Highlights

Open-Source Implementation of Classical and Micromorphic Phase-Field Fracture Models in Code_Aster

Jakob Degn Larsen,Basile Marchand,Michele Ferraiuolo,Giuseppe Abbiati

- Phase-field fracture implemented in Code_Aster using MFront.
- Classical AT1 & AT2 models solved through heat-transfer analogy.
- Micromorphic models solved through a gradient-enhanced internal-variable framework.
- Numerical verification against FEniCSx on SENT and SENS benchmarks.

Open-Source Implementation of Classical and Micromorphic Phase-Field Fracture Models in Code_Aster

Jakob Degn Larsen[a,*], Basile Marchand[b], Michele Ferraiuolo[c] and Giuseppe Abbiati[a]

[a]Department of Civil and Architectural Engineering, Aarhus University, Aarhus, Denmark
[b]Simvia – EDF Group, Palaiseau, France
[c]Innovative Materials and Thermal Structures, Italian Aerospace Research Centre (CIRA), Capua (CE), Italy

ARTICLE INFO

Keywords:
Phase-field fracture
Code_Aster
Open source
Micromorphic
Finite element method
Fracture mechanics

ABSTRACT

Phase-field fracture provides a flexible framework for simulating complex crack processes, but implementations in open-source finite element solvers with structural-analysis capabilities remain limited. This technical note presents an implementation of phase-field fracture models in Code_Aster, with the aim of documenting how established formulations can be realised within its existing solver architecture without modifying the source code. Two complementary strategies are developed. Classical AT1 and AT2 models are implemented through a heat-transfer analogy and solved using a staggered coupling scheme with a bound-constrained phase-field solver. A micromorphic formulation is implemented using an existing gradient-enhanced internal-variable framework, enabling phase-field cohesive-zone models through constitutive behaviours written in MFront. Two benchmark problems are used to verify the implementations against FEniCSx reference solutions. Close agreement is obtained in both force-displacement response and crack path, demonstrating that the considered phase-field fracture models can be implemented consistently in Code_Aster. This provides a reproducible basis for phase-field fracture simulations in Code_Aster as an open-source structural-mechanics environment.

1. Introduction

Phase-field fracture (PFF) has become a widely used framework for computational fracture mechanics because it represents cracks by a continuous damage-like field rather than by explicit crack surfaces [1–4]. This regularised description enables the simulation of crack propagation, branching and coalescence within a finite element setting and has led to many successful implementations in research-oriented computational mechanics codes [5–7].

PFF has been implemented in standalone research codes, e.g. [8–10], proprietary finite element frameworks, e.g. [11–14] and general PDE-oriented and multiphysics frameworks, e.g. [15–19]. These implementations have enabled substantial methodological progress, but they do not always combine extensibility, flexibility and the structural-analysis capabilities required in engineering workflows. By contrast, implementation in open-source structural-mechanics-oriented finite element solvers is less common, see e.g. [20–22]. This limits the practical availability of PFF in software environments that provide mature structural-analysis capabilities, element libraries and pre-/post-processing tools.

This technical note describes the implementation of PFF models in Code_Aster [23], an open-source finite element solver developed by Électricité de France (EDF). The objective is not to introduce a new phase-field formulation, but to document how PFF models can be realised within the existing architecture of Code_Aster using its standard user-level interfaces and without modifying its source code. To the authors' knowledge, this is the first implementation of PFF in Code_Aster.

The note is organised as follows. Section 2 summarises the phase-field fracture formulations and Section 3 describes the realisation in Code_Aster. Section 4 presents verification examples and Section 5 concludes with limitations and possible extensions.

*Corresponding author
jdl@cae.au.dk (J.D. Larsen); basile.marchand@simvia.tech (B. Marchand); m.ferraiuolo@cira.it (M. Ferraiuolo); abbiati@cae.au.dk (G. Abbiati)
ORCID(s): 0009-0006-9247-3287 (J.D. Larsen); 0009-0000-3171-6202 (B. Marchand); 0000-0002-3200-321X (M. Ferraiuolo); 0000-0002-5048-8505 (G. Abbiati)

2. Phase-field formulation

This section summarises the two phase-field fracture formulations implemented in Code_Aster. Small-strain kinematics are assumed, with displacement field $\boldsymbol{u}$, strain tensor $\boldsymbol{\varepsilon}$ and scalar phase-field variable $d \in [0,1]$, where $d = 0$ denotes intact material and $d = 1$ a fully fractured material. Crack irreversibility is imposed through the bound constraint $d_n \leq d \leq 1$, where d_n is the phase field from the previous load step.

