) = 0$  and  $w(1) = 1$ .
- (2) It is strictly monotone increasing on  $[0, 1]$  for all  $\beta \in [-1, 1]$ .
- (3) It is convex for all  $\beta \leq 0$ , and strictly convex for all  $\beta < 0$ .

For the rest of the paper we will assume the  $w(\cdot)$  takes the form (12) with  $\beta \leq 0$  such that  $w(\cdot)$  is guaranteed to be convex and quadratic.

### 3 Discretization and the algebraic increment potential

We use a fully implicit discretization in time, and Lagrange finite elements for discretization in space. By using a fully implicit time discretization we retain the variational structure of the problem. Most of this section is spent investigating the properties of the increment functional.

#### 3.1 Time discretization

It is shown in [39] that there is a natural time discretization for (1) that consists of sequences of minimization problems. To simplify the presentation we will not derive the time discretization from an energetic formulation of the time-dependent problem (cf. Remark 2.1) but first discretize the variational formulation (4) and then reformulate the time-discrete problem as a sequence of minimization problems. Let the time interval  $[0, T]$  be subdivided by time points  $t_n$ ,  $n = 0, 1, 2, \dots$  and denote by  $(\mathbf{u}_n, d_n) \in \mathbf{H}_{\mathbf{u}_0} \times H_{d_0}$  the discrete approximation of  $(\mathbf{u}(t_n), d(t_n))$ . Approximating the time derivative  $\dot{d}(t_{n+1})$  by the backward difference quotient  $(d_{n+1} - d_n)/\tau_n$  for the time step size  $\tau_n := t_{n+1} - t_n$ , and

inserting this into the time-continuous variational inequality (4) we obtain the time-discrete variational inequality for  $(\mathbf{u}_{n+1}, d_{n+1})$

$$\begin{aligned} & \left\langle D(\mathbf{u}_{n+1}, d_{n+1}) \mathcal{E}(t_{n+1}, \mathbf{u}_{n+1}, d_{n+1}), \right. \\ & \left. (\mathbf{v}, e) - (\mathbf{u}_{n+1}, (d_{n+1} - d_n)/\tau_n) \right\rangle \\ & + \mathcal{R}(e) - \mathcal{R}((d_{n+1} - d_n)/\tau_n) \geq 0 \\ & \forall (\mathbf{v}, e) \in \mathbf{H}_{\mathbf{u}_0}^1 \times H_0^1. \end{aligned}$$

Testing with

$$(\mathbf{v}, e) = \left( \mathbf{u}_{n+1} + \frac{1}{\tau_n}(\hat{\mathbf{v}} - \mathbf{u}_{n+1}), \frac{1}{\tau_n}(\hat{e} - d_n) \right)$$

for  $(\hat{\mathbf{v}}, \hat{e}) \in \mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1$ , using the fact that  $\mathcal{R}$  is 1-homogeneous, and relabeling  $(\hat{\mathbf{v}}, \hat{e})$  to  $(\mathbf{v}, e)$  yields

$$\begin{aligned} & \left\langle D(\mathbf{u}_{n+1}, d_{n+1}) \mathcal{E}(t_{n+1}, \mathbf{u}_{n+1}, d_{n+1}), (\mathbf{v}, e) - (\mathbf{u}_{n+1}, d_{n+1}) \right\rangle \\ & + \mathcal{R}(e - d_n) - \mathcal{R}(d_{n+1} - d_n) \geq 0 \quad \forall (\mathbf{v}, e) \in \mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1. \end{aligned} \tag{13}$$

This is the first-order optimality system for the minimization problem

$$(\mathbf{u}_{n+1}, d_{n+1}) := \arg \min_{(\tilde{\mathbf{u}}, \tilde{d}) \in \mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1} \Pi_{n+1}^\tau(\tilde{\mathbf{u}}, \tilde{d}), \tag{14}$$

with the increment potential

$$\begin{aligned} \Pi_{n+1}^\tau(\mathbf{u}, d) & := \mathcal{E}(t_{n+1}, \mathbf{u}, d) + \mathcal{R}(d - d_n) \\ & = \int_{\Omega} \psi(\boldsymbol{\varepsilon}(\mathbf{u}), d) dV \\ & \quad + \int_{\Omega} g_c \gamma(d, \nabla d) dV + P_{\text{ext}}(t_{n+1}, \mathbf{u}) \\ & \quad + \int_{\Omega} I_{[d_n, 1]}(d) dV. \end{aligned}$$

Although the variational inequality (13) is not equivalent to the minimization problem (14) because  $\mathcal{E}$  is not convex, we will use the minimization formulation in the following.

Note that the time step size does not appear in this functional, which means that the model is rate-independent. Note also that the increment potential depends on the previous time step only through the indicator functional.

Next we investigate the properties—notably the semicontinuity—of the increment potential  $\Pi_{n+1}^\tau$ . While related results for Ambrosio–Tortorelli-type functionals have already been shown in the seminal work [3], we give proofs here for exactly the range of functionals considered in Sect. 2.

**Lemma 3.1** *Assume that  $\Gamma_{D,\mathbf{u}}$  is non-trivial in the sense that its  $m - 1$ -dimensional Hausdorff-measure is positive. Then the functional  $\Pi_{n+1}^\tau$  is coercive on  $\mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1$ .*

**Proof** Using the uniform coercivity (P5) of  $\psi(\cdot, d)$ ,  $w(1) > 0$ , and  $w(d) \geq 0$  for  $d \in [0, 1]$  we get

$$\int_{\Omega} \psi(\boldsymbol{\varepsilon}(\mathbf{u}), d) dV + \int_{\Omega} g_c \gamma(d, \nabla d) dV \geq C \int_{\Omega} |\boldsymbol{\varepsilon}(\mathbf{u})|_F^2 + |\nabla d|^2 dV$$

for some constant  $C > 0$ . Using Korn’s inequality for  $\mathbf{u}$ , the Poincaré inequality for  $d$ , and the fact that  $P_{\text{ext}}(t_{n+1}, \mathbf{u})$  grows at most linearly we get for another constant  $C > 0$

$$\begin{aligned} \Pi_{n+1}^\tau(\mathbf{u}, d) &\geq C \left( \|\mathbf{u}\|_1^2 + \|d\|_1^2 - 1 - \left( \int_{\Omega} d dV \right)^2 \right) \\ &\quad + \int_{\Omega} I_{[d_n, 1]}(d) dV \\ &\geq C \left( \|\mathbf{u}\|_1^2 + \|d\|_1^2 - 1 - |\Omega|^2 \right), \end{aligned}$$

where we have used that the constraint  $d \in [0, 1]$  implies  $|\int_{\Omega} d dV| \leq |\Omega|$  in the second inequality.  $\square$

**Lemma 3.2** *The functional  $\Pi_{n+1}^\tau$  is weakly lower semicontinuous on  $\mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1$ .*

**Proof** Since weak lower semicontinuity of the other terms in  $\Pi_{n+1}^\tau$  follows from convexity and lower semicontinuity of the integrands, we only need to consider the non-convex term

$$\int_{\Omega} \psi(\boldsymbol{\varepsilon}(\mathbf{u}), d) + I_{[d_n, 1]}(d) dV. \tag{15}$$

To this end we note that (15) can be written as  $J(\mathbf{u}, d, \nabla \mathbf{u})$  for

$$J(\mathbf{u}, d, \xi) := \int_{\Omega} F(x, (\mathbf{u}(x), d(x)), \xi) dV$$

and the density  $F : \Omega \times (\mathbb{R}^m \times \mathbb{R}) \times \mathbb{R}^{m \times m} \rightarrow \mathbb{R} \cup \{\infty\}$  given by

$$F(x, (\mathbf{u}, d), \xi) = \psi\left(\frac{1}{2}(\xi + \xi^T), d\right) + I_{[d_n, 1]}(d).$$

Since  $F$  is a Carathéodory function, non-negative (and thus uniformly bounded from below), and convex in  $\xi$  for all  $(x, (\mathbf{u}, d)) \in \Omega \times (\mathbb{R}^m \times \mathbb{R})$ , it satisfies the assumptions of Theorem 3.4 in [14].

Now let  $(\mathbf{u}^\nu, d^\nu) \rightharpoonup (\mathbf{u}, d)$  be a weakly convergent sequence in  $\mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1$ . Then, by the compactness of the embedding into  $L^2(\Omega, \mathbb{R}^m \times \mathbb{R})$  we get

$$(\mathbf{u}^\nu, d^\nu) \rightarrow (\mathbf{u}, d) \quad \text{in } L^2(\Omega, \mathbb{R}^m \times \mathbb{R}).$$

Furthermore, the  $H^1(\Omega, \mathbb{R}^m \times \mathbb{R})$ -weak convergence of  $(\mathbf{u}^\nu, d^\nu)$  implies  $L^2(\Omega, \mathbb{R}^{m \times m})$ -weak convergence of  $\nabla \mathbf{u}^\nu$

$$\xi^\nu := \nabla \mathbf{u}^\nu \rightharpoonup \nabla \mathbf{u} =: \xi \quad \text{in } L^2(\Omega, \mathbb{R}^{m \times m}),$$

because  $(\mathbf{u}, d) \mapsto \eta(\nabla \mathbf{u})$  is in  $H^1(\Omega, \mathbb{R}^m \times \mathbb{R})'$  for each  $\eta \in L^2(\Omega, \mathbb{R}^{m \times m})'$ . Now Theorem 3.4 of [14] provides

$$\liminf_{\nu \rightarrow \infty} J(\mathbf{u}^\nu, d^\nu, \xi^\nu) \geq J(\mathbf{u}, d, \xi) = J(\mathbf{u}, d, \nabla \mathbf{u}). \quad \square$$

As a direct consequence of coercivity and weak lower semicontinuity we get existence of a minimizer of the increment functional:

**Theorem 3.3** *There is a solution to the minimization problem (14), i.e., there exists a global minimizer  $(\mathbf{u}_{n+1}, d_{n+1}) \in \mathbf{H}_{\mathbf{u}_0}^1 \times H_{d_0}^1$  of  $\Pi_{n+1}^\tau$ .*

### 3.2 Finite element discretization

The increment problem (14) of the previous section is posed on the pair of spaces  $\mathbf{H}_{\mathbf{u}_0}^1$  for the displacements and  $H_{d_0}^1$  for the damage variable. Let  $\mathcal{G}$  be a conforming finite element grid for  $\Omega$ . We discretize the function spaces by standard first-order Lagrangian finite elements. In order to derive an algebraic form of the discretized increment functional we make use of the standard scalar nodal basis  $\{\theta_i\}_{i=1}^M$  associated to the grid nodes  $\{p_1, \dots, p_M\} =: \mathcal{N} \subset \Omega$ . Identifying the  $\mathbb{R}^m$ -valued and scalar finite element functions  $\mathbf{u}$  and  $d$  with their coefficient vectors  $\mathbf{u} \in \mathbb{R}^{M, m}$  and  $d \in \mathbb{R}^M$ , respectively, we write

$$\mathbf{u}_j = \sum_{i=1}^M u_{i,j} \theta_i, \quad d = \sum_{i=1}^M d_i \theta_i,$$

where  $u_{i,j} = \mathbf{u}_j(p_i)$  and  $d_i = d(p_i)$ . For the integration we use three kinds of quadrature rules: Integrals of smooth nonlinear terms over a grid element  $e$  are approximated using a higher-order quadrature rule  $\int_{e,h} dV$ , while the integral over the nonsmooth term  $I_{[d_n, 1]}(d)$  is approximated using the grid nodes  $p_i$  as quadrature points, which is often referred to

as *lumping*. All polynomial terms are integrated exactly using appropriate quadrature rules. Using these approximations we obtain the algebraic increment functional  $\mathcal{J} := \Pi_{n+1}^{\tau, \mathcal{G}}$  given by

$$\begin{aligned} \mathcal{J}(\mathbf{u}, d) := & \underbrace{\int_{\Omega, h} (\psi(\boldsymbol{\varepsilon}(\mathbf{u}), d)) dV + \int_{\Omega} g_c \gamma(d, \nabla d) dV + P_{\text{ext}}(t_{n+1}, \mathbf{u})}_{=: \mathcal{J}_0(\mathbf{u}, d)} \\ & + \underbrace{\sum_{i=1}^M I_{[d_n(p_i), 1]}(d_i)}_{=: \varphi(d)}. \end{aligned} \tag{16}$$

Here the quadrature rule  $\int_{\Omega, h} (\cdot) dV$  is given by

$$\begin{aligned} \int_{\Omega, h} f dV &:= \sum_{e \in \mathcal{G}} \int_{e, h} f dV, \\ \int_{e, h} f dV &:= \sum_{\alpha=1}^{\alpha_{\max}} f(q_{e, \alpha}) \omega_{e, \alpha} \end{aligned}$$

with positive weights  $\omega_{e, \alpha}$  on each element  $e$ . Notice that we do not need quadrature weights in the last term of  $\mathcal{J}$ , because the indicator function only takes values in  $\{0, \infty\}$ .

To elucidate the algebraic structure of  $\mathcal{J}$  we introduce the linear operator  $\mathcal{L} : (\mathbb{R}^{M, m} \times \mathbb{R}^M) \rightarrow ((\mathbb{S}^m \times \mathbb{R})^{\alpha_{\max}})^{\mathcal{G}}$  with

$$\begin{aligned} \mathcal{L}(\mathbf{u}, d)_{e, \alpha} &:= ((\boldsymbol{\varepsilon}(\mathbf{u})) (q_{e, \alpha}), d(q_{e, \alpha})) \\ \alpha &= 1, \dots, \alpha_{\max}, e \in \mathcal{G}. \end{aligned}$$

Then the first part  $\mathcal{J}_0$  of the functional can be written as

$$\begin{aligned} \mathcal{J}_0(\mathbf{u}, d) &= \underbrace{\sum_{e \in \mathcal{G}} \sum_{\alpha=1}^{\alpha_{\max}} \psi(\mathcal{L}(\mathbf{u}, d)_{e, \alpha}) \omega_{e, \alpha}}_{=: A(\mathbf{u}, d)} \\ &+ \underbrace{\int_{\Omega} g_c \gamma(d, \nabla d) dV + P_{\text{ext}}(t_{n+1}, \mathbf{u})}_{=: B(\mathbf{u}, d)}. \end{aligned}$$

Note that for the price of a more complex index notation, the linear operator  $\mathcal{L}$  can also be written as a sparse matrix with suitable blocking structure. In this case  $\mathcal{L}(\cdot, \cdot)_{e, \alpha} : (\mathbb{R}^{M, m} \times \mathbb{R}^M) \rightarrow (\mathbb{S}^m \times \mathbb{R})$  corresponds to the  $(e, \alpha)$ -th sparse row of this matrix.

As an approximation of the boundary conditions from  $H_{\mathbf{u}_0}^1 \times H_{d_0}^1$  we will consider  $\mathcal{J}$  on the affine subspace  $H^{\text{alg}} = H_{\mathbf{u}_0}^{\text{alg}} \times H_{d_0}^{\text{alg}}$  where

$$\begin{aligned} H_{\mathbf{u}_0}^{\text{alg}} &:= \{ \mathbf{u} \in \mathbb{R}^{M, m} \mid \mathbf{u}(p) = \mathbf{u}_0(p) \quad \forall p \in \mathcal{N} \cap \Gamma_{D, \mathbf{u}} \}, \\ H_{d_0}^{\text{alg}} &:= \{ d \in \mathbb{R}^M \mid d(p) = d_0(p) \quad \forall p \in \mathcal{N} \cap \Gamma_{D, d} \}. \end{aligned}$$

The associated homogeneous subspace of  $H^{\text{alg}}$  is denoted by  $H_0^{\text{alg}}$ . In the following we make the assumption that

$\mathcal{N} \cap \Gamma_{D, \mathbf{u}}$  contains the vertices of at least one boundary grid face, which ensures a discrete Korn inequality such that  $\|\boldsymbol{\varepsilon}(\mathbf{u})\|_0 \geq C \|\mathbf{u}\|_1$  holds for all  $\mathbf{u} \in H_{\mathbf{u}_0}^{\text{alg}}$ . Furthermore we introduce the discrete feasible set

$$\mathcal{K}^{\text{alg}} := H_{\mathbf{u}_0}^{\text{alg}} \times \left( H_{d_0}^{\text{alg}} \cap \mathcal{K}_d^{\text{alg}} \right), \quad \mathcal{K}_d^{\text{alg}} := \prod_{i=1}^M [d_n(p_i), 1]$$

that additionally incorporates the pointwise irreversibility constraints.

