SoftwareX 24 (2023) 101594

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

GPFniCS: A generalised phase field method to model fracture

Manish Kumar [a,*], Roberto Alessi [b], Enrico Salvati [a]

[a] Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, Via delle Scienze 206, Udine 33100, Italy
[b] Department of Civil and Industrial Engineering, University of Pisa, Largo Lucio Lazzarino 2, 56122 Pisa, Italy

ARTICLE INFO

Keywords:
Phase field method
Cohesive zone phase field method
Crack Propagation
Brittle Fracture
FEniCS

ABSTRACT

Advanced damage tolerance design of materials and mechanical components heavily relies on fracture failure analysis. A robust, efficient, and versatile software (GPFniCS) *is developed and provided for public access to perform fracture analyses based on the Generalised Phase-Field Method.* GPFniCS *software is developed on top of* FEniCS, *an open-source finite element library. One-dimensional and two-dimensional mixed mode problems are validated with* GPFniCS *and provided as illustrative examples in a public repository. The software shows excellent potential for computational fracture studies, and it is open to further developments in various fields like thermal loading, fatigue loading, solidification, and many more.*

Code Metadata

Nr	Code metadata description	Please fill in this column
C1	Current code version	v 1.0
C2	Permanent link to code/repository used for this code version	https://github.com/Manishkumar923/GPFniCS.git
C3	Permanent link to reproducible capsule	None
C4	Legal code license	GNU General Public License v3.0
C5	Code versioning system used	None
C6	Software code languages, tools and services used	Python3, FEniCS, and ParaView
C7	Compilation requirements, operating environments and dependencies	Linux
C8	If available, link to developer documentation/manual	https://github.com/Manishkumar923/GPFniCS/blob/main/README.md
C9	Support email for questions	mkumar2@me.iitr.ac.in

1. Motivation and significance

In today's era of modern industrialisation, the use of resources up to their maximum capacity is a key objective to reduce costs and impact on the environment. To achieve these goals, designing mechanical components/machines for realistic operating conditions plays a significant role. The computer-aided designing and numerical analysis software such as stress analysis, fracture failures, life prediction, tear and wear analyses, temperature effects, and multi-physics coupling help for better designs. Among these analyses, numerical studies of fracture failures received a remarkable attraction because these can estimate the structural life of engineering products and assess to which extent these can operate in the presence of damage (or nucleated cracks), i.e., during stable crack propagation. GPFniCS *is developed to assist structural engineers and scientists to simulate and predict fracture failures numerically.*

The methods used for the numerical analysis of fracture are broadly categorised into discrete and smeared methods [1], referred to as the numerical representation of cracks. The most commonly used numerical analysis method is Finite Element Method (FEM), which itself is a discrete method. However, FEM is not a well-suited option for fracture analysis as the discretisation of geometry should be aligned with the crack propagation path, and after every crack propagation step, re-discretisation along with data transfer from old to new discretisation is necessary. These issues can be overcome by advanced variations of discrete methods like the eXtended Finite Element Method (XFEM), boundary element method, meshfree method, and many more [2]. The analysis for the cases of crack nucleation, crack branching, complex crack paths, and crack merging are still tricky tasks in discrete methods [3]. On the other hand, smeared approaches like the Phase Field Method (PFM) or gradient-enhanced damage models can tackle these issues [4-5]. In smeared methods, the crack is modelled as a variable in a diffusive way, and the value of this variable, generally ranging from 0 to 1, describes the presence of crack. PFM is based on the variational formulation of Griffith's theory. In PFM, a smooth transition from crack to bulk is devised, which allows capturing several phenomena, like crack branching, merging, and interplay with other physics of the problem. PFM acquired a lot of recognition because it is robust, efficient, and easy to implement. FEM is

* Corresponding author.
E-mail address: manish.kumar@uniud.it (M. Kumar).

https://doi.org/10.1016/j.softx.2023.101594
Received 20 September 2023; Received in revised form 20 November 2023; Accepted 20 November 2023
Available online 2 December 2023

usually employed to couple and solve the continuum mechanics problems with PFM, in which the crack is traced automatically with the value of the phase field on the same discretised geometry [6]. In PFM, a characteristic length is defined, which controls the smeared crack width description. A small value of this length is required for a sharp crack representation, which demands a fine discretisation and makes numerical analysis computationally expensive [7]. Therefore, an efficient discretisation of geometry is essential for PFM [8].

2.1. Software architecture

The governing equation of PFM consists of stored energy (strain energy) and dissipation energy, as shown in Eq. (1) [9,18]. The strain energy is reduced by the degradation function, which is a function of the phase field variable. The dissipated energy depends on the phase field dissipation functional.

$$\vartheta(\boldsymbol{u},\phi) = \int_{\Omega} \overbrace{\overbrace{\omega(\phi)}^{\text{Degradation function}}}^{\text{Stored Energy}} \psi_0(\varepsilon(\boldsymbol{u}))\, dV + \int_{\Omega} \overbrace{\frac{G_c}{c_w l_0}\left(\overbrace{\gamma(\phi)}^{\text{Phase field evolution function}} + l_0^2 \nabla\phi.\nabla\phi\right)}^{\text{Dissipation Energy}} dV - \int_{\partial\Omega_T} \overbrace{\overbrace{F_{ext}.\boldsymbol{u}}^{\text{Externally applied energy}}} dS \tag{1}$$