2.1. Classical phase-field formulation

The classical PFF formulation [7, 24] is written in terms of the primary fields $\boldsymbol{u}$ and d. The weak form equations are

$$\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\,\mathrm{d}V - \int_\Omega \bar{\boldsymbol{b}} \cdot \delta\boldsymbol{u}\,\mathrm{d}V - \int_{\partial\Omega_t} \bar{\boldsymbol{t}} \cdot \delta\boldsymbol{u}\,\mathrm{d}A = 0, \quad (1)$$

$$\int_\Omega \frac{G_c}{c_w \ell}\left[w'(d)\delta d + 2\ell^2 \nabla d \cdot \nabla \delta d\right]\mathrm{d}V - \int_\Omega Y\,\delta d\,\mathrm{d}V \geq 0, \quad (2)$$

Nomenclature

$\bar{\boldsymbol{b}}$	Prescribed body forces	c	Nonlocal parameter
c_p	Specific heat capacity	c_w	Normalisation constant
d	Crack phase field	$\mathbf{d}$	Nodal crack phase-field vector
d_n	Crack phase field at previous load step	E	Young's modulus
f_t	Tensile strength	G_c	Critical energy release rate
$g(d)$	Degradation function	$\mathbf{C}$	Heat-transfer capacity matrix
$\mathbf{K}$	Heat-transfer conductivity matrix	$\mathbf{K}^d$	Phase-field tangent stiffness matrix
ℓ	Phase-field length scale	Q	Heat source term
$\mathbf{q}$	Heat-transfer source vector	r	Penalty parameter
$\mathbf{r}^d$	Phase-field residual vector	$\bar{\boldsymbol{t}}$	Prescribed tractions
$\boldsymbol{u}$	Displacement field	$\bar{\boldsymbol{u}}$	Prescribed displacement field
$w(d)$	Geometric crack function	Y	Crack-driving force
$\bar{Y}$	Effective crack-driving force	α	Micromorphic field
$\boldsymbol{\varepsilon}$	Infinitesimal strain	λ	Lagrange multiplier field
λ_t	Thermal conductivity	ν	Poisson's ratio
ρ	Mass density	$\boldsymbol{\sigma}$	Cauchy stress
$\boldsymbol{\sigma}_D$, $\boldsymbol{\sigma}_R$	Degradable/residual stress	ψ_D, ψ_R	Degradable/residual strain energy density
Ω	Computational domain	$\partial\Omega_t$	Boundary with prescribed tractions
$\delta(\square)$	Admissible variation	$\partial_x(\square)$	Partial derivative with respect to x

for all admissible variations $\delta\boldsymbol{u}$ and δd satisfying the essential boundary conditions and the bound constraints, respectively. The stress tensor and crack-driving force are written as

$$\boldsymbol{\sigma} = g(d)\boldsymbol{\sigma}_D + \boldsymbol{\sigma}_R, \qquad Y = -g'(d)\bar{Y}, \tag{3}$$

where in this work, we use $\boldsymbol{\sigma}_D = \partial_\varepsilon \psi_D$, $\boldsymbol{\sigma}_R = \partial_\varepsilon \psi_R$ and $\bar{Y} = \psi_D$ with ψ_D and ψ_R obtained from an energy split [25]. Specifically, we use none (ND) [2], spectral (SD) [4] and volumetric-deviatoric (VD) [26] energy decompositions. The relevant PFF models, namely, AT1 [27], AT2 [2] and CZM [24], are obtained by varying $g(d)$ and $w(d)$.

2.2. Micromorphic phase-field formulation

The micromorphic PFF formulation [28] introduces a nonlocal micromorphic field α and a Lagrange multiplier field λ, while the phase field d is treated as a local internal variable. The mechanical equilibrium equation in Eq. (1) is retained and coupled to the following micromorphic balance and constraint equations:

$$\int_\Omega \frac{2G_c\ell}{c_w}\nabla\alpha\cdot\nabla\delta\alpha\,\mathrm{d}V + \int_\Omega \left[\lambda\delta\alpha + r(\alpha - d)\delta\alpha\right]\mathrm{d}V = 0, \tag{4}$$

$$\int_\Omega (\alpha - d)\delta\lambda\,\mathrm{d}V = 0. \tag{5}$$

The constitutive relations in Eq. (3) remain unchanged, and the local phase-field variable is obtained by solving

$$\frac{G_c}{c_w\ell}w'(d) - Y - \lambda - r(\alpha - d) = 0, \tag{6}$$

still subject to the bounds $d_n \leq d \leq 1$. This formulation is useful in Code_Aster because the nonlocal part can be represented through the existing gradient-enhanced internal-variable framework [29], while the bound-constrained phase-field update remains local.