### 3.3 Properties of the discrete incremental potential

The convergence property of the TNNMG algorithm heavily rely on the algebraic structure of the problem. Hence we now collect the essential structural properties of the algebraic increment functional  $\mathcal{J}$ . While stronger properties hold true for some splittings of  $\psi$ , we only note the necessary properties shared by all of the proposed splittings. In order to preserve the significant properties in the presence of numerical quadrature, we assume that the quadrature rule  $\int_{e, h} f dV$  can at least integrate the isotropic energy  $f = |\boldsymbol{\varepsilon}(\mathbf{u})|_F^2$  exactly for any finite element function  $\mathbf{u}$ .

**Lemma 3.4** *The functional  $\mathcal{J}_0 = A + B$  has the following properties:*

- (1)  $\mathcal{J}_0(\cdot, d) \in C^{1,1}$  and  $\mathcal{J}_0(\mathbf{u}, \cdot) \in C^2$  for any  $d \in \mathcal{K}_d^{\text{alg}}$  and  $\mathbf{u} \in \mathbb{R}^{M, m}$ .
- (2) The gradient  $\nabla \mathcal{J}_0(\cdot, d)$  is semismooth for any  $d \in \mathcal{K}_d^{\text{alg}}$ .
- (3) The gradient  $\nabla \mathcal{J}_0(\cdot, d)$  is globally Lipschitz continuous uniformly in  $d$ .
- (4)  $\mathcal{J}_0(\cdot, d)$  is strongly convex uniformly in  $d$  on  $H_{\mathbf{u}_0}^{\text{alg}}$ .
- (5)  $\mathcal{J}_0(\mathbf{u}, \cdot)$  is convex on  $\mathcal{K}_d^{\text{alg}}$  for any  $\mathbf{u} \in \mathbb{R}^{M, m}$ .

**Proof** The smoothness properties, uniform global Lipschitz continuity, and convexity follow from the corresponding properties of  $\psi$  and  $\gamma$ , and from linearity of  $\mathcal{L}$ .

To see uniform strong convexity, we note that uniform strong convexity of  $\psi(\cdot, d)$  implies that there is some  $\eta > 0$  such that  $\phi(\boldsymbol{\varepsilon}, d) := \psi(\boldsymbol{\varepsilon}, d) - \frac{\eta}{2} |\boldsymbol{\varepsilon}|_F^2$  is convex. Using the exactness assumption on the quadrature rule we get

$$\sum_{e \in \mathcal{G}} \sum_{\alpha=1}^{\alpha_{\max}} \phi(\mathcal{L}(\mathbf{u}, d)_{e, \alpha}) \omega_{e, \alpha} = A(\mathbf{u}, d) - \frac{\eta}{2} \|\boldsymbol{\varepsilon}(\mathbf{u})\|_0^2.$$

Since this is a weighted sum of convex functions  $\phi(\cdot, d(q_{e, \alpha}))$  with positive weights  $\omega_{e, \alpha}$ , it is itself convex with respect to  $\mathbf{u}$ . Thus  $A(\mathbf{u}, d)$  is the sum of a convex function and the function  $\|\boldsymbol{\varepsilon}(\mathbf{u})\|_0^2$ . Since the latter is strongly convex on  $H_{\mathbf{u}_0}^{\text{alg}}$  independently of  $d$ , the same applies to  $A(\mathbf{u}, d)$  and  $(A + B)(\cdot, d)$ .

Finally we note that convexity of  $g$  and  $\gamma$  imply convexity of  $(A + B)(\mathbf{u}, \cdot)$ .  $\square$

The TNNMG algorithm is based on a crucial property called block-separability, which states that the nonsmooth part of the objective functional can be written as a sum, such that the sets of independent variables of the addend functionals are disjoint. We note that  $\mathcal{J} = \mathcal{J}_0 + \varphi$  is of the desired form, with a smooth part  $\mathcal{J}_0 = A + B$  and a block-separable nonsmooth part

$$\varphi(d) := \sum_{i=1}^M \varphi_i(d_i), \quad \varphi_i(\xi) := I_{[d_n(p_i), 1]}(\xi), \quad (17)$$

which can also be written as the indicator functional  $\varphi(d) = I_{\mathcal{K}_d^{\text{alg}}}(d)$  of the feasible set  $\mathcal{K}_d^{\text{alg}}$  of the  $n + 1$ -th time step.

Due to the nonsmoothness of  $\varphi$ , the smoothness properties of  $\mathcal{J}_0$  do obviously not carry over to the full functional  $\mathcal{J}$ . Furthermore  $\mathcal{J}$  is in general not convex as a whole. However we still have the following:

**Lemma 3.5** *The functional  $\mathcal{J}$  is proper, lower semicontinuous, and coercive on  $H^{\text{alg}}$ . Furthermore it is convex in  $\mathbf{u}$  and convex in  $d$ .*

**Proof** Being the indicator function of the closed, nonempty, convex set  $\mathcal{K}_d^{\text{alg}}$  it is clear that the separable nonsmooth functional  $\varphi$  is convex, proper, and lower semicontinuous. Combining this with smoothness of  $\mathcal{J}_0$  we get that  $\mathcal{J}$  is proper and lower semicontinuous. Similarly, convexity in  $\mathbf{u}$  and  $d$  follows from the corresponding properties of  $\mathcal{J}_0$  and  $\varphi$ .

Using the uniform coercivity (P5) of  $\psi(\cdot, d)$ ,  $w(1) > 0$ , and  $w(d) \geq 0$  (as in the proof of Lemma 3.1) and the exactness assumption on the quadrature rule (as in the proof of Lemma 3.4) we get

$$\begin{aligned} A(\mathbf{u}, d) + B(\mathbf{u}, d) - P_{\text{ext}}(t_{n+1}, \mathbf{u}) \\ \geq C \int_{\Omega} |\boldsymbol{\varepsilon}(\mathbf{u})|_F^2 + |\nabla d|^2 dV \end{aligned}$$

for some constant  $C > 0$ . Now we can proceed as in the proof of Lemma 3.1 to show coercivity of  $\mathcal{J}$ .  $\square$

#### 4 Truncated nonsmooth Newton multigrid for brittle fracture

The truncated nonsmooth Newton multigrid method (TNNMG) is designed to solve nonsmooth block-separable minimization problems on Euclidean spaces. In a nutshell, one step of the TNNMG method consists of a nonlinear Gauß–Seidel-type smoother and a subsequent inexact

Newton-type correction in a constrained subspace. The nonlinear smoother computes local corrections by subsequent (possibly inexact) solving of reduced minimization problems in small subspaces. As the nonlinear smoother is responsible for ensuring convergence, while the Newton corrections accelerate the convergence, the ingredients of the nonlinear smoother have to be selected carefully.

It is a well known result [19] that nonsmooth Gauß–Seidel-type methods can easily get stuck if the subspace decomposition used to construct localized minimization problems is not aligned with the decomposition induced by the block-separable nonsmooth term. In our case, the nonsmooth term  $\varphi$  is separable with respect to the decomposition of unknowns induced by the grid vertices. An additional requirement is that the local minimization problems must be uniquely solvable, which is typically ensured by choosing the decomposition such that the local problems are strictly convex.

In view of these requirements we first decompose the space according to the grid vertices and then with respect to the local  $\mathbf{u}$ - and  $d$ -degrees of freedom leading to a decomposition

$$\mathbb{R}^{M,m} \times \mathbb{R}^M = (\mathbb{R}^m \times \mathbb{R})^M = \sum_{j=1}^M (V_{j,\mathbf{u}} + V_{j,d}). \quad (18)$$

Here the  $m$ -dimensional subspace  $V_{j,\mathbf{u}}$  represents the displacement components at the  $j$ -th grid vertex, while the one-dimensional subspace  $V_{j,d}$  represents the  $d$ -component at this vertex. All other components are set to zero in these spaces such that the subspace decomposition can be written as a direct sum. For simplicity we use a plain enumeration of these subspaces in alternating order

$$V_{2j-1} = V_{j,\mathbf{u}}, \quad V_{2j} = V_{j,d}, \quad j = 1, \dots, M. \quad (19)$$

Notice that with this splitting none of the nonsmooth terms  $\varphi_i$  in (17) couples across different subspaces. Furthermore, by Lemma 3.4 the restriction of  $\mathcal{J}$  to any affine subspace  $(\mathbf{u}, d) + V_i$ ,  $i = 1, \dots, 2M$  is convex.

We will now introduce the TNNMG method. For simplicity we first assume that  $\Gamma_{D,\mathbf{u}}$  and  $\Gamma_{D,d}$  are empty and that  $\mathcal{J}_0$  is  $C^2$ . Let  $\nu \in \mathbb{N}_0$  denote the iteration number. Given a previous iterate  $\mathbf{U}^\nu = (\mathbf{u}, d)^\nu \in \mathbb{R}^{M,m} \times \mathbb{R}^M$ , one iteration of the TNNMG method consists of the following four steps:

##### (1) Nonlinear presmoothing

- (a) Set  $\mathbf{W}^0 = \mathbf{U}^\nu$
- (b) For  $i = 1, \dots, 2M$  compute  $\mathbf{W}^i \in \mathbf{W}^{i-1} + V_i$  as

$$\mathbf{W}^i \approx \arg \min_{\mathbf{W} \in \mathbf{W}^{i-1} + V_i} \mathcal{J}(\mathbf{W}) \quad (20)$$

(c) Set  $U^{v+\frac{1}{2}} = W^{2M}$

(2) **Inexact linear correction**

- (a) Determine the maximal subspace  $W_v \subset \mathbb{R}^{M,m} \times \mathbb{R}^M$  such that the restriction  $\mathcal{J}|_{W_v}$  is  $C^2$  at  $U^{v+\frac{1}{2}}$
- (b) Compute  $c^v \in W_v$  as an inexact Newton step on  $W_v$

$$c^v \approx -(\mathcal{J}''(U^{v+\frac{1}{2}})|_{W_v \times W_v})^{-1}(\mathcal{J}'(U^{v+\frac{1}{2}})|_{W_v}) \tag{21}$$

(3) **Projection**

Compute the Euclidean projection  $c_{pr}^v = P_{\text{dom } \mathcal{J}-U^{v+1/2}} \times (c^v)$ , i.e., choose  $c_{pr}^v$  such that  $U^{v+\frac{1}{2}} + c_{pr}^v$  is closest to  $U^{v+\frac{1}{2}} + c^v$  in  $\text{dom } \mathcal{J}$

(4) **Damped update**

- (a) Compute a  $\rho_v \in [0, \infty)$  such that  $\mathcal{J}(U^{v+\frac{1}{2}} + \rho_v c_{pr}^v) \leq \mathcal{J}(U^{v+\frac{1}{2}})$
- (b) Set  $(u, d)^{v+1} = U^{v+1} = U^{v+\frac{1}{2}} + \rho_v c_{pr}^v$

The algorithm is easily generalized to non-trivial Dirichlet boundary conditions by leaving out all subspaces associated to Dirichlet vertices during the nonlinear smoothing, and by additionally requiring  $W_v \subset H_0^{\text{alg}}$  for the linear correction subspace. Then, if the initial iterate satisfies the boundary conditions, i.e., if  $(u_d)^0 \in H^{\text{alg}}$ , the method will only iterate within this affine subspace, which preserves the Dirichlet boundary conditions for all iterates.

The canonical choice for the linear correction step (21) is a single linear multigrid step, which explains why the overall method is classified as a multigrid method. If a grid hierarchy is available, then a geometric multigrid method is preferable. Otherwise, a suitable constructed algebraic multigrid step for small-strain elasticity problems will work just as well. In the case of the phase-field brittle-fracture increment functional considered here, the nonlinear presmoothing step takes a large part of the run-time of a single iteration. The convergence speed can therefore be improved considerably by doing a small fixed number (larger than 1) of multigrid steps, without appreciably increasing the time per iteration.

Section 4.1 will discuss convergence of the method based on an abstract convergence theory. The abstract theory will be used as a guideline for the discussion of nonlinear smoothers in Sect. 4.2. Finally 4.3 will discuss the linear correction in more detail.

**4.1 Convergence results**

The TNNMG method was originally introduced for convex problems where global convergence to global minimizers can be shown [20, 21, 23]. These classical results cannot be applied here, due to the non-convexity of  $\mathcal{J}$ . As a generalization of previous results, [22] introduced an abstract

convergence theory that also covers non-convex problems. In the following we will summarize some results from this work. These will later be used as a guideline for specifying how to solve the local subproblems (20) and the linear correction problem (21). In order to simplify the presentation some of the terminology and notation used in [22] is avoided in favor of a more specific notation adjusted to the algorithm as introduced above.

**Theorem 4.1** *Let  $\mathcal{J} : \mathbb{R}^L \rightarrow \mathbb{R} \cup \{\infty\}$  be coercive, proper, lower semicontinuous, and continuous on its domain, and assume that  $\mathcal{J}(V + (\cdot))$  has a unique global minimizer in  $V_i$  for all  $i$  and each  $V \in \text{dom } \mathcal{J}$ . Assume that the inexact local corrections  $W^i$  are given by  $W^i = \mathcal{M}_i(W^{i-1})$  for local correction operators*

$$\mathcal{M}_i : \text{dom } \mathcal{J} \rightarrow \text{dom } \mathcal{J}, \quad \mathcal{M}_i - \text{Id} : \text{dom } \mathcal{J} \rightarrow V_i$$

having the properties:

- (1) *Monotonicity:*  $\mathcal{J}(\mathcal{M}_i(V)) \leq \mathcal{J}(V)$  for all  $V \in \text{dom } \mathcal{J}$ .
- (2) *Continuity:*  $\mathcal{J} \circ \mathcal{M}_i$  is continuous.
- (3) *Stability:*  $\mathcal{J}(\mathcal{M}_i(V)) < \mathcal{J}(V)$  if  $\mathcal{J}(V)$  is not minimal in  $V + V_i$ .

Furthermore assume that the initial iterate is feasible, i.e.,  $U^0 \in \text{dom } \mathcal{J}$ , and that the linear correction is monotone, i.e.,  $\mathcal{J}(U^{v+1}) \leq \mathcal{J}(U^{v+\frac{1}{2}})$ . Then any accumulation point  $U$  of  $(U^v)$  is stationary in the sense that

$$\mathcal{J}(U) \leq \mathcal{J}(U + V) \quad \forall V \in V_i, \quad \forall i. \tag{22}$$

**Proof** This is Theorem 4.1 in [22]. □

**Remark 4.2** Note that the statement requires both global lower semicontinuity of  $\mathcal{J}$  and continuity on its domain, because neither property implies the other. In the proof given in [22], lower semicontinuity is needed to show that accumulation points of the iteration have finite energy and are thus feasible, while continuity of  $\mathcal{J}$  on its domain is needed to show that they are stationary.

Now we discuss the application of this theorem to the phase-field brittle-fracture problem. First we interpret the stationarity result.