3. Numerical implementation

This section describes the two implementation strategies used to solve the phase-field formulations introduced in Section 2. We first describe the implementation of the classical formulation, followed by the implementation of the micromorphic implementation.

3.1. Classical implementation: heat-transfer analogy

The classical PFF formulation is solved using a staggered scheme, in which the mechanical and phase-field subproblems are solved sequentially. The mechanical equilibrium equation is handled by the native nonlinear mechanical solver in Code_Aster, with the constitutive response provided by an MFront [30] behaviour. The phase-field equation is assembled through the transient heat-transfer operator in Code_Aster and solved by coupling the resulting algebraic system to the *reduced-space active-set Newton solver* available in the PETSc SNES library.

The heat-transfer analogy is used only at the algebraic level. The phase-field variable d is represented by the temperature field, and the thermal capacity and conductivity operators are chosen such that the assembled matrices reproduce the reaction and gradient terms of the phase-field equation. In particular, the capacity matrix normally associated with the transient term is not used to represent a physical time derivative; instead, it provides the pseudo-mass matrix multiplying the current phase-field vector $\mathbf{d}$. The source term is chosen to complete the residual of the AT1 or AT2 phase-field equation. In semi-discrete form, the heat-transfer operator provides the global arrays $\mathbf{C}$, $\mathbf{K}$ and $\mathbf{q}$, corresponding to the capacity matrix, conductivity matrix and source vector, respectively. These arrays are used to construct the phase-field residual and tangent stiffness

$$\mathbf{r}^d(\mathbf{d}) = (\mathbf{C} + \mathbf{K})\,\mathbf{d} - \mathbf{q}, \qquad \mathbf{K}^d = \mathbf{C} + \mathbf{K}, \tag{7}$$

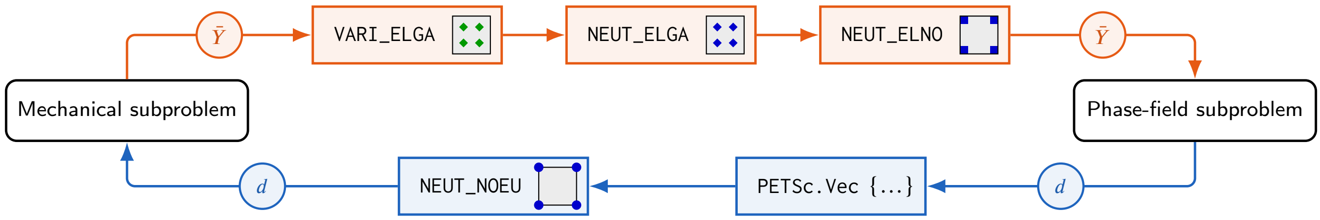


Figure 1: Implemented variable transfer in Code_Aster for the classical formulation. The effective crack driving force $\bar{Y}$ is output from the mechanical subproblem as an internal variable field at the element Gauss points (VARI_ELGA) and converted to a neutral field at the element nodes (NEUT_ELNO) to allow usage in the phase-field subproblem. Similarly, the phase field d is output as a PETSc vector and recast in Code_Aster as a neutral field at the mesh nodes (NEUT_NOEU) for usage in the mechanical subproblem.

Table 1
Coefficient correspondence used to assemble phase-field equations through the transient heat-transfer operator.

Problem	Capacity	Conductivity	Source
Heat-transfer	ρc_p	λ_t	Q
AT1	$2\bar{Y}$	$\frac{3}{4}G_c\ell$	$2\bar{Y} - \frac{3G_c}{8\ell}$
AT2	$\frac{G_c}{\ell} + 2\bar{Y}$	$G_c\ell$	$2\bar{Y}$

where the coefficients defining $\mathbf{C}$, $\mathbf{K}$ and $\mathbf{q}$ depend on the selected phase-field model and on the effective crack-driving force $\bar{Y}$. For the AT1 and AT2 models, the corresponding coefficients are summarised in Table 1. The CZM-type phase-field model is not treated by this route, since its evolution equation does not fit the adopted heat-transfer template.

The field transfer between subproblems follows the procedure shown in Figure 1. This additional step is required because Code_Aster distinguishes fields by both physical nature and discretisation location; consequently, the phase field and effective crack-driving force must be converted into field types that can be used as command variables, i.e. known external fields that enter the material behaviour.

3.2. Micromorphic implementation: GRAD_VARI formulation

The micromorphic formulation is implemented using the native GRAD_VARI model [31]. This model solves the exact weak form equations Eqs. (1), (4) and (5) when setting the nonlocal parameter $c = 2G_c\ell/c_w$. The constitutive relations, i.e. Eqs. (3) and (6), are implemented through MFront where Eq. (6) is solved for the local phase-field variable using the standard Newton-Raphson method and subsequently projected onto the admissible bounds. The lower bound is obtained by declaring the previous damage value as an auxiliary state variable in MFront, allowing it to be passed to Code_Aster and retrieved at the subsequent time step.

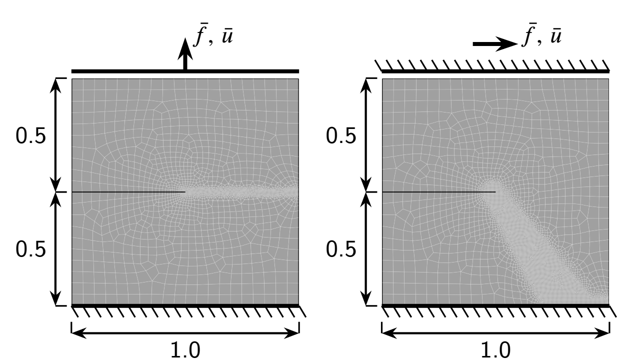
We note that the GRAD_VARI model is restricted to use P2 (quadratic) elements for $\boldsymbol{u}$ and P1 (linear) elements for α and λ with coinciding Gauss points corresponding to the P1 elements. The difference in interpolation order is required to ensure kinematic and numerical consistency since α is driven by the $\boldsymbol{\varepsilon}$, which depends on the spatial derivatives of $\boldsymbol{u}$. Furthermore, we note that the solution scheme is monolithic which is known to have convergence issues in cases of strong crack evolution [32]. To remedy this, we use a path-following method which controls the maximum strain increment [33].

4. Numerical verification

This section proceeds to verify the proposed implementation in Code_Aster. The objective of the verification is not to assess the predictive capability of the PFF models, but to demonstrate that established formulations can be reproduced consistently within Code_Aster. Reference results are obtained from an implementation in FEniCSx [34, 35].

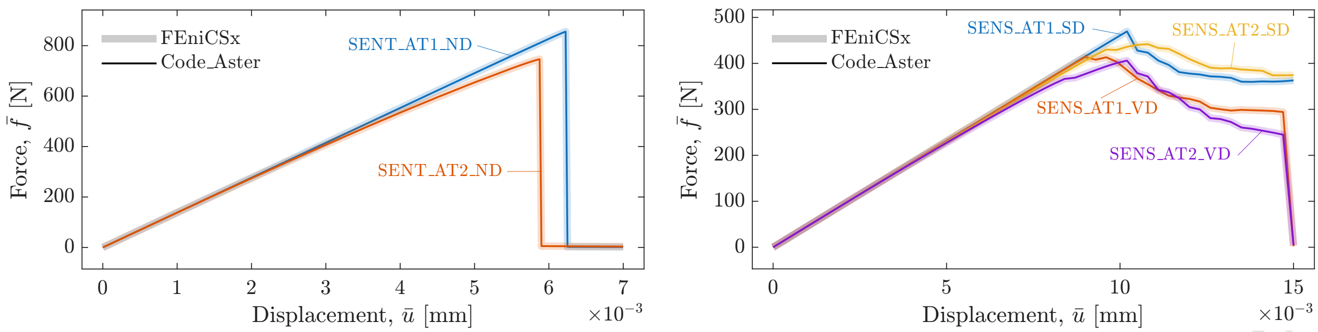
4.1. Benchmark problems

Two common PFF benchmark problems are considered: a single-edge notched specimen subjected to tension (SENT) and a single-edge notched specimen subjected to shear (SENS). The corresponding geometries, meshes and boundary conditions are shown in Figure 2. In both cases, the initial notch is represented geometrically and no phase-field boundary conditions are imposed. All simulations are performed under plane-strain conditions with $E = 210\,\text{GPa}$, $\nu = 0.3$, $G_c = 2.7\,\text{N}\,\text{mm}^{-1}$ and $\ell = 0.01$ mm. For the CZM cases, the tensile strength is set to $f_t = 2000\,\text{MPa}$, and in the micromorphic formulation, the penalty parameter is written as $r = \beta\frac{G_c}{l}$ with $\beta = 100$ in Code_Aster.