**Proposition 4.3** *Let  $\mathcal{J}$  be given by (16) and the subspaces  $V_i$  by (18) and (19). Then any stationary point  $U$  in the sense of (22) is first-order optimal in the sense of*

$$\langle \nabla \mathcal{J}_0(U), W - U \rangle \leq 0 \quad \forall W \in \text{dom } \mathcal{J}.$$

**Proof** The stationarity (22) implies the variational inequalities

$$\langle \nabla \mathcal{J}_0(\mathbf{U}), \mathbf{W}_i - \mathbf{U} \rangle \leq 0 \quad \forall \mathbf{W}_i \in \text{dom } \mathcal{J} \cap (\mathbf{U} + V_i)$$

for each subspace  $V_i$ . Now let  $\mathbf{W} \in \text{dom } \mathcal{J}$ . Since the splitting (18) is direct one can splitting  $\mathbf{W}$  uniquely into

$$\mathbf{W} = \mathbf{U} + \sum_{i=1}^{2M} \mathbf{V}_i, \quad \mathbf{V}_i \in V_i.$$

Using the product structure of  $\text{dom } \mathcal{J}$  we find that  $\mathbf{W}_i := \mathbf{U} + \mathbf{V}_i \in \text{dom } \mathcal{J} \cap (\mathbf{U} + V_i)$ . Summing up the variational inequalities for those  $\mathbf{W}_i$  we obtain the assertion.  $\square$

Next we investigate the assumptions of the theorem. First we note that  $\mathcal{J}$  as given in (16) is coercive, proper, and lower semicontinuous by Lemma 3.5. Furthermore,  $\mathcal{J}_0$  is continuous and the indicator function  $\varphi$  is continuous on its domain. Hence the latter is also true for  $\mathcal{J} = \mathcal{J}_0 + \varphi$ . Subspaces  $V_i$  with odd index  $i$  only vary in  $\mathbf{u}(p_i)$  such that existence of a unique minimizer of  $\mathcal{J}(\mathbf{W} + (\cdot))|_{V_i}$  follows from the strong convexity of  $\mathcal{J}(\cdot, d)$  shown in Lemma 3.4. For even  $i$  these subspaces are associated to nodal damage degrees of freedom  $d(p_i)$ . Although  $\mathcal{J}(\mathbf{u}, \cdot)$  is in general only convex, but not strictly convex, the restriction  $\mathcal{J}(\mathbf{W} + (\cdot))|_{V_i}$  to a single node is a strictly convex quadratic functional, which again implies existence of a unique minimizer. Finally the monotonicity  $\mathcal{J}(\mathbf{U}^{\nu+1}) \leq \mathcal{J}(\mathbf{U}^{\nu+\frac{1}{2}})$  of the linear correction is a direct consequence of the damped update.

It remains to identify proper local correction operators  $\mathcal{M}_i$  satisfying the above assumptions. As a first result we show that solving the local minimization problems (20) exactly leads to a convergent algorithm in the sense given above.

**Lemma 4.4** *Let  $\mathcal{J}$  be given by (16) and the subspaces  $V_i$  by (18) and (19). Then the exact local solution operators*

$$\mathcal{M}_i(\mathbf{W}) := \arg \min_{\mathbf{V} \in \mathbf{W} + V_i} \mathcal{J}(\mathbf{V})$$

*satisfy the assumptions of Theorem 4.1.*

**Proof** This is Lemma 5.1 in [22].  $\square$

Depending on the damaged energy density  $\psi$ , solving the restricted problems exactly may not be practical. As a remedy, it is also shown in [22] that inexact minimization is sufficient as long as it guarantees sufficient decrease of the energy. In fact we do not need  $\mathbf{W}^i = \mathcal{M}_i(\mathbf{W}^{i-1})$  exactly but may relax this to  $\mathcal{J}(\mathbf{W}^i) \leq \mathcal{J}(\mathcal{M}_i(\mathbf{W}^{i-1}))$  for a suitable continuous  $\mathcal{M}_i$ . However, sufficient decrease is in general hard to check rigorously. In the following we cite one inexact variant from [22] where sufficient descent is guaranteed a priori.

**Lemma 4.5** *Let  $\mathcal{J}$  be given by (16) and the subspaces  $V_i$  by (18) and (19). For each subspace  $V_i$  let  $C_i$  be a symmetric positive definite matrix that satisfies*

$$\begin{aligned} \langle \nabla \mathcal{J}_0(\mathbf{W} + \mathbf{V}) - \nabla \mathcal{J}_0(\mathbf{W}), \mathbf{V} \rangle &\leq \langle C_i \mathbf{V}, \mathbf{V} \rangle \\ \forall \mathbf{W} \in \text{dom } \mathcal{J}, \mathbf{V} \in V_i. \end{aligned}$$

*Then the correction operators*

$$\begin{aligned} \mathcal{M}_i(\mathbf{W}) := &\arg \min_{\mathbf{V} \in (\mathbf{W} + V_i) \cap \text{dom } \mathcal{J}} \mathcal{J}(\mathbf{W}) + \langle \nabla \mathcal{J}_0(\mathbf{W}), \mathbf{V} - \mathbf{W} \rangle \\ &+ \frac{1}{2} \langle C_i (\mathbf{V} - \mathbf{W}), \mathbf{V} - \mathbf{W} \rangle \end{aligned}$$

*satisfy the assumptions of Theorem 4.1.*

**Proof** This is Lemma 5.8 in [22].  $\square$

## 4.2 Smoothers for brittle fracture problems

The smoother of the TNNMG method performs a sequence of (inexact) minimization problems in low-dimensional subspaces  $V_i$ . Different approaches are possible here, implementing different compromises between convergence speed, wall-time per iteration, and ease of programming. As there are two types of degrees of freedom, two types of solvers are needed as well.

### 4.2.1 Subspaces of displacement degrees of freedom

We first consider the subspaces  $V_i = V_{(i+1)/2, \mathbf{u}}$  for odd  $i$  spanned by the  $m$  displacement degrees of freedom at the vertex  $p_{(i+1)/2}$ . Noting that elements of this subspace only vary in the displacement component, the minimization problem (20) is equivalent to

$$\arg \min_{(\mathbf{v}, 0) \in V_i} L_i(\mathbf{v}). \quad (23)$$

Here, the restricted functional  $L_i(\mathbf{v}) := \mathcal{J}(\mathbf{W}^{i-1} + (\mathbf{v}, 0))$  takes the form

$$\begin{aligned} L_i(\mathbf{v}) = &\int_{\Omega, h} (g(d) + k) \psi_0^+(\boldsymbol{\varepsilon}(\mathbf{u} + \mathbf{v})) \\ &+ (1 + k) \psi_0^-(\boldsymbol{\varepsilon}(\mathbf{u} + \mathbf{v})) dV + P_{\text{ext}}(t_{n+1}, \mathbf{v}) + \text{const}, \end{aligned} \quad (24)$$

where we have used  $(\mathbf{u}, d) = \mathbf{W}^{i-1}$ . The precise nature of  $L_i$  depends on the type of energy splitting used by the model. If the isotropic splittings (6) or (7) are used, (24) is a strictly convex quadratic functional on a vector space, and can be minimized exactly by solving an  $m \times m$  system of linear equations.

For the anisotropic splittings (8) and (9), the functional is still strictly convex and once continuously differentiable. The classical Hesse matrix, however, is not guaranteed to exist. However, by Lemma 3.4, the increment functional is semismooth. This suggests various natural choices for local solvers, such as steepest-descent methods or nonsmooth Newton methods [51]. When these are used to solve the local problems (20) exactly, global convergence of the overall TNNMG method follows from Theorem 4.1 and Lemma 4.4 above.

However, as mentioned in the previous section, Theorem 4.1 is more general, and also shows convergence for certain types of inexact local solvers (such as the one in Lemma 4.5). Such a setup can make iterations much faster, while keeping the corresponding deterioration of the convergence rate within acceptable limits. Possible approaches are:

- One Newton-type step where the  $\psi''$ -term in the Hessian  $\mathcal{J}_0''$  of the differentiable part  $\mathcal{J}_0$  of  $\mathcal{J}$  is replaced by the quadratic upper bound  $(1 + k)\psi_0''$ , that is, the undamaged St. Venant–Kirchhoff energy density (scaled with  $(1 + k)$ ), which is strictly convex and quadratic. By the construction of the splittings in Sect. 2.1, this term bounds the degraded energy density (5) for any admissible value of  $d$ . We call this approach a preconditioned smoother.
- One (or another fixed number of) semismooth Newton steps.
- One gradient step with exact line search.

For the first variant, global convergence of the TNNMG solver follows from Lemma 4.5. For the other two, the problem of showing convergence is open. Section 5 will show how the first two choices perform in practice.

### 4.2.2 Subspaces spanned by damage degrees of freedom

For subspaces  $V_i = V_{i/2,d}$  with even  $i$ , i.e., subspaces spanned by the damage degrees of freedom at the vertices  $p_{i/2}$ , the minimization problem (20) is equivalent to

$$\arg \min_{(0,v) \in V_i} L_i(v),$$

with a restricted functional  $L_i(v) := \mathcal{J}(W^{i-1} + (0, v))$ . For all choices of damage functions  $g$  described in Sect. 2 this is a strictly convex quadratic functional on a closed interval, whose minimizer can be computed directly by computing the unconstrained minimizer and projecting onto the admissible interval. We therefore always assume that these problems are solved exactly.

### 4.3 Linear multigrid corrections

For the linear correction step (21) we need to compute a constrained Newton-type correction

$$c^v \approx -(\mathcal{J}''(U^{v+\frac{1}{2}})|_{W_v \times W_v})^{-1} (\mathcal{J}'(U^{v+\frac{1}{2}})|_{W_v}) \tag{25}$$

at least inexactly. This requires to determine the subspace  $W_v$ , to compute the constrained first- and second-order derivatives  $\mathcal{J}'(U^{v+\frac{1}{2}})|_{W_v}$  and  $\mathcal{J}''(U^{v+\frac{1}{2}})|_{W_v \times W_v}$  on this subspace, and finally to solve the system inexactly.

It is easy to see that the largest subspace  $W_v$  where  $\mathcal{J}$  is differentiable in a neighborhood of  $U^{v+\frac{1}{2}} = (u^{v+\frac{1}{2}}, d^{v+\frac{1}{2}})$  is given by

$$W_v = \left\{ U = (u, d) \mid d_i = 0 \text{ if } (d^{v+\frac{1}{2}})_i \notin (d_n(p_i), 1) \right\}.$$

In this subspace the nonsmooth indicator functional  $\varphi$  is identical to zero such that we only need to compute first- and second-order derivatives of the smooth part  $\mathcal{J}_0$ , which are then restricted to the degrees of freedom that are allowed to be nonzero in  $W_v$ . This can easily be achieved by setting rows and columns of  $\mathcal{J}'$  and  $\mathcal{J}''$  to zero for degrees of freedom not contained in  $W_v$ . For all splittings where the degraded density  $\psi$  is not  $C^2$ , it is at least locally Lipschitz and semismooth. In this case a generalized second-order derivative  $\mathcal{J}_0''(U^{v+\frac{1}{2}})$  can be used as a replacement of the classical Hesse matrix making (21) a semismooth Newton step. Such a generalized Hessian can be obtained by the following procedure: The density  $\psi$  is piecewise  $C^2$ , and every point  $(\epsilon, d)$  where it is not  $C^2$  is at the boundary of several subdomains on which  $\psi$  is  $C^2$ . Whenever the second derivative  $\psi''$  needs to be computed at such a point, one uses instead the second derivative from any of these adjacent subdomains.

To practically compute the first and second derivatives of the degraded elastic energy with the splitting (9) based on eigenvalues, recall that the positive and negative parts  $\psi_0^\pm$  of that splitting are spectral functions

$$\psi_0^\pm = \widehat{\psi}_0^\pm \circ \text{Eig} : \mathbb{S}^m \rightarrow \mathbb{R},$$

where  $\text{Eig} : \mathbb{S}^m \rightarrow \mathbb{R}^m$  is the ordered eigenvalue function, and the  $\widehat{\psi}_0^\pm : \mathbb{R}^m \rightarrow \mathbb{R}$  are invariant under permutations of their arguments. Such functions are called *spectral*, and they are once or twice differentiable if and only if the functions  $\widehat{\psi}_0^\pm$  are once or twice differentiable, respectively. The first and second derivatives of  $\psi_0^\pm$  can be expressed in closed form in terms of the derivatives of  $\widehat{\psi}_0^\pm$ . This is shown, for example, in [33, Lemma 3.1] for first-order derivatives, and in [33, Theorem 3.3] for second-order ones. The expressions involve computing the eigenvector decomposition of the argument of  $\psi_0^\pm$ .

For the inexact solution (25) one step of a classical linear multigrid method can be used. Here we only need to take care that the linear smoother can deal with the non-trivial kernel resulting from constraining the linearization. For a linear Gauß–Seidel smoother this amounts to omitting corrections for rows with zero diagonal entry. When using the TNNMG method for problems with a nonquadratic smooth part such as the phase-field brittle-fracture problem considered here, the smoother is relatively expensive. One can then improve the overall convergence rate by doing more than one multigrid iteration, without increasing the time per iteration.

## 5 Numerical examples

In this last section we demonstrate the speed and robustness of the TNNMG solver with two numerical examples. For both of them we study four instances from the family of models discussed in this manuscript: the isotropic and the spectral splittings ((6) and (9), respectively), combined with the AT-1 and AT-2 crack density functionals ( $w(d) = d$  and  $w(d) = d^2$ , respectively).

In the experiments we will consider two variants of the TNNMG method with two different nonlinear smoothers, both based on the splitting (19) which alternates displacement and damage degrees of freedom. For the first variant—denoted TNNMG-EX in the following—the smoother will solve the local displacement problems (23) inexactly in the sense that one damped (generalized) Newton step is applied. In the second variant—denoted TNNMG-PRE—the smoother will solve approximate local displacement problems exactly. To this end it employs a single Newton-like step where the  $\psi''$ -term in the Hessian  $\mathcal{J}_0''$  of the differentiable part  $\mathcal{J}_0$  of  $\mathcal{J}$  is replaced by the quadratic upper bound  $(1+k)\psi_0''$ .

The constrained quadratic local damage problems are solved exactly by both smoothers. By Lemma 4.5, the TNNMG-PRE smoother satisfies the assumptions of the convergence Theorem 4.1, but the TNNMG-EX smoother does so only for the degraded elastic energy density without a splitting (otherwise its local solution operator is not continuous). For the linear correction step (21) we use three standard  $V(3, 3)$  linear geometric multigrid steps with a block Gauß–Seidel smoother operating on the canonical  $(d+1) \times (d+1)$  blocks. The coarse linear problems of the multigrid iteration are solved using the UMFPack sparse direct solver [15]. Damage degrees of freedom are truncated when they are less than  $10^{-10}$  away from their lower bound. The TNNMG iteration is set to run until the relative degraded energy norm of the correction drops below  $10^{-7}$ . The degraded energy norm for the coupled problem combines the energy norm of the

degraded linear elasticity problem with the energy norm of the AT-2 crack surface energy density.<sup>4</sup>

We measure iteration numbers, wall-time, and memory consumption. The TNNMG algorithm and the operator-splitting algorithm used for comparison are implemented in C++ using the DUNE libraries<sup>5</sup> [6, 46].

### 5.1 An operator splitting method for phase-field brittle fracture models

We compare the performance of the TNNMG method to an operator-splitting method from the literature. This method is described here in some detail to allow readers to reproduce the results. We choose an operator-splitting method because such methods are widely used, and reported to be robust [2, 57].

Starting from an initial displacement  $\mathbf{u}^0$  and damage field  $d^0$  the operator-splitting method alternately repeats the following steps:

$$\mathbf{u}^{v+1} = \arg \min_{\mathbf{u} \in H_{\mathbf{u}_0}^{\text{alg}}} \mathcal{J}(\mathbf{u}, d^v) \quad (26)$$

$$d^{v+1} = \arg \min_{d \in H_{d_0}^{\text{alg}}} \mathcal{J}(\mathbf{u}^{v+1}, d). \quad (27)$$

The iteration is terminated under the same condition as the TNNMG method above. For simplicity we omit any modifications such as proposed, e.g., in [10].