(a) SENT (b) SENS

Figure 2: Geometry, mesh and boundary conditions for the benchmark problems. All measurements are in mm.



(a) SENT

(b) SENS

Figure 3: Force-displacement responses for the classical test cases.

4.2. Classical phase-field implementation

For the classical formulation, the FEniCSx implementation reproduces the Code_Aster implementation one-to-one, except for solver-specific convergence criteria and the line-search strategy in the mechanical subproblem. The line search was only used to obtain convergence for cases involving the VD energy split.

Figure 3 compares the force-displacement responses obtained with Code_Aster and FEniCSx for the SENT and SENS benchmarks. The responses are practically indistinguishable for all considered cases, verifying the heat-transfer analogy and staggered coupling strategy. Representative final phase-field distributions are shown in Figure 4. The crack paths and damage fields coincide with the reference solutions; the remaining differences are close to machine precision for the SENT cases and remain below 10^{-5} for the SENS cases.

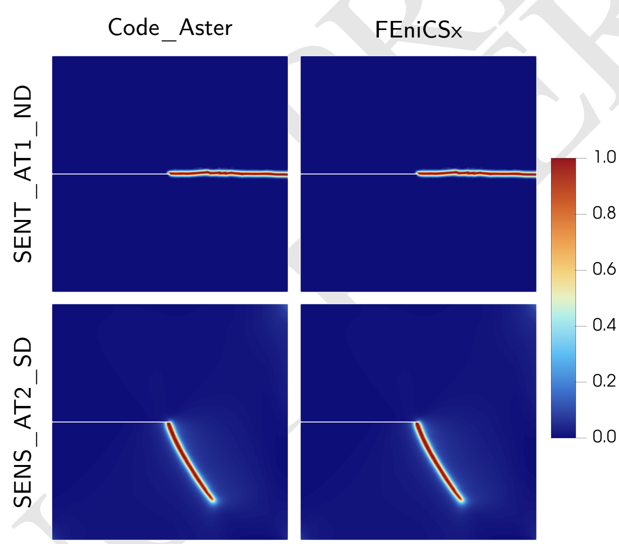


Figure 4: Comparison of final phase-field solutions for selected classical test cases.

4.3. Micromorphic phase-field implementation

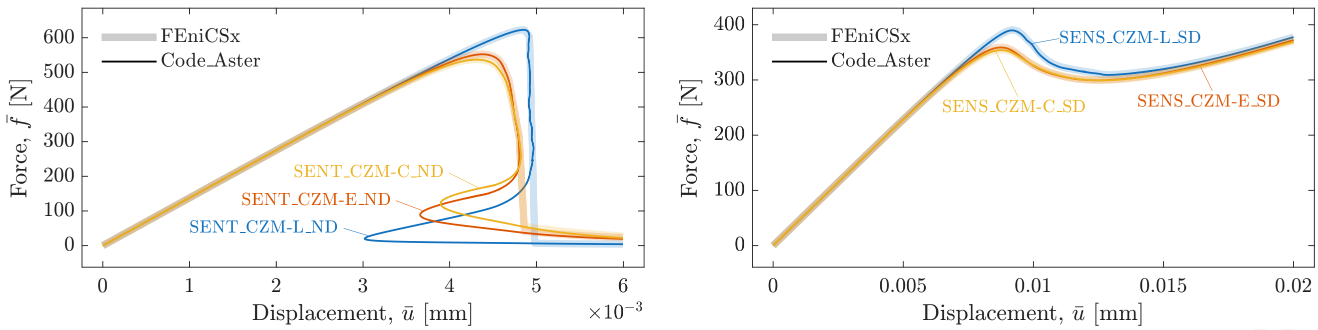
For the micromorphic formulation, the FEniCSx implementation solves the same global equations but is not a strict algorithmic replica, since, for simplicity, the Lagrange multiplier formulation, monolithic solution scheme and path-following procedure adopted in Code_Aster are not reproduced (hence $\beta = 400$ in FEniCSx). Therefore, agreement is assessed in terms of global response and crack pattern rather than pointwise numerical equivalence. In the following, CZM-L, CZM-E and CZM-C denote cohesive-zone models with linear, exponential and Cornelissen softening laws, respectively.

Figure 5 compares the force-displacement responses for the SENT and SENS benchmarks. The Code_Aster results agree closely with FEniCSx up to crack initiation and reproduce the same overall post-cracking behaviour. Because the two implementations use different constraint enforcement and loading strategies, small discrepancies are observed after unstable crack propagation. Representative final micromorphic fields are shown in Fig. 6. The same crack paths are obtained in both implementations, and the observed local differences remain consistent with the expected sensitivity of the post-cracking response.

5. Conclusions

This technical note presented two implementation strategies for phase-field fracture in Code_Aster. The first strategy implements the classical AT1 and AT2 formulations through a heat-transfer analogy and a staggered coupling scheme. This approach is robust and reuses existing thermal finite-element operators in Code_Aster for assembly, while the bound-constrained phase-field problem is solved through the coupling to PETSc. Its main limitation is that it applies only to formulations whose evolution equations can be expressed through the adopted heat-transfer analogy.

The second strategy uses the native `GRAD_VARI` framework to implement a micromorphic phase-field formulation. This route is more flexible, since different phase-field models can be introduced through the constitutive behaviour and was used here for CZM-type models. Its main limitation is the monolithic solution scheme, which may require additional globalisation or control strategies to achieve convergence during unstable crack propagation; in this work, a path-following strategy was employed.



(a) SENT

(b) SENS

Figure 5: Force-displacement responses for the micromorphic test cases.

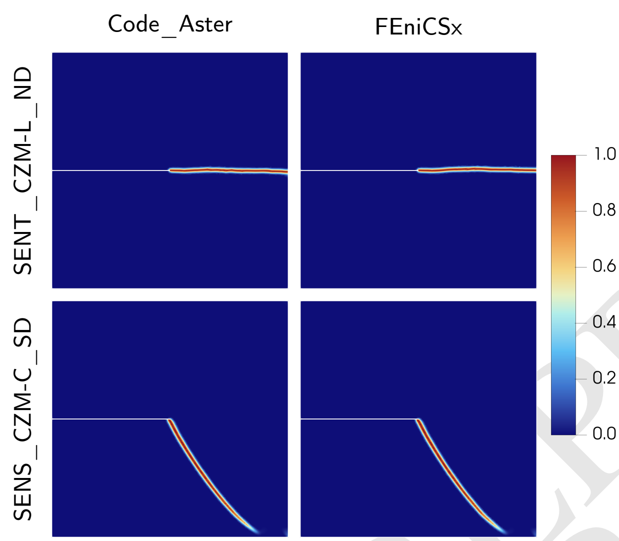


Figure 6: Comparison of final micromorphic solutions for selected micromorphic test cases.

Both implementations were verified against FEniCSx reference solutions for single-edge notched specimens in tension and shear, showing good agreement in force-displacement responses and crack paths. The two approaches are therefore complementary: the heat-transfer analogy is robust but model-restricted, whereas the `GRAD_VARI` formulation is more general but numerically demanding.

Declaration of generative AI in the writing process

During the preparation of this manuscript, the authors used ChatGPT to assist with improving the language and readability of the text, and provide critical feedback on draft versions. The authors reviewed and edited all AI-generated suggestions and takes full responsibility for the content of the published work.

Declaration of competing interest

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

Acknowledgements

The authors gratefully acknowledge Dr. Thomas Helfer (Commissariat à l'énergie atomique et aux Énergies Alternatives (CEA), France) for his guidance on the implementation of the micromorphic approach in MFront. This work has been financially supported by the CP-SENS project (Cyber-Physical Sensing for Machines and Structures), REF. 2081-00006B, 2022-2026, Danish Innovation Fund.

Data availability

The numerical codes and data supporting this study are available at https://doi.org/10.5281/zenodo.21503467 [36]. The repository contains all files and documentation required to reproduce the Code_Aster and FEniCSx benchmark simulations. All software components developed as part of this work are released under the BSD 3-Clause License.

CRediT authorship contribution statement

Jakob Degn Larsen: Conceptualization, Methodology, Software, Validation, Visualization, Writing – original draft. **Basile Marchand:** Methodology, Software, Validation, Writing – review and editing. **Michele Ferraiuolo:** Conceptualization, Methodology, Writing – review and editing. **Giuseppe Abbiati:** Conceptualization, Supervision, Funding acquisition, Writing – review and editing.