If the degraded elastic energy  $\psi(\boldsymbol{\varepsilon}, d)$  is of the type (6), which does not distinguish between compressive and tensile strains, then (26) is a linear problem with a symmetric positive definite matrix. Our implementation solves these problems using the CHOLMOD direct sparse solver [13].<sup>6</sup>

If the degraded elastic energy incorporates a compressive–tensile split such as (9), then  $\mathcal{J}$  is not a quadratic functional in  $\mathbf{u}$ . In this case, as suggested by [37, Equation (66)], we perform a single Newton step at  $(\mathbf{u}^v, d^v)$ , viz.

$$\mathbf{u}^{v+1} = \mathbf{u}^v - \left( \nabla_{\mathbf{u}\mathbf{u}}^2 \mathcal{J}(\mathbf{u}^v, d^v) \right)^{-1} \nabla_{\mathbf{u}} \mathcal{J}(\mathbf{u}^v, d^v),$$

where  $\nabla_{\mathbf{u}} \mathcal{J}$  and  $\nabla_{\mathbf{u}\mathbf{u}}^2 \mathcal{J}$  are the first and second derivatives of  $\mathcal{J}$ , respectively, with respect to  $\mathbf{u}$  only. Because of the splitting, the Hesse matrix  $\nabla_{\mathbf{u}\mathbf{u}}^2 \mathcal{J}$  does not depend continuously on  $\mathbf{u}$ . At points where an entry of  $\nabla_{\mathbf{u}\mathbf{u}}^2 \mathcal{J}$  jumps, we

<sup>4</sup> Because the AT-1 model does not induce a norm.

<sup>5</sup> [www.dune-project.org](http://www.dune-project.org)

<sup>6</sup> It is well-known that sparse direct solvers do not scale well to larger problems both run-time and memory-wise. Therefore, an alternative method such as linear multigrid may be a better choice here. However, direct solvers are widely used in practice, and we therefore consider a comparison with such a solver worthwhile.



simply pick one of the one-sided limits, which are elements of the generalized Jacobian of  $\nabla_{\mathbf{u}}\mathcal{J}$  in the sense of Clarke (cf. Remark 2.10).

For the AT-1 and AT-2 models, the damage subproblem (27) is a quadratic minimization problem subject to lower bound constraints

$$d_i^v \leq d_i \leq 1 \quad i = 1, \dots, M.$$

The Hesse matrix is

$$\begin{aligned} \left(\nabla_{dd}^2 \mathcal{J}(\mathbf{u}^{v+1}, d)\right)_{ij} &= \int_{\Omega} \left[ g''(d) \theta_i \theta_j \psi_0^+(\boldsymbol{\varepsilon}(\mathbf{u}^{v+1})) \right. \\ &\quad \left. + \underbrace{\frac{g_c}{2c_w l} \theta_i \theta_j}_{\text{AT-2 only}} + \frac{g_c l}{2c_w} \nabla \theta_i \nabla \theta_j \right] dx, \end{aligned}$$

with  $g''(d) = 2$  for our choice  $g(d) = (1 - d)^2$ . For the AT-2 model this Hesse matrix is always positive definite. For the AT-1 model, it is only positive definite if the tensile elastic energy  $\psi_0^+$  is positive everywhere, and positive semidefinite otherwise. This did not lead to any problems in the numerical tests done for this article. If necessary, a small amount of  $L^2$ -regularization can be added to increase the robustness.

Following a suggestion by [12], we use a projected Newton method to solve the constrained damage problems. We use the variant proposed by [7], because it is well described and, for the quadratic problems considered here, converges locally quadratically [7, Proposition 4]. The method combines projected gradient steps for degrees of freedom in the vicinity of the constraints with Newton-type steps for the other degrees of freedom, and an Armijo-style line search. Consult the original article [7] of Bertsekas for a detailed description. In the notation of that article, we let  $M$  be the identity matrix and  $D_k$  ( $k$  being the iteration number) the truncated Hesse matrix. This matrix results from the true Hesse matrix  $\nabla_{dd}^2 \mathcal{J}(\mathbf{u}^{v+1}, d)$  by replacing the rows and columns for which the degrees of freedom are close to or at the constraint by the corresponding rows and columns of the identity matrix.<sup>7</sup> The maximum truncation distance is  $\varepsilon = 10^{-5}$ . For the line search parameters we set  $\beta = 0.5$  and  $\sigma = 0.49$ . We solve the linear subproblems with the CHOLMOD solver.

Note that the projected Newton method is closely related to the TNNMG method. Indeed, the TNNMG method could also be applied to solve only the damage problem (27) within an operator splitting loop. The main differences are that TNNMG has a presmoothing, and that it does not solve the linear correction problems exactly.

<sup>7</sup> Note that this way of truncating degrees of freedom differs from the one employed by the TNNMG method (the construction of the space  $W_v$ ) in Sect. 4.3.

## 5.2 Pure tension test of a notched, symmetric specimen

The first numerical example is a two-dimensional, square-shaped notched specimen of size  $L \times L$  under tension. Due to symmetry we simulate only its upper half. Geometry and boundary conditions are shown in Fig. 2. On the top edge a time-dependent normal displacement  $\bar{u}$  is prescribed, while the horizontal displacement is left free. The bottom edge of the upper half is clamped vertically for all  $x > L/2$ , and fixed vertically and horizontally at the single point  $(L/2, 0)$ , which is where the initial crack tip is. With increasing normal displacement  $\bar{u}$ , the preexisting crack opens, and the specimen ruptures suddenly, when a limit load is exceeded. By symmetry, the crack energy is accounted for correctly, even though only one half of the crack profile appears in the simulation result.

The simulations are performed with parameters taken from the corresponding experiment in [38], viz.  $L = 1$  mm, Lamé parameters  $\lambda = 121$  kN/mm<sup>2</sup> and  $\mu = 80$  kN/mm<sup>2</sup>, critical energy release rate  $g_c = 2.7$  N/mm, and residual stiffness  $k = 10^{-5}$ . The phase-field regularization parameter is set to  $l = 0.03125$  mm. We apply the loading in 160 steps, and set the displacement load  $\bar{u}$  at step  $i$  to

$$\bar{u}_i = i \cdot 2 \cdot 10^{-5} \text{ mm} \cdot \mathbf{e}_2, \quad i = 1, \dots, 160,$$

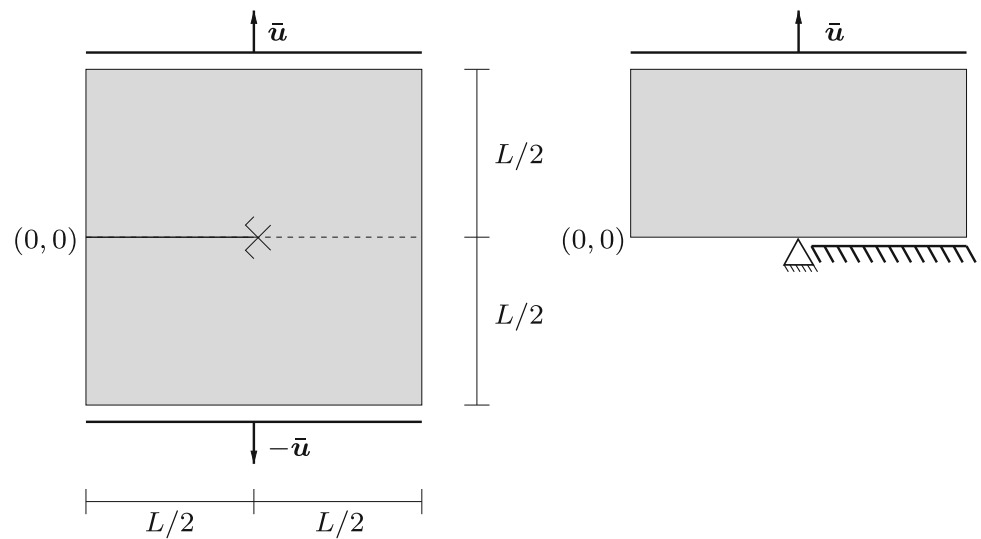
where  $\mathbf{e}_2$  is the canonical basis vector pointing upwards. We start the evolution with no displacement and no damage anywhere.

For the spatial discretization we use three different uniform grids with  $256 \times 128$  ( $h_1$ ),  $512 \times 256$  ( $h_2$ ), and  $1024 \times 512$  ( $h_3$ ) quadrilateral elements, respectively. These were all constructed by uniform refinement of a grid with  $32 \times 16$  elements, and hence the grid hierarchy for the multi-grid solver consists of 4, 5, and 6 levels, respectively. To separate the effect of the discretization parameter  $h$  from the modeling parameter  $l$ , we explicitly study different element sizes  $h$  to illustrate the performance of the algorithm, instead of choosing a single element size proportional to the crack width. Relating the grid resolution to the average fracture width  $l$ , the average element edge length corresponds to  $l/8$ ,  $l/16$ , and  $l/32$ , respectively.

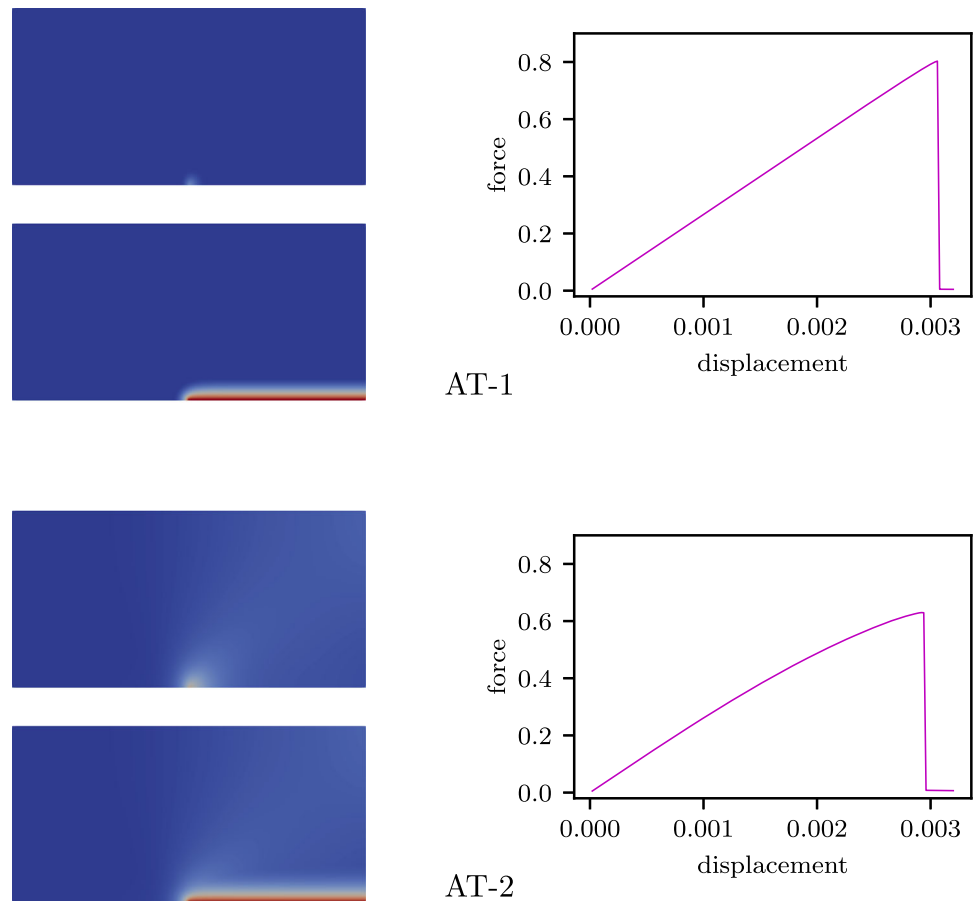
### 5.2.1 Isotropic splitting

We first consider the model with the isotropic splitting (6) of the elastic energy density, where all elastic strains contribute to the degradation of the material. Figure 3 shows the evolution and displacement–force curves, both for the AT-1 and AT-2 functionals. The force plotted here is the total normal force on the top edge of the specimen.

**Fig. 2** Notched square with a vertical displacement load (left). Exploiting symmetry we only simulate on the upper half of the domain (right)



**Fig. 3** 2d example, isotropic energy split: evolution at time steps 145 and 160 computed with the TNNMG algorithm, and displacement–force curves for the AT-1 (top) and AT-2 (bottom) models

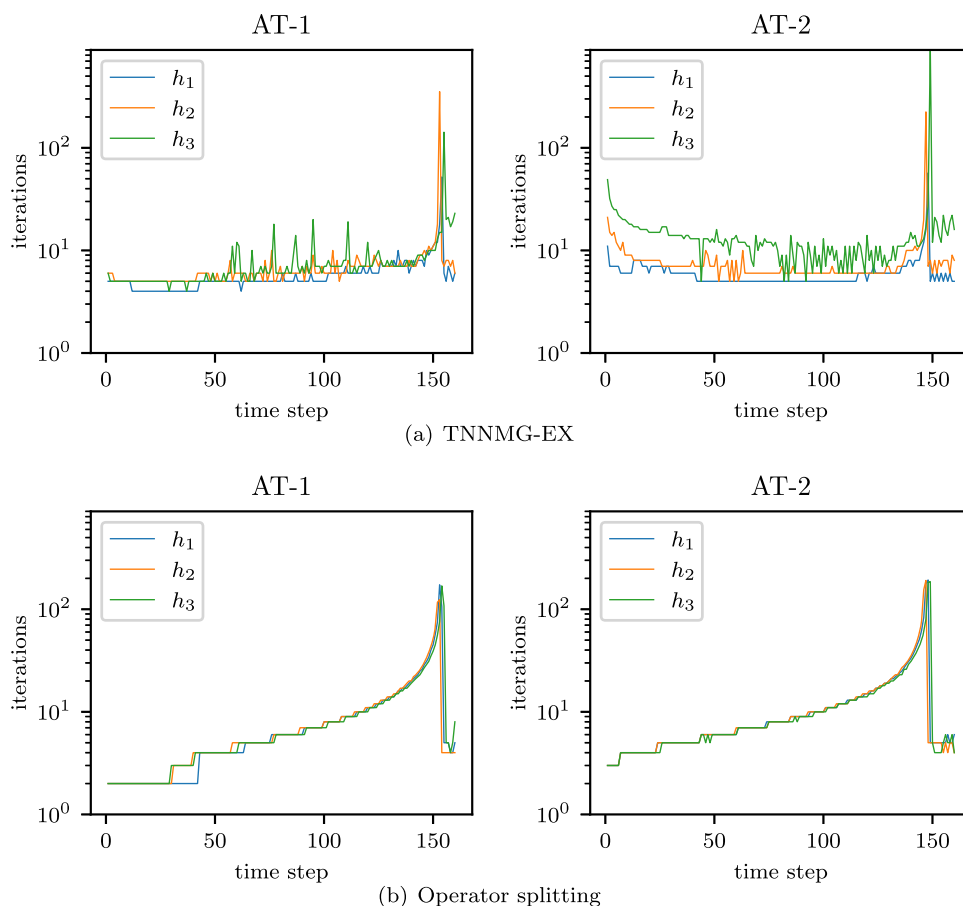


We first compare iteration numbers. At each loading step, the increment problem is solved starting from the solution of the previous time step until the energy norm of the correction normalized by the energy norm of the previous iterate drops below  $10^{-7}$ . The upper row of Fig. 4 shows the number of iterations for the TNNMG solver with the exact smoother (TNNMG-EX), with logarithmically scaled vertical axis.

As can be seen, the iteration numbers remain essentially bounded independently from the grid resolution. The peak shortly before the 150th load step is where the material ruptures. In this situation, the system becomes highly unstable, which results in a higher number of iterations.

After the rupture, the iteration numbers do depend on the grid resolution. This is typical for a multigrid solver—

**Fig. 4** 2d example, isotropic energy split: iterations per time step, for grid sizes  $h_1, h_2, h_3$



presumably it is caused by the fact that, given the particular boundary conditions, the completely ruptured specimen is essentially an ill-posed problem.