References

[1] G. Francfort, J.-J. Marigo, Revisiting brittle fracture as an energy minimization problem, Journal of the Mechanics and Physics of Solids 46 (8) (1998) 1319–1342. doi:10.1016/S0022-5096(98)00034-9.

[2] B. Bourdin, G. Francfort, J.-J. Marigo, Numerical experiments in revisited brittle fracture, Journal of the Mechanics and Physics of Solids 48 (4) (2000) 797–826. doi:10.1016/S0022-5096(99)00028-9.

[3] C. Miehe, F. Welschinger, M. Hofacker, Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations, International Journal for Numerical Methods in Engineering 83 (10) (2010) 1273–1311. doi:10.1002/nme.2861.

[4] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, Computer Methods in Applied Mechanics and Engineering 199 (45-48) (2010) 2765–2778. doi:10.1016/j.cma.2010.04.011.

[5] B. Bourdin, J.-J. Marigo, C. Maurini, P. Sicsic, Morphogenesis and Propagation of Complex Cracks Induced by Thermal Shocks, Physical Review Letters 112 (1) (2014) 014301. doi:10.1103/PhysRevLett.112.014301.

[6] M. Hofacker, C. Miehe, A phase field model of dynamic fracture: Robust field updates for the analysis of complex crack patterns, International Journal for Numerical Methods in Engineering 93 (3) (2013) 276–301. doi:10.1002/nme.4387.

[7] J.-Y. Wu, V. P. Nguyen, C. T. Nguyen, D. Sutula, S. Sinaie, S. P. Bordas, Phase-field modeling of fracture, in: Advances in Applied Mechanics, Vol. 53, Elsevier, 2020, pp. 1–183. doi:10.1016/bs.aams.2019.08.001.

[8] B. Bourdin, Numerical implementation of the variational formulation for quasi-static brittle fracture, Interfaces and Free Boundaries 9 (3) (2007) 411–430. doi:10.4171/ifb/171.

[9] F. Vicentini, P. Carrara, L. De Lorenzis, Phase-field modeling of brittle fracture in heterogeneous bars, European Journal of Mechanics - A/Solids 97 (2023) 104826. doi:10.1016/j.euromechsol.2022.104826.

[10] N. Shauer, Less than 500 lines self-contained Python finite element implementation of the phase-field method for fracture mechanics, Advances in Engineering Software 210 (2025) 104013. doi:10.1016/j.advengsoft.2025.104013.

[11] Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda, A Unified Abaqus Implementation of the Phase Field Fracture Method Using Only a User Material Subroutine, Materials 14 (8) (2021) 1913. doi:10.3390/ma14081913.

[12] S. Zhou, T. Rabczuk, X. Zhuang, Phase field modeling of quasi-static and dynamic crack propagation: COMSOL implementation and case studies, Advances in Engineering Software 122 (2018) 31–49. doi:10.1016/j.advengsoft.2018.03.012.

[13] D. Kosov, A. Tumanov, V. Shlyannikov, ANSYS implementation of the phase field fracture approach, Frattura ed Integrità Strutturale 18 (70) (2024) 133–156. doi:10.3221/IGF-ESIS.70.08.

[14] C. Kuhn, A. Schlüter, R. Müller, On degradation functions in phase field fracture models, Computational Materials Science 108 (2015) 374–384. doi:10.1016/j.commatsci.2015.05.034.

[15] Hirshikesh, S. Natarajan, R. K. Annabattula, A FEniCS implementation of the phase field method for quasi-static brittle fracture, Frontiers of Structural and Civil Engineering 13 (2) (2019) 380–396. doi:10.1007/s11709-018-0471-9.

[16] T. Heister, T. Wick, pfm-cracks: A parallel-adaptive framework for phase-field fracture propagation, Software Impacts 6 (2020) 100045. doi:10.1016/j.simpa.2020.100045.

[17] T. Hu, B. Talamini, A. J. Stershic, M. R. Tupek, J. E. Dolbow, A variational phase-field model For ductile fracture with coalescence dissipation, Computational Mechanics 68 (2) (2021) 311–335. doi:10.1007/s00466-021-02033-1.

[18] J.-Y. Wu, A generalized phase-field cohesive zone model (μpf-czm) for fracture, Journal of the Mechanics and Physics of Solids 192 (2024) 105841. doi:10.1016/j.jmps.2024.105841.

[19] T. You, K. Yoshioka, On poroelastic strain energy degradation in the variational phase-field models for hydraulic fracture, Computer Methods in Applied Mechanics and Engineering 416 (2023) 116305. doi:10.1016/j.cma.2023.116305.