For comparison, the lower row of Fig. 4 shows the iteration numbers of the operator splitting method. One can see that for the first two thirds of the loading history, this method needs less iterations than the TNNMG algorithm, and that iteration numbers are independent from the grid resolution. Recall, however, that TNNMG iterations are much cheaper than operator-splitting iterations, because they are essentially a low fixed number multigrid iterations, whereas each operator-splitting iteration involves solving two global linear systems. In contrast to the TNNMG method, the number of iterations increases with increasing load, and shortly before the peak iteration numbers are higher.

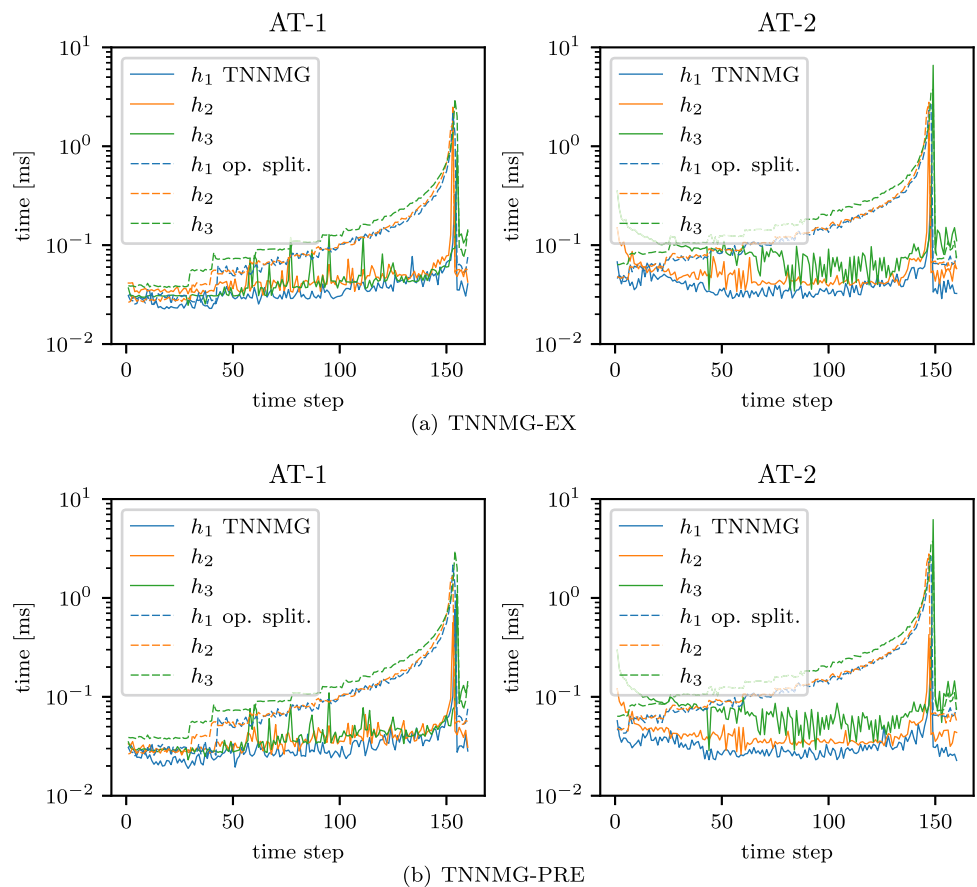
We do not show the iteration numbers of TNNMG with the inexact smoother (TNNMG-PRE), because they coincide with the results for TNNMG-EX. This is not surprising: As the model uses the isotropic splitting of the elastic energy, the local displacement problems are quadratic, and a single Newton step solves them exactly. TNNMG-PRE uses a preconditioner for those quadratic problems, which in this particular situation is very similar to the actual problem. Therefore, preconditioning here has only a limited impact on

the smoothing and thus on the speed of convergence. However, since the preconditioner is independent of the current iterate, the corresponding local matrices can be precomputed.

Wall-time behavior is discussed next. We plot wall-time per time step for the two multigrid variants TNNMG-EX and TNNMG-PRE, and for the operator-splitting method (Fig. 5, again with logarithmic vertical axes). The plots show the time normalized by the number of degrees of freedom.

We see that TNNMG is about 2 to 3 times faster than operator-splitting. The time per degree of freedom stays roughly constant for both methods, independent of the grid resolution. Presumably, the superlinear complexity of the direct solver used in the operator-splitting method only shows for larger grids. Table 1 shows the accumulated normalized run-times for the entire load history. It shows that the speed difference for the  $h_3$  grid accumulates to a factor of about 3.5 for the AT-1 model. For the AT-2 model the difference is only a factor of about 2. This is a bit surprising: The factor is larger for the smaller grids, but TNNMG, for the AT-2 model, slows down considerably when going from the  $h_2$  grid to the  $h_3$  grid. This is caused by higher iteration numbers throughout the load history, as can be seen in Fig. 4. The reason for this behavior is unclear.

**Fig. 5** 2d example, isotropic energy split: wall-time per degree of freedom per time step, for grid sizes  $h_1, h_2, h_3$



**Table 1** 2d example, isotropic energy split: total wall time per degree of freedom (in milliseconds)

	AT-1			AT-2		
	TNNMG-EX	TNNMG-PRE	OS	TNNMG-EX	TNNMG-PRE	OS
$h_1$	5.89	5.67	20.47	6.74	5.39	29.19
$h_2$	9.56	6.58	20.51	9.94	7.18	29.43
$h_3$	8.13	7.81	28.84	20.78	18.29	40.92

Figure 5 and Table 1 also show the wall-times for TNNMG-PRE, the TNNMG variant smoothing with an inexact local solver. It can be observed that using that smoother decreases the computation time by roughly further 5% to 30%. This is the effect of precomputing the local preconditioned Hessians in TNNMG-PRE.

Finally, we point out that the AT-1 model is solved more quickly than the AT-2 one, even though the threshold for damage formation makes it more challenging.

### 5.2.2 Spectral splitting of the elastic energy density

For the next experiment we exchange the isotropic energy splitting (6) by the spectral one (9). As the example specimen is loaded in tension we expect few differences to the previous experiments, and indeed, the displacement–load curves (in

Fig. 6) show only minor differences compared to the ones of the isotropic splitting (Fig. 3).

The purpose of this test is primarily to assess the cost of the two-dimensional eigenvalue decomposition and its derivatives required for the spectral splitting. Figure 7 shows the iteration numbers per time step for the three methods. As the model is not quadratic in the displacement anymore, we now distinguish between the TNNMG-EX and the TNNMG-PRE smoothers. Not surprisingly though, the iteration numbers are virtually identical. The iterations needed by the operator splitting method, shown in the lowest row of Fig. 7, have not changed appreciably either. As an exception to this, the TNNMG-PRE method on the  $h_3$  grid needs a much larger number of iterations at the actual rupture, where the problem is very ill-conditioned. While this is only a single load step, it markedly influences the accumulated wall-times.

**Fig. 6** 2d example, spectral energy split: evolution at time steps 145 and 160, and displacement–force curves for the AT-1 (top) and AT-2 (bottom) models

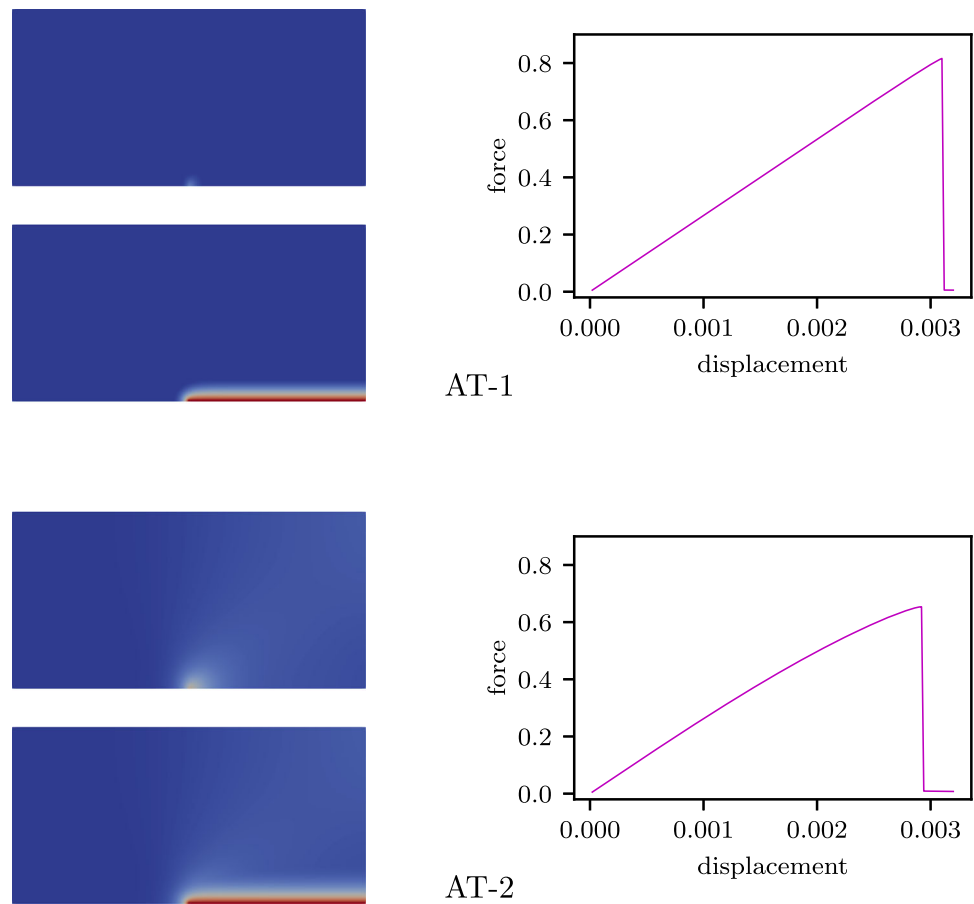


Figure 8 shows the time needed to solve the increment problems, again normalized by the number of degrees of freedom. One can see that the overall behavior remains unchanged, but that the time needed by TNNMG goes up a bit, in particular for the AT-2 model. As the number of iterations per load step remains largely unchanged, the observed cost increase is caused by the spectral decompositions performed by the smoother. Interestingly, the operator-splitting solver for the AT-2 problem is a bit faster during the rupturing of the specimen than it was for the isotropically split energy.

When looking at the accumulated run-times shown in Table 2 (still normalized by the number of degrees of freedom), we see that the time needed by the operator-splitting method remains virtually unchanged. The TNNMG methods have gotten slower, though, and have only a small wall-time lead over operator-splitting. As in the case of the isotropic splitting, moving from the  $h_2$  grid to the  $h_3$  grid increases the time per degree of freedom considerably for the AT-2 model. The reason is unclear.

Using the preconditioned smoother instead of the exact one now leads to a significant decrease in the accumulated wall-time, with time reductions between 10% and 40%. This is caused by the fact that the preconditioned smoother computes much fewer eigenvector decompositions. No such

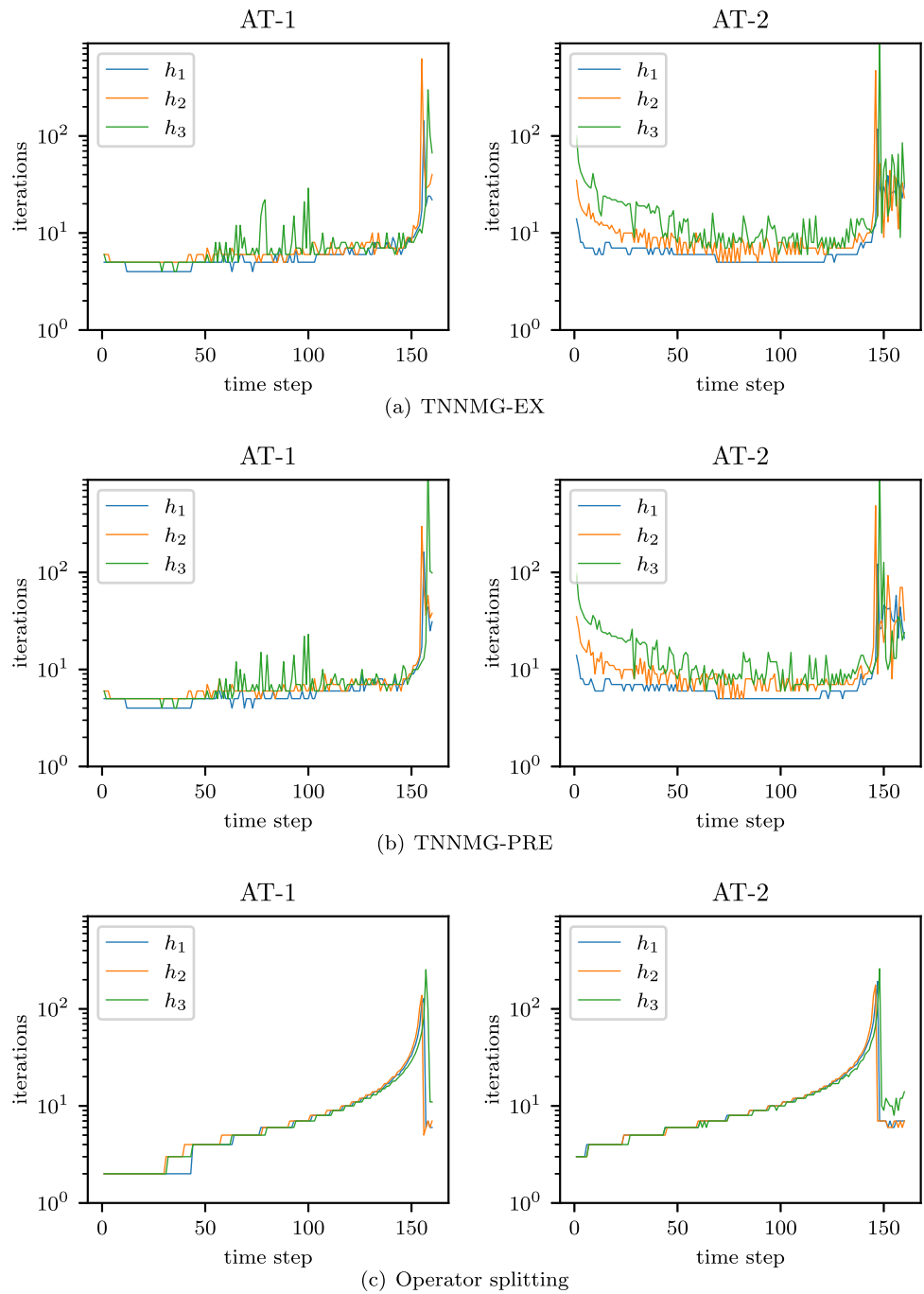
speedup can be seen for the AT-1 model on the  $h_3$  grid, where the preconditioned smoother is actually slower than the exact one. As mentioned, this is because the algorithm needs many more iterations at the load step with the rupturing in this case. The reason is unknown.

### 5.2.3 Comparison to an interior-point solver from the literature

To allow further assessment of the solver wall-times, we compare them with results published by [52, Chapter 4.1], that use an interior-point solver for a very similar example. There, the authors consider the same problem geometry and loading (quantitatively), and similar material parameters. They model the material with an AT-2 functional and an elastic energy using Amor’s volumetric–deviatoric splitting [4], see Sect. 2.1.2. Note that this splitting is cheaper to evaluate than the eigenvalue-based one whose times are given in Table 2.