[20] Y. Lu, T. Helfer, B. Bary, O. Fandeur, An efficient and robust staggered algorithm applied to the quasi-static description of brittle fracture by a phase-field approach, Computer Methods in Applied Mechanics and Engineering 370 (2020) 113218. doi:10.1016/j.cma.2020.113218.

[21] S. Riad, D. Bardel, J. Réthoré, Effect of microstructural length scales on crack propagation in elastic Cosserat media, Engineering Fracture Mechanics 267 (2022) 108399. doi:10.1016/j.engfracmech.2022.108399.

[22] A. Jaccon, B. Prabel, G. Molnár, J. Bluthé, A. Gravouil, Adaptive mesh refinement and cycle jumps for phase-field fatigue fracture modeling, Finite Elements in Analysis and Design 224 (2023) 104004. doi:10.1016/j.finel.2023.104004.

[23] É. de France, Finite element *Code_Aster*, Analysis of Structures and Thermomechanics for Studies and Research, Open source at www.code-aster.org (1989–2026).

[24] J.-Y. Wu, A unified phase-field theory for the mechanics of damage and quasi-brittle failure, Journal of the Mechanics and Physics of Solids 103 (2017) 72–99. doi:10.1016/j.jmps.2017.03.015.

[25] F. Vicentini, C. Zolesi, P. Carrara, C. Maurini, L. De Lorenzis, On the energy decomposition in variational phase-field models for brittle fracture under multi-axial stress states, International Journal of Fracture 247 (3) (2024) 291–317. doi:10.1007/s10704-024-00763-w.

[26] H. Amor, J.-J. Marigo, C. Maurini, Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments, Journal of the Mechanics and Physics of Solids 57 (8) (2009) 1209–1229. doi:10.1016/j.jmps.2009.04.011.

[27] K. Pham, H. Amor, J.-J. Marigo, C. Maurini, Gradient Damage Models and Their Use to Approximate Brittle Fracture, International Journal of Damage Mechanics 20 (4) (2011) 618–652. doi:10.1177/1056789510386852.

[28] R. Bharali, F. Larsson, R. Jänicke, A micromorphic phase-field model for brittle and quasi-brittle fracture, Computational Mechanics 73 (3) (2024) 579–598. doi:10.1007/s00466-023-02380-1.

[29] E. Lorentz, V. Godard, Gradient damage models: Toward full-scale computations, Computer Methods in Applied Mechanics and Engineering 200 (21) (2011) 1927–1944. doi:10.1016/j.cma.2010.06.025.

[30] T. Helfer, B. Michel, J.-M. Proix, M. Salvo, J. Sercombe, M. Casella, Introducing the open-source mfront code generator: Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform, Computers & Mathematics with Applications 70 (5) (2015) 994–1023. doi:10.1016/j.camwa.2015.06.027.

[31] E. Lorentz, Lois de comportement à gradients de variables internes: construction, formulation variationnelle et mise en œuvre numérique, Ph.D. thesis, Université Pierre et Marie Curie (Paris VI) (1999).

[32] T. Gerasimov, L. De Lorenzis, A line search assisted monolithic approach for phase-field computing of brittle fracture, Computer Methods in Applied Mechanics and Engineering 312 (2016) 276–303. doi:10.1016/j.cma.2015.12.017.

[33] F. Loiseau, V. Lazarus, Path-following methods for phase-field simulation of quasi-static crack propagation, International Journal of Solids and Structures 334 (2026) 113974. doi:10.1016/j.ijsolstr.2026.113974.

[34] I. A. Baratta, J. P. Dean, J. S. Dokken, M. Habera, J. S. Hale, C. N. Richardson, M. E. Rognes, M. W. Scroggs, N. Sime, G. N. Wells, DOLFINx: the next generation FEniCS problem solving environment, preprint, Zenodo (2023). doi:10.5281/zenodo.10447666.

[35] M. S. Alnaes, A. Logg, K. B. Ølgaard, M. E. Rognes, G. N. Wells, Unified form language: A domain-specific language for weak formulations of partial differential equations, ACM Transactions on Mathematical Software 40 (2) (2014). doi:10.1145/2566630.

[36] J. Degn Larsen, B. Marchand, M. Ferraiuolo, G. Abbiati, Accompanying code and data for the article "Open-Source Implementation of Classical and Micromorphic Phase-Field Fracture Models in Code_Aster", Zenodo (2026). doi:10.5281/zenodo.21503467.