As we do, the authors simulate a linearly increasing displacement load until a bit after complete rupturing, but they use only 60 load steps compared to our 160 ones. Apparently they simulate the entire square, but with 17421 vertices, their grid is a bit coarser than our  $h_1$  grid (which has 33153 vertices). They give cumulative simulation times for four types

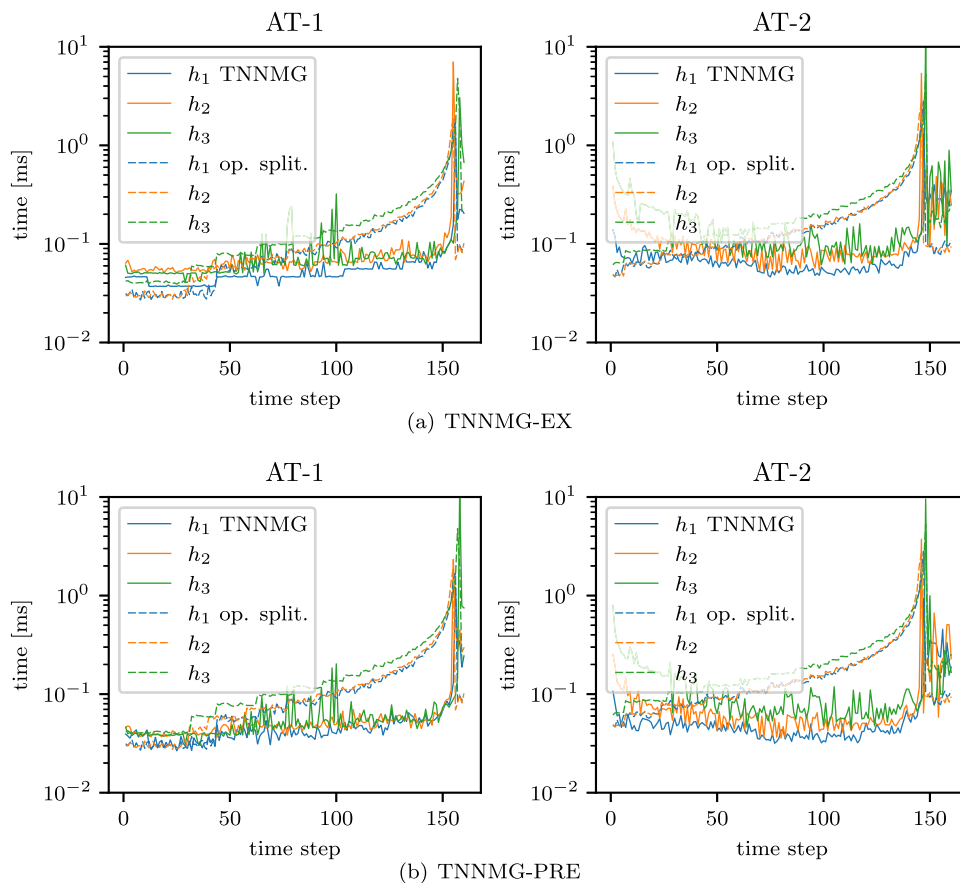
**Fig. 7** 2d example, spectral energy split: iterations per time step, for grid sizes  $h_1, h_2, h_3$



**Table 2** 2d example, spectral energy split: total wall time per degree of freedom (in milliseconds)

	AT-1			AT-2		
	TNNMG-EX	TNNMG-PRE	OS	TNNMG-EX	TNNMG-PRE	OS
$h_1$	10.20	9.09	20.99	14.78	11.17	29.59
$h_2$	19.58	11.70	21.76	23.55	16.78	29.68
$h_3$	16.26	21.79	32.84	39.64	28.65	40.96

**Fig. 8** 2d example, spectral energy split: wall-time per degree of freedom per time step, for grid sizes  $h_1, h_2, h_3$



of solvers, of which three are of operator-splitting type (with different local solvers), and one is a monolithic interior-point solver. For the operator-splitting solvers they report wall-times per degree of freedom between 6.9ms and 27.6ms. For a better comparison with our simulation with 160 load steps we multiply these times with  $\frac{160}{60}$  and obtain values between 18.4ms and 73.5ms. Likewise, for the monolithic solver they report a cumulative time per degree of freedom of 82.7ms (for 60 load steps). Scaled to 160 load steps this gives a value of about 220ms. These times should be compared to the ones of the AT-2 column of Tables 1 and 2. One can see that our implementations are quite a bit faster.

Note, however, that the two example problems are not exactly the same, that termination criteria differ,<sup>8</sup> and that both hardware and implementation differ as well. Also, the grid used by [52] is coarser than ours. Therefore, the comparison should not be over-interpreted. On the other hand, the interior-point implementation involves sparse matrix factorizations, and it is therefore to be expected that the run-times deteriorate rapidly for increasing grid resolution, in particular for three-dimensional problems.

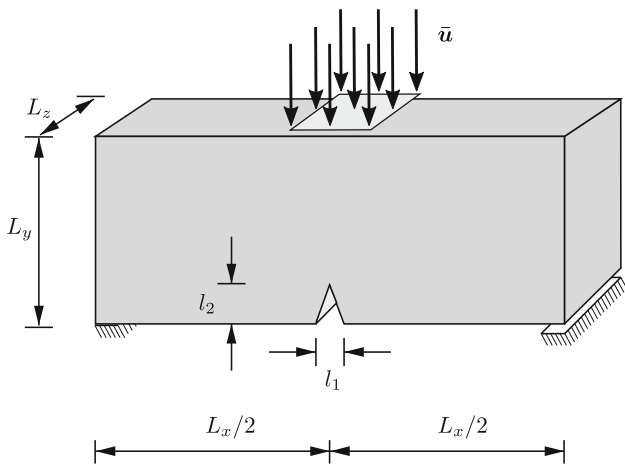
<sup>8</sup> The termination criterion in [52] involves Lagrange multipliers for a KKT system that are not present in the problem formulation considered here.

### 5.3 Bending test of a notched bar in three dimensions

The second example uses a three-dimensional object. In three dimensions, stiffness matrices get denser, and hence direct solvers for global linear systems get more expensive and need considerably more memory. In contrast, the TNNMG memory consumption remains linear in the number of unknowns. Also, eigenvalue decompositions are more expensive for  $3 \times 3$  matrices, and we therefore expect larger run-time differences between the isotropic and the spectrally split model, and between the two TNNMG smoothers.

We consider a bending test for a rectangular bar with a triangular notch. In this setting, the decomposition of the elastic energy density plays a crucial role. Under the given loading, parts of the specimen undergo severe compression, and material models that degrade under such compression will therefore show unphysical results. We test the solvers with the isotropic splitting (6) nevertheless, to obtain an idea of the cost of the spectral splitting.

The example setting is again taken from [38]. The geometry and the boundary conditions are visualized in Fig. 9. The dimensions of the specimen are  $L_x = 8\text{ mm}$ ,  $L_y = 2\text{ mm}$  and  $L_z = 1\text{ mm}$ . Width and height of the triangular notch are  $l_1 = 0.4\text{ mm}$  and  $l_2 = 0.2\text{ mm}$ , respectively. We use three



**Fig. 9** Boundary value problem of bending test in three dimensions

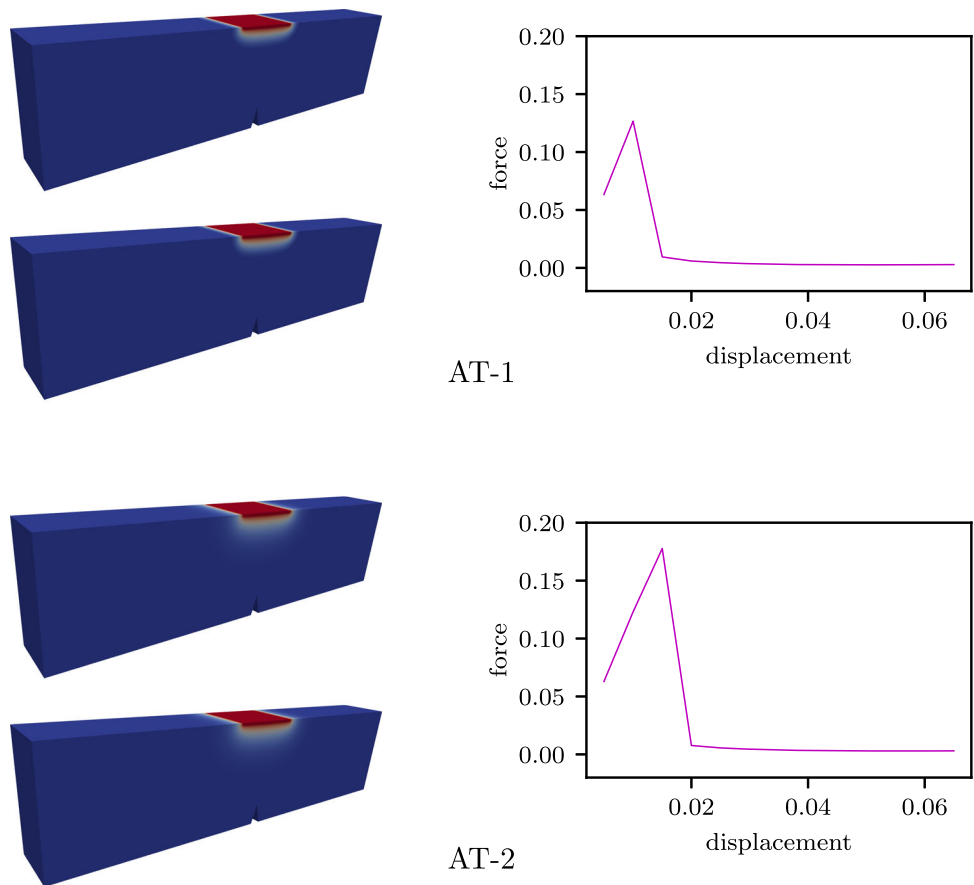
unstructured hexahedral grids to discretize the domain, with 1920, 15 360, and 122 880 elements, respectively. In the figures these are denoted by  $h_1$ ,  $h_2$ , and  $h_3$ , respectively. The grids were constructed by 2, 3, and 4 steps of uniform refinement of a coarse grid with 30 elements, respectively, and this refinement history is used by the linear geometric multigrid step of the TNNMG algorithm. The grids are graded a bit towards the expected fracture, and have an average edge

length of about  $l/2.3$  (for  $h_1$ ),  $l/4.6$  (for  $h_2$ ) and  $l/9.2$  (for  $h_3$ ) there.

As in [38], the Lamé parameters are set to  $\lambda = 121 \text{ kN/mm}^2$  and  $\mu = 80 \text{ kN/mm}^2$ . For the further parameters we set  $g_c = 2.7 \text{ N/mm}$ ,  $l = 0.2 \text{ mm}$ , and  $k = 10^{-5}$ . The object is loaded with a time-dependent displacement load in downward direction in a strip of width 1.2 mm in the center of the top surface. In that strip, the surface is fixed in  $z$ -direction, but free to move in  $x$ -direction. The displacement load is set to  $\bar{u}_i = -i \cdot 5 \cdot 10^{-3} \text{ mm} \cdot \mathbf{e}_3$  for the load steps  $i = 1, \dots, 13$ . No adaptive load-stepping as in [38] is necessary, because time discretization and solvers are stable enough for this range of load step sizes. The object is fully clamped at a strip of width 0.4 mm at the left end of the lower surface. At the right end of the same surface 0.4 mm is fixed in the  $y$  and  $z$  directions.

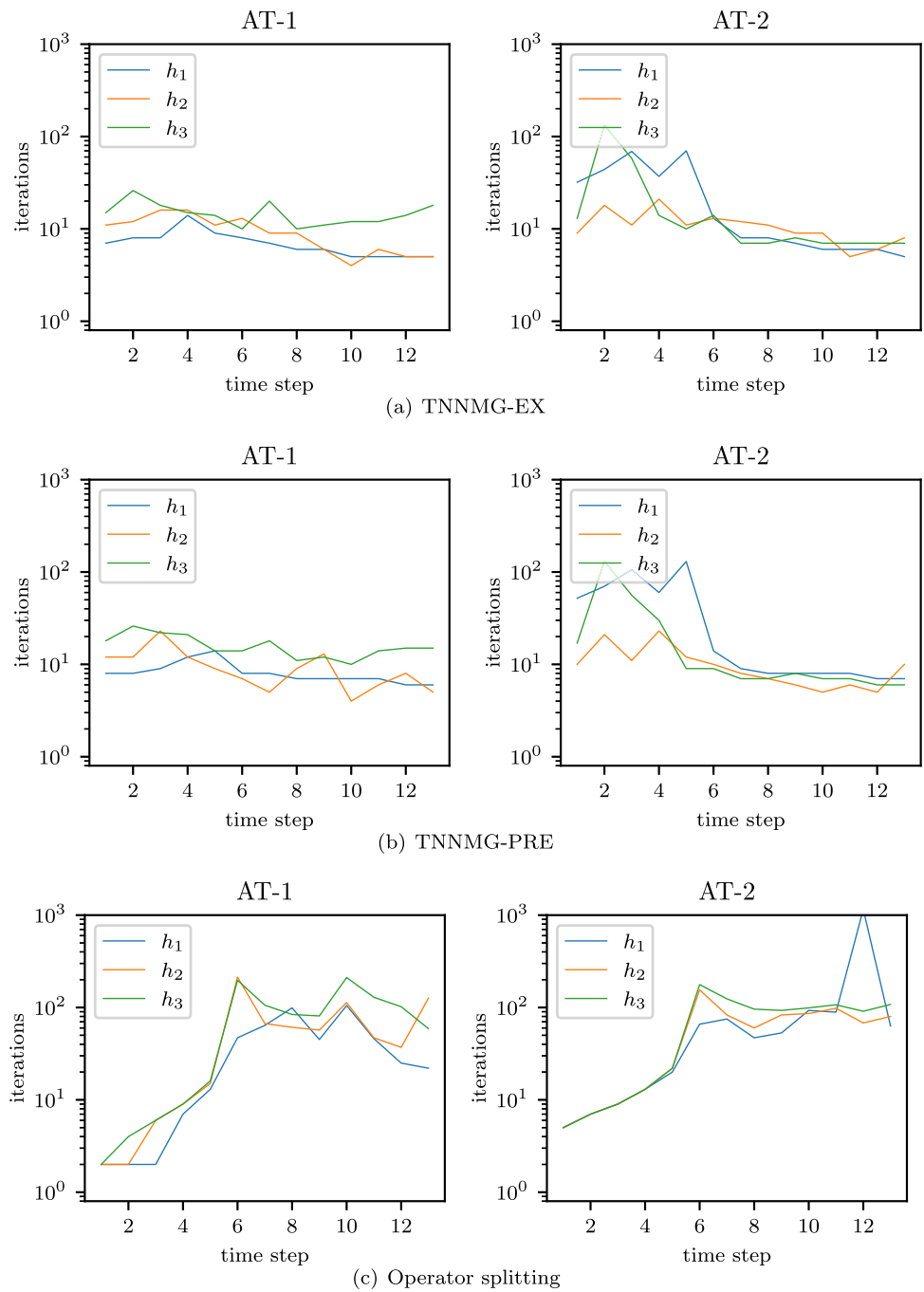
For each time step, we solve the increment problem with the same solver settings as for the two-dimensional example: Each iteration starts at the solution of the previous load step, and we terminate when the degraded energy norm of the correction, scaled by the degraded energy norm of the previous iterate, drops below  $10^{-7}$ . The linear correction step (21) consists of three geometric multigrid  $V(3, 3)$ -cycle steps with a block-Gauß–Seidel smoother, and damage degrees of free-

**Fig. 10** 3d example, isotropic energy split: configurations at load steps 7 and 13, and displacement–force curves





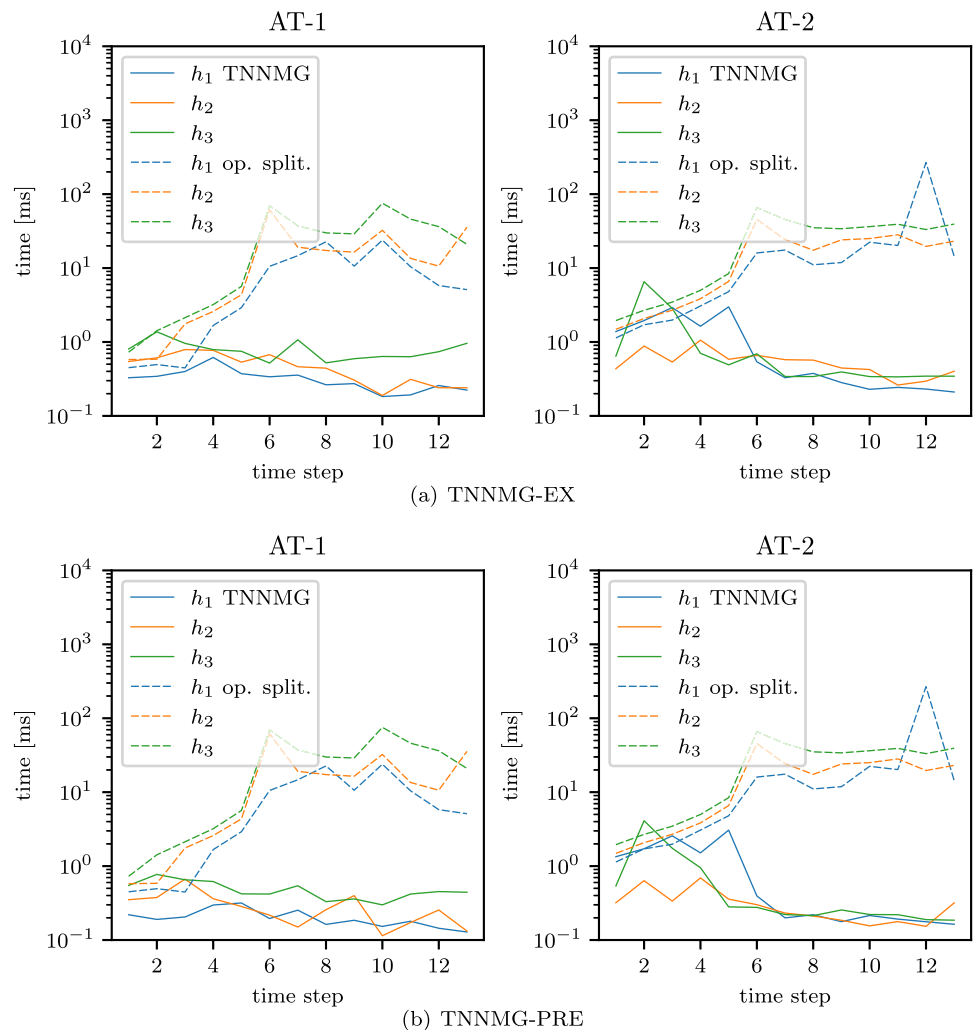
**Fig. 11** 3d example, isotropic energy split: iterations per time step, for grid sizes  $h_1, h_2, h_3$



**Table 3** 3d example, isotropic energy split: total wall time per degree of freedom (in milliseconds), for grid sizes  $h_1, h_2, h_3$

	AT-1			AT-2		
	TNNMG-EX	TNNMG-PRE	OS	TNNMG-EX	TNNMG-PRE	OS
$h_1$	4.15	2.63	109.77	13.31	11.94	393.92
$h_2$	6.11	3.74	216.42	7.12	4.07	223.96
$h_3$	10.34	6.28	357.35	14.36	9.40	349.97

**Fig. 12** 3d example, isotropic energy split: wall-time per degree of freedom over time step, for grid sizes  $h_1, h_2, h_3$



dom are truncated when their current value is less than  $10^{-10}$  away from the lower obstacle.

### 5.3.1 Isotropic splitting

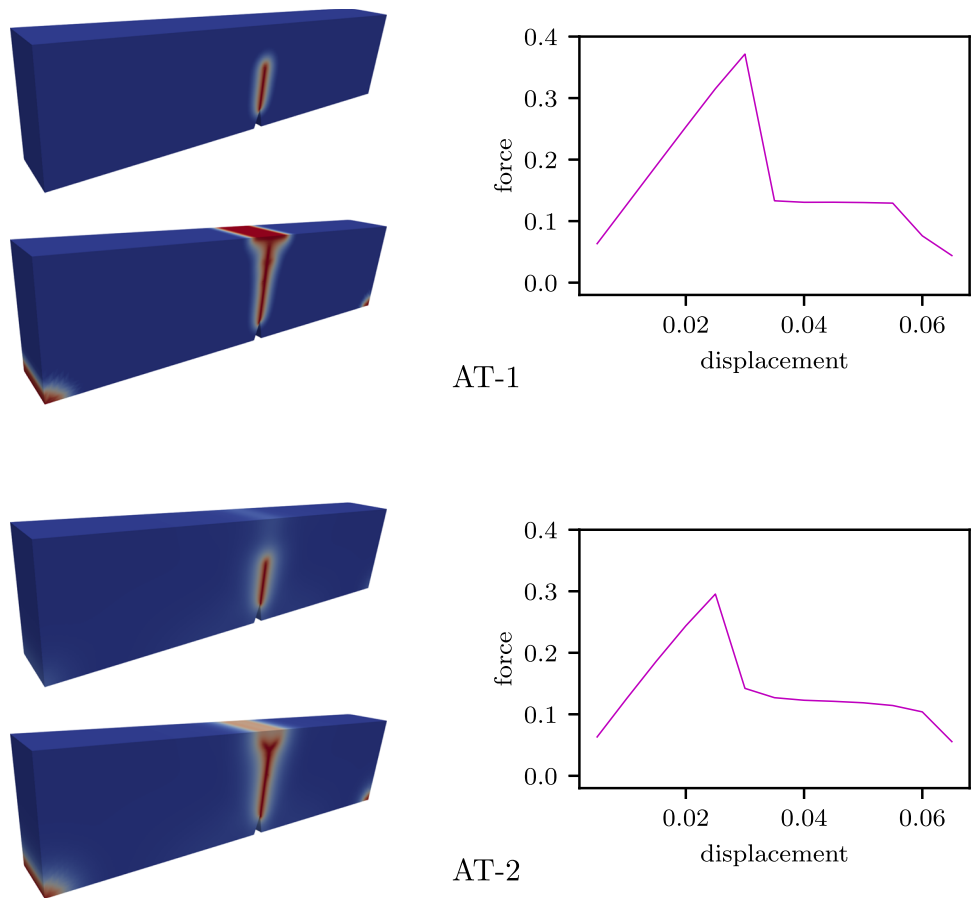
We first consider the model with the isotropic splitting (6) of the elastic energy density, i.e., the model where all elastic strains contribute to the degradation of the material. For this particular benchmark we expect unphysical results: Virtually all damage will happen in the vicinity of the load, whereas the region around the notch will remain intact. Indeed, the simulation results in Fig. 10 show that this is exactly what is happening. We are nevertheless interested in the solver behavior for this model, primarily to assess the cost of the spectral splitting in the next section. No costly eigenvalue decompositions are necessary for the isotropic splitting considered here, and the local displacement problems (24) are quadratic. Consequently, the two smoother variants TNNMG-EX and TNNMG-PRE solve almost the same problems, and we expect them to behave more or less

the same, too, with a small run-time advantage for TNNMG-PRE.

To assess solver performance, we again first compare iteration numbers. Figure 11 shows the iteration numbers per time step needed by the two TNNMG variants, and by the operator-splitting iteration. Note again that the vertical axis is scaled logarithmically. In contrast to the two-dimensional experiment we did notice different iteration numbers for the exact and preconditioned smoothers, and we therefore show both plots.

We see that TNNMG needs about 10 iterations for each load step for the AT-1 model. For the AT-2 model, for which the damage variable changes throughout the domain immediately, the solver also needs about 10 iterations on the  $h_2$  grid, but almost 10 times as many for the other two grids for the first 5 load steps. Operator-splitting, on the other hand, behaves roughly the same for both models. Unlike TNNMG, it needs less iterations initially, but many more later on. More specifically, the method needs around 10 iterations or even less for the first 5 load steps, but after that it consistently

**Fig. 13** 3d example, spectral energy split: evolution at time steps 7 and 13, and displacement–force curves



needs about 100 iterations per load step for all grids, with one outlier even going up to 1000 iterations. For an attempt of an explanation, recall that in this example the specimen never breaks into two parts (Fig. 10), and the problem therefore remains much better conditioned than the previous ones. In this situation, the monolithic TNNMG method seems to have a clear advantage.

The difference in iteration numbers together with the lower cost of TNNMG iterations compared to operator-splitting iterations leads to a tremendous speed difference. As shown in Fig. 12, for all load steps beyond the first few, the TNNMG method needs about two orders of magnitude less wall-time than the operator-splitting method. This difference can also be seen in Table 3, which shows the accumulated wall-times for the entire load history, normalized by the number of degrees of freedom. We see that TNNMG-EX is about 25 to 35 times faster than operator-splitting for the AT-1 model, and 25 to 30 times faster for the AT-2 model. Using the preconditioned smoother makes the wall-time decrease further. In this three-dimensional situation, not having to recompute the matrix block diagonal entries does lead to large savings. The speed advantage of TNNMG-PRE over operator-splitting rises to about 42–58 for the AT-1 model, and to 33–57 for the AT-2 one. In situations such as hydraulic

fracturing where an isotropic model is justified, the advantage of TNNMG is therefore considerable.

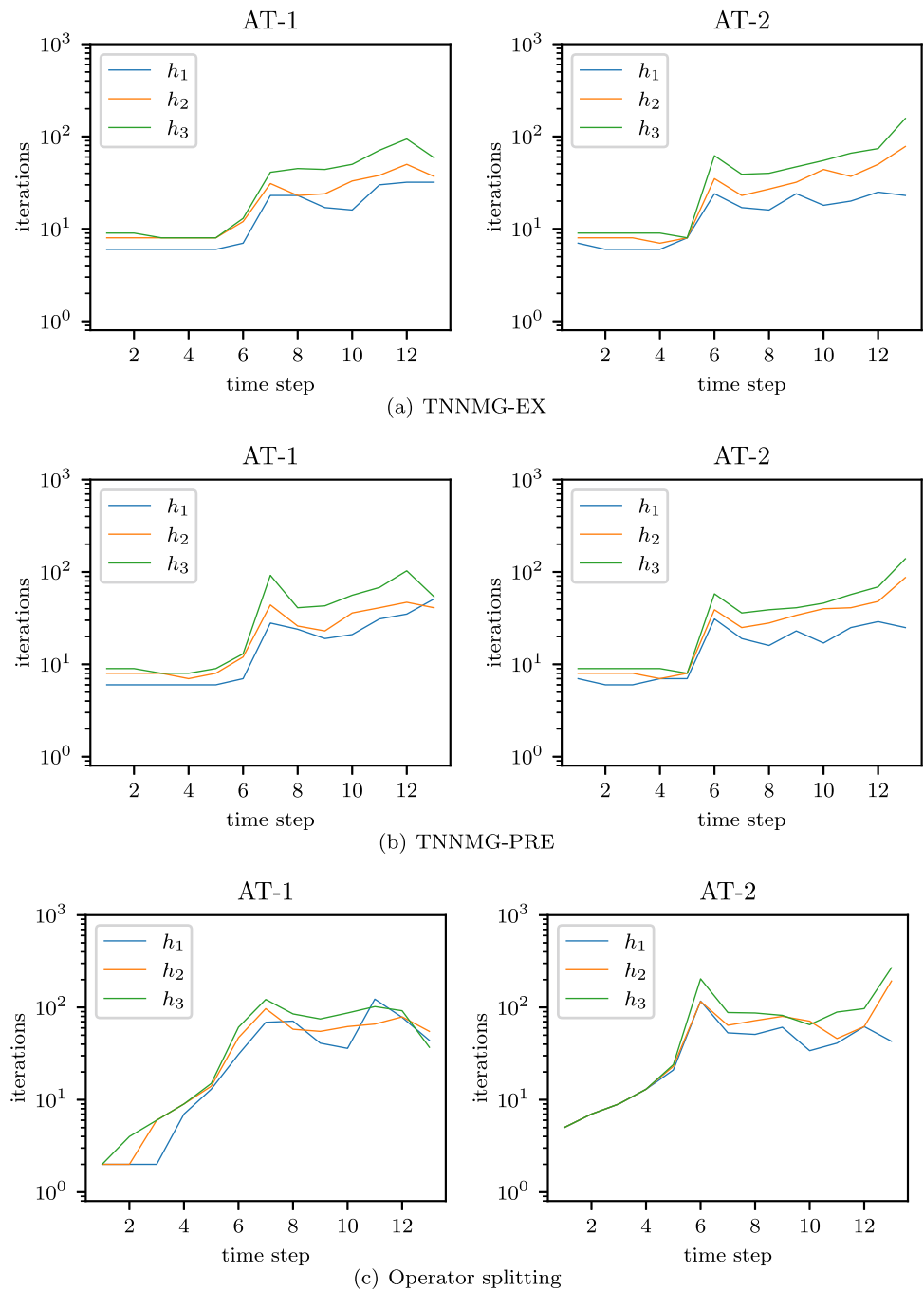
### 5.3.2 Spectral splitting of the elastic energy density

In the second set of experiments we replace the unsplit degraded energy density (6) by the one with the spectral splitting according to (9). Figure 13 shows two snapshots from the problem evolution, and the reaction force as a function of the applied displacement. One can clearly see the differences to the simulation with the unsplit energy. As one would expect, the damage now happens primarily near the notch, and the specimen does now break into two parts.

Figure 14 shows the iteration numbers for TNNMG with both smoothers and the operator-splitting method again. Iteration numbers for the multigrid algorithms and both AT models are roughly the same. In particular, TNNMG-PRE, the multigrid method with the inexact smoother avoiding the costly  $3 \times 3$  eigenvalue decomposition, does not need more iterations than TNNMG-EX. The operator-splitting method needs slightly less iterations than TNNMG for the first few load steps, and a few more after that.

When looking at wall-times, the TNNMG algorithm is consistently faster than the operator-splitting method. Fig-

**Fig. 14** 3d example, spectral energy split: iterations per time step, for grid sizes  $h_1, h_2, h_3$

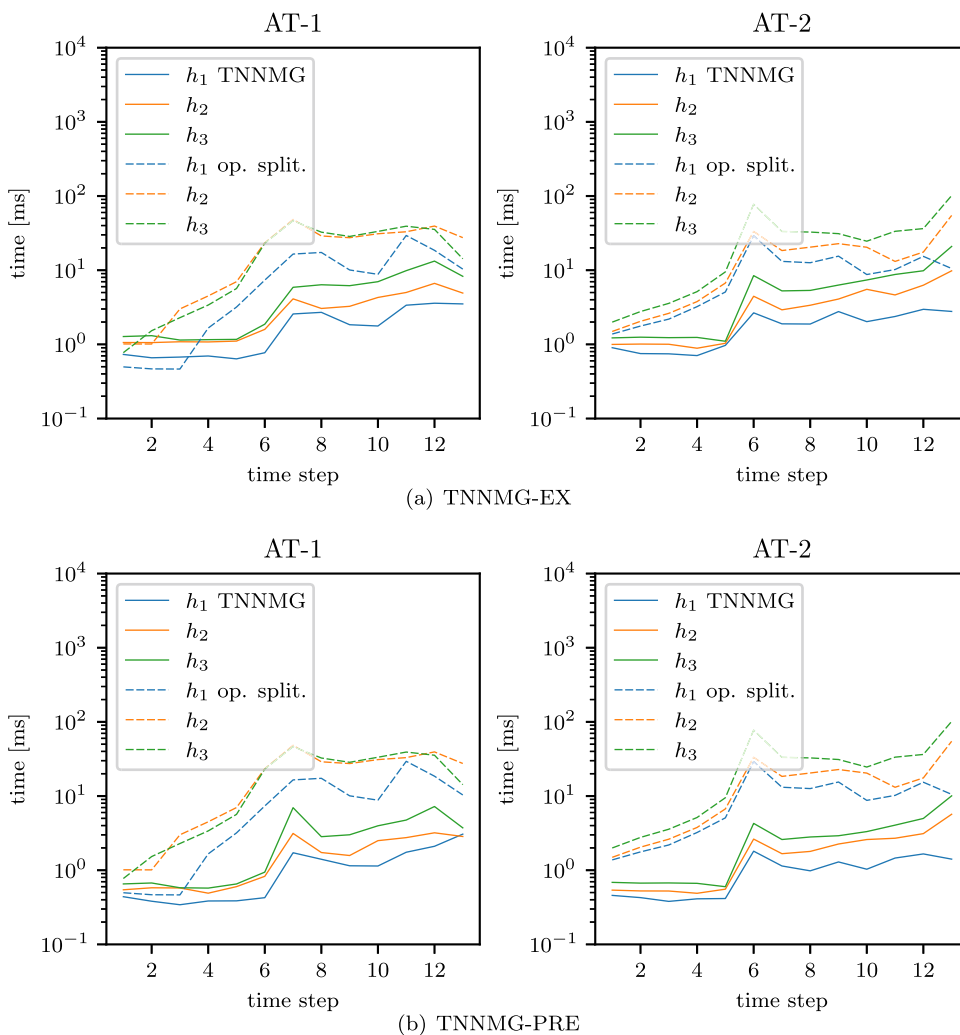


ure 15 shows the normalized times in the same way as in the previous sections. The TNNMG-EX method is roughly five times faster in each load step. This shows the effect of the cheaper iteration times: In three spaces dimensions, tangent matrices are denser than in two dimensions, and direct solving of linear systems with such matrices gets more expensive. Consequently, operator-splitting iterations get relatively more expensive than multigrid ones. This is also reflected in Table 4, which shows the accumulated run-times for the entire load history per degree of freedom. Here, TNNMG-

EX is about 4 to 7 times faster than operator-splitting for the AT-1 model, and roughly 5 times faster for the AT-2 model.

The speed difference gets even larger when using the preconditioned smoother. Computing eigenvector decompositions (and the derivative transformation formulas for spectral functions; Sect. 4.3) is much more expensive for  $3 \times 3$  matrices than for  $2 \times 2$  ones. Therefore, using the preconditioned smoother, which avoids many of these computations, saves a lot of time. As can be seen from Fig. 15

**Fig. 15** 3d example, spectral energy split: wall-time per degree of freedom over time step, for grid sizes  $h_1, h_2, h_3$



**Table 4** 3d example, spectral energy split: total wall time per degree of freedom (in milliseconds), for grid sizes  $h_1, h_2, h_3$

	AT-1			AT-2		
	TNNMG-EX	TNNMG-PRE	OS	TNNMG-EX	TNNMG-PRE	OS
$h_1$	23.52	14.69	124.63	23.42	12.87	128.93
$h_2$	38.26	21.36	275.54	46.02	25.07	217.47
$h_3$	64.76	36.55	266.59	78.29	38.31	393.45

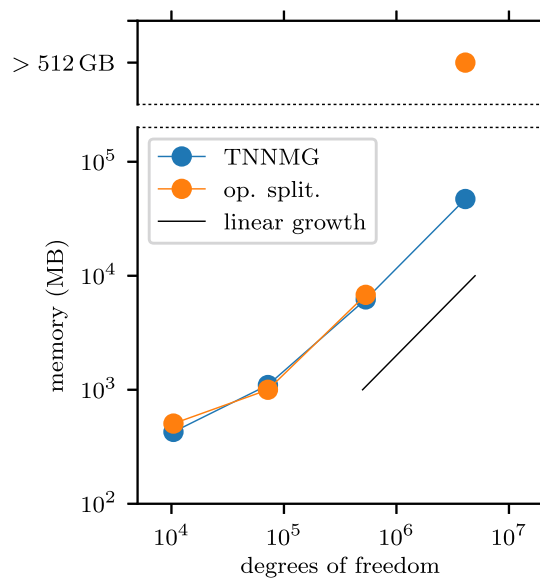
and Table 4, TNNMG-PRE is about 7 to 12 times faster for the AT-1 model, and 8 to 10 times faster for the AT-2 model.

Finally, we investigate the memory consumption of the two solver implementations. Figure 16 shows the maximum amount of memory used by the two algorithms for the problem of this section, as a function of the grid size. Memory consumption was measured using the `valgrind-massif` tool<sup>9</sup> with the `--pages-as-heap` option. One can see that TNNMG clearly requires a linear amount of memory. On the other hand, the proportionality factor is rather large, because

TNNMG needs the full Newton tangent matrix, restriction operators, and approximate tangent matrices on coarser grid levels. Also, in order to obtain first- and second-order derivatives efficiently during the local solves, our implementation precomputes and stores the values and derivatives of the shape functions at all quadrature points (what Miehe et al. call the *global interpolation matrix* [37, Chap. 4.2]).

In contrast, operator splitting does not use coarser grid levels and only assembles tangent matrices for the displacement and damage problems separately. This leads to a smaller memory footprint for small grids. Once the grid gets finer,

<sup>9</sup> <https://valgrind.org/>



**Fig. 16** 3d example: memory consumption for solving one increment problem of the AT-2 functional

though, the superlinear memory complexity of the direct solver begins to show. To highlight this effect we measured the memory consumption for one further step of grid refinement. The resulting grid has 983 040 elements, and the simulation using TNNMG needs about 47 GB of memory. The corresponding operator-splitting implementation, however, would exhaust even a machine with 512 GB of main memory (Fig. 16).

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

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Alberti G (2000) Variational models for phase transitions, an approach via  $\Gamma$ -convergence. In: Buttazzo G, Marino A, Murthy MKV (eds) *Calculus of variations and partial differential equations: topics on geometrical evolution problems and degree theory*, pp 95–114. Springer, Berlin. [https://doi.org/10.1007/978-3-642-57186-2\\_3](https://doi.org/10.1007/978-3-642-57186-2_3)
- Ambati M, Gerasimov T, Lorenzis LD (2015) A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Comput Mech* 55:383–405. <https://doi.org/10.1007/s00466-014-1109-y>
- Ambrosio L, Tortorelli VM (1990) Approximation of functional depending on jumps by elliptic functional via  $\Gamma$ -convergence. *Commun Pure Appl Math* 43(8):999–1036. <https://doi.org/10.1002/cpa.3160430805>
- Amor H, Marigo J-J, Maurini C (2009) Regularized formulation of the variational brittle fracture with unilateral contact: numerical experiments. *J Mech Phys Solids* 57(8):1209–1229. <https://doi.org/10.1016/j.jmps.2009.04.011>
- Baes M (2007) Convexity and differentiability properties of spectral functions and spectral mappings on Euclidean Jordan algebras. *Linear Algebra Appl* 422(2):664–700. <https://doi.org/10.1016/j.laa.2006.11.025>
- Bastian P, Blatt M, Dedner A, Dreier N-A, Engwer C, Fritze R, Gräser C, Grüninger C, Kempf D, Klöforn R, Ohlberger M, Sander O (2021) The DUNE framework: basic concepts and recent developments. *Comput Math Appl* 81:75–112. <https://doi.org/10.1016/j.camwa.2020.06.007>
- Bertsekas DP (1982) Projected Newton methods for optimization problems with simple constraints. *SIAM J Control Optim* 20(2):221–246. <https://doi.org/10.1137/0320018>
- Bourdin B (2007) Numerical implementation of the variational formulation for quasi-static brittle fracture. *Interfaces Free Bound* 9(3):411–430
- Bourdin B, Francfort G, Marigo JJ (2000) Numerical experiments in revisited brittle fracture. *J Mech Phys Solids* 48:797–826
- Brun MK, Wick T, Berre I, Nordbotten JM, Radu FA (2020) An iterative staggered scheme for phase field brittle fracture propagation with stabilizing parameters. *Comput Methods Appl Mech Eng*. 361:112752. <https://doi.org/10.1016/j.cma.2019.112752>
- Burke S, Ortner C, Süli E (2010) An adaptive finite element approximation of a variational model of brittle fracture. *SIAM J Numer Anal* 48(3):980–1012
- Burke S, Ortner C, Süli E (2013) An adaptive finite element approximation of a generalized Ambrosio–Tortorelli functional. *Math Models Methods Appl Sci* 23(9):1663–1697
- Chen Y, Davis TA, Hager WW, Rajamanickam S (2008) Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Trans Math Softw* 35(3):22:1–22:14. <https://doi.org/10.1145/1391989.1391995>
- Dacorogna B (1989) *Direct methods in the calculus of variations*, 1st edn. Springer, Berlin
- Davis TA (2004) Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans Math Softw* 30(2):196–199. <https://doi.org/10.1145/992200.992206>
- Farrell P, Maurini C (2017) Linear and nonlinear solvers for variational phase-field models of brittle fracture. *Int J Numer Methods Eng* 109(5):648–667
- Gerasimov T, De Lorenzis L (2016) A line search assisted monolithic approach for phase-field computing of brittle fracture. *Comput Methods Appl Mech Eng* 312:276–303
- Gerasimov T, De Lorenzis L (2019) On penalization in variational phase-field models of brittle fracture. *Comput Methods Appl Mech Eng* 354:990–1026. <https://doi.org/10.1016/j.cma.2019.05.038>
- Glowinski R (1984) *Numerical methods for nonlinear variational problems*, 3rd edn. Springer series in computational physics. Springer, Berlin
- Gräser C, Kornhuber R (2009) Multigrid methods for obstacle problems. *J Comput Math* 27(1):1–44
- Gräser C, Sander O (2014) Truncated nonsmooth Newton multigrid methods for simplex-constrained minimization problems. Preprint 384, IGPM Aachen

22. Gräser C, Sander O (2019) Truncated nonsmooth Newton multigrid methods for block-separable minimization problems. *IMA J Numer Anal* 39:454–481. <https://doi.org/10.1093/imanum/dry073>
23. Gräser C, Sack U, Sander O (2009) Truncated nonsmooth Newton multigrid methods for convex minimization problems. In Bercovier M, Gander M, Kornhuber R, Widlund O (eds) *Domain decomposition methods in science and engineering XVIII*, volume 70 of lecture notes in computational science and engineering. Springer
24. Heister T, Wheeler MF, Wick T (2015) A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach. *Comput Methods Appl Mech Eng* 290:466–495. <https://doi.org/10.1016/j.cma.2015.03.009>
25. Jodlbauer D, Langer U, Wick T (2020) Matrix-free multigrid solvers for phase-field fracture problems. *Comput Methods Appl Mech Eng* 372:113431. <https://doi.org/10.1016/j.cma.2020.113431>
26. Kopaničáková A, Kothari H, Krause R (2023) Nonlinear field-split preconditioners for solving monolithic phase-field models of brittle fracture. *Comput Methods Appl Mech Eng* 403: 115733. <https://doi.org/10.1016/j.cma.2022.115733>
27. Kopaničáková A, Krause R (2020) A recursive multilevel trust region method with application to fully monolithic phase-field models of brittle fracture. *Comput Methods Appl Mech Eng* 360: 112720. <https://doi.org/10.1016/j.cma.2019.112720>
28. Kornhuber R (1997) *Adaptive monotone multigrid methods for nonlinear variational problems*. Vieweg + Teubner Verlag. ISBN 3519027224
29. Kristensen PK, Martínez-Pañeda E (2020) Phase field fracture modelling using quasi-Newton methods and a new adaptive step scheme. *Theoret Appl Fract Mech* 107:102446. <https://doi.org/10.1016/j.tafmec.2019.102446>
30. Kuhn C, Schlüter A, Müller R (2015) On degradation functions in phase field fracture models. *Comput Mater Sci* 108:374–384
31. Lancioni G, Royer-Carfagni G (2009) The variational approach to fracture mechanics. A practical application to the french Panthéon in Paris. *J Elast* 95:1–30. <https://doi.org/10.1007/s10659-009-9189-1>
32. Lewis AS (1996) Convex analysis on the Hermitian matrices. *SIAM J Optim* 6(1):164–177
33. Lewis AS, Sendov HS (2001) Twice differentiable spectral functions. *SIAM J Matrix Anal Appl* 23(2):368–386. <https://doi.org/10.1137/S089547980036838X>
34. Mang K, Wick T, Wollner W (2020) A phase-field model for fractures in nearly incompressible solids. *Comput Mech* 65(1):61–78
35. Marigo J-J, Maurini C, Pham K (2016) An overview of the modelling of fracture by gradient damage models. *Meccanica* 51:3107–3128. <https://doi.org/10.1007/s11012-016-0538-4>
36. May S, Vignollet J, de Borst R (2016) A new arc-length control method based on the rates of the internal and the dissipated energy. *Eng Comput* 33(1):100–115
37. Miehe C, Hofacker M, Welschinger F (2010) A phase field model for rate-independent crack propagation: robust algorithmic implementation based on operator splits. *Comput Methods Appl Mech Eng* 199:2765–2778
38. Miehe C, Welschinger F, Hofacker M (2010) Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. *Int J Numer Meth Eng* 83:1273–1311
39. Mielke A, Roubíček T (2015) *Rate-independent systems*. Springer, Berlin
40. Modica L, Mortola S (1977) Un esempio di  $\Gamma$ -convergenza. *Boll Un Mat Ital B* 14:285–299
41. Modica L, Mortola S (1977b) The  $\Gamma$ -convergence of some functionals. preprint 77-7, Istituto Matematico ‘Leonida Tonelli’, Università di Pisa
42. Neitzel I, Wick T, Wollner W (2017) An optimal control problem governed by a regularized phase-field fracture propagation model. *SIAM J Control Optim* 55(4):2271–2288. <https://doi.org/10.1137/16M1062375>
43. Pham K, Amor H, Marigo J-J, Maurini C (2011) Gradient damage models and their use to approximate brittle fracture. *Int J Damage Mech* 20(4):618–652. <https://doi.org/10.1177/1056789510386852>
44. Qi H, Yang X (2003) Semismoothness of spectral functions. *SIAM J Matrix Anal Appl* 25(3):766–783
45. Qi L, Sun J (1993) A nonsmooth version of Newton’s method. *Math Prog* 58:353–367
46. Sander O (2020) *DUNE—The distributed and unified numerics environment*. Springer, Berlin. <https://doi.org/10.1007/978-3-030-59702-3>
47. Sander O, Jaap P (2020) Solving primal plasticity increment problems in the time of a single predictor–corrector iteration. *Comput Mech* 65:663–685. <https://doi.org/10.1007/s00466-019-01788-y>
48. Singh N, Verhoosel C, de Borst R, van Brummelen E (2016) A fracture-controlled path-following technique for phase-field modeling of brittle fracture. *Finite Elem Anal Des* 113:14–29. <https://doi.org/10.1016/j.finel.2015.12.005>
49. Steinke C, Kaliske M (2019) A phase-field crack model based on directional stress decomposition. *Comput Mech*, vol 63: 1019–1046. <https://doi.org/10.1007/s00466-018-1635-0>
50. Thomas M (2010) *Rate-independent damage processes in nonlinearly elastic materials*. PhD thesis, Humboldt-Universität zu Berlin
51. Ulbrich M (2002) *Nonsmooth Newton-like methods for variational inequalities and constrained optimization problems in function spaces*. Technische Universität München, Habilitationsschrift
52. Wambacq J, Ulloa J, Lombaert G, François S (2021) Interior-point methods for the phase-field approach to brittle and ductile fracture. *Comput Methods Appl Mech Eng* 375: 113612. <https://doi.org/10.1016/j.cma.2020.113612>
53. Wheeler M, Wick T, Wollner W (2014) An augmented-Lagrangian method for the phase-field approach for pressurized fractures. *Comput Methods Appl Mech Eng* 271:69–85
54. Wick T (2017) An error-oriented Newton/inexact augmented Lagrangian approach for fully monolithic phase-field fracture propagation. *SIAM J Sci Comput* 39(4):B589–B617
55. Wick T (2017) Modified Newton methods for solving fully monolithic phase-field quasi-static brittle fracture propagation. *Comput Methods Appl Mech Eng* 325:577–611. <https://doi.org/10.1016/j.cma.2017.07.026>
56. Wu J (2018) Numerical implementation of non-standard phase-field damage models. *Comput Methods Appl Mech Eng* 340:767–797
57. Wu J-Y, Huang Y, Nguyen VP (2020) On the BFGS monolithic algorithm for the unified phase field damage theory. *Comput Methods Appl Mech Eng* 360:112704. <https://doi.org/10.1016/j.cma.2019.112704>