The classical PFM was initially developed for purely brittle fracturing processes that have been widely exploited and applied to materials showing a homogeneous linear elastic behaviour. However, quasi-brittle fracture modelling is still not well explored. Recently, a novel constitutive phase field model based on classical PFM was proposed, which can reduce to the cohesive zone model as the characteristic length tends to zero. Thus, this modified PFM is named as cohesive zone phase field method (CZ-PFM). In CZ-PFM, general types of softening, like linear, exponential, and hyperbolic, can be modelled by adjusting parameters in the total energy term [9], as explained in the next section. The same approach is implemented for gradient-enhanced damage model [10] and multi-physics problems [11].

CZ-PFM has a great potential to model fracture failures in composites for various loading conditions and coupling with other physics. However, no public software/codes to perform simulations using CZ-PFM are available. Therefore, a generalised software, GPFniCS, is developed and made available to the interested community to perform fracture analyses using PFM/CZ-PFM while helping future researchers to develop new features. This software is based on the open-source library FEniCS. FEniCS is an intuitive, flexible, and efficient library to solve partial differential equations by FEMs [12]. With the current version, GPFniCS can simulate fracture failure in two-dimensional linear elastic isotropic materials subject to monotonic loading. It can compute mode I, mode II, and mixed mode fractures. GPFniCS is validated for various cases and presented in this article for the sake of reference. GPFniCS can be further extended to three-dimensional geometries [13], including the effect of other physics, different types of loading like fatigue [14-15], residual stresses [16-17], and material damaging phenomena. The article consists of five major sections. Section 1 describes the scientific importance and significance of GPFniCS and its capabilities. Section 2 contains the architecture and key components of GPFniCS, along with details of the pre-processing and post-processing steps. The validations and illustrative examples of GPFniCS are provided in Section 3. In Section 4, the contributions and capabilities of GPFniCS are presented. At last, conclusions are given in Section 5.

2. Software description

GPFniCS is developed in the Python 3.8.10 environment. The whole code can be downloaded from https://github.com/Manishkumar923/GPFniCS.git. This code is based on the FEniCS and NumPy libraries available on https://fenicsproject.org/download/ and https://numpy.org/install/, respectively. For post-processing, open-source software ParaView is used for contour plotting, while line plots are generated via Python 3.8.10. The installation procedure for FEniCS can be found at https://docs.fenicsproject.org/dolfinx/main/python/installation.html#source. The architecture and functionalities of GPFniCS are explained in the following sub-sections.

The function and functional control the existence and extent of the softening regime by changing the values of involved parameters, which can be found in [9]. The fundamental detail of the PFM is provided in the Github repository that is available on https://github.com/Manishkumar923/GPFniCS/blob/main/Phase_Field_Method.pdf. In current work, the crack growth is considered to be triggered only by the presence of tensile stresses; therefore, strain is decomposed into two parts: positive and negative parts, using strain spectral decomposition [19-20]. These are the fundamental aspects used in the development of GPFniCS.

The key components and architecture of GPFniCS are shown in Fig. 1 [21]. The material properties and discretised geometry are the primary inputs for GPFniCS. The current version supports only the T3 (Linear triangular element) element type. Therefore, the geometry must be discretised using either structured or unstructured T3 elements. First, the required libraries are imported into the software, followed by defining the material parameters. The discretised geometry is called in .xml format using the mesh command in GPFniCS. A subdomain is created to calculate reaction force, and boundaries are identified to apply boundary conditions. The function spaces and element type are defined according to the unknown fields; for instance, the displacement field is a vector, and the phase field is a scalar. Therefore, in the provided version of GPFniCS, function spaces are defined with Continuous Galerkin (CG: standard Lagrange family) element type of degree one using VectorFunctionSpace and FunctionSpace commands for displacement and phase fields, respectively. According to the governing equations, Functions, TrialFunctions, and TestFunctions are declared using the earlier defined function spaces. Several Python functions are defined to perform various computations such as eps, sigma_0, sigma, a, w, eps_positive, and eps_negative, respectively, for strain, stress, degraded stress, strain energy degradation function, dissipation functional, positive part of strain, and negative part of strain.

The upper and lower limits of the phase field (0 to 1) are initially declared on the phase field function space. Phase field boundary conditions are also applied to these limits. Strong forms of the governing equations are defined using the pre-defined functions, and their derivatives are computed using the derivative operator of FEniCS with respect to trial and test functions. In GPFniCS, the displacement and phase field are solved through a staggered approach, given the higher efficiency than the monolithic approach. In the staggered approach, both fields are calculated in an iterative way, which means the converged displacement field is used to compute the phase field and the newly calculated phase field is used to re-evaluate the displacement field. This process continues until convergence is achieved for the phase field. NonlinearVariationalSolver class with mumps solver and PETScTAOSolver class with umfpack solver are used to compute displacement and phase fields, respectively. Many other solvers are available in FEniCS, but these turned out to be best suited for this problem to date, as comprehensively explored by the authors. At the end of each load increment, displacement and phase fields are saved in a .pvd file, which can be opened in ParaView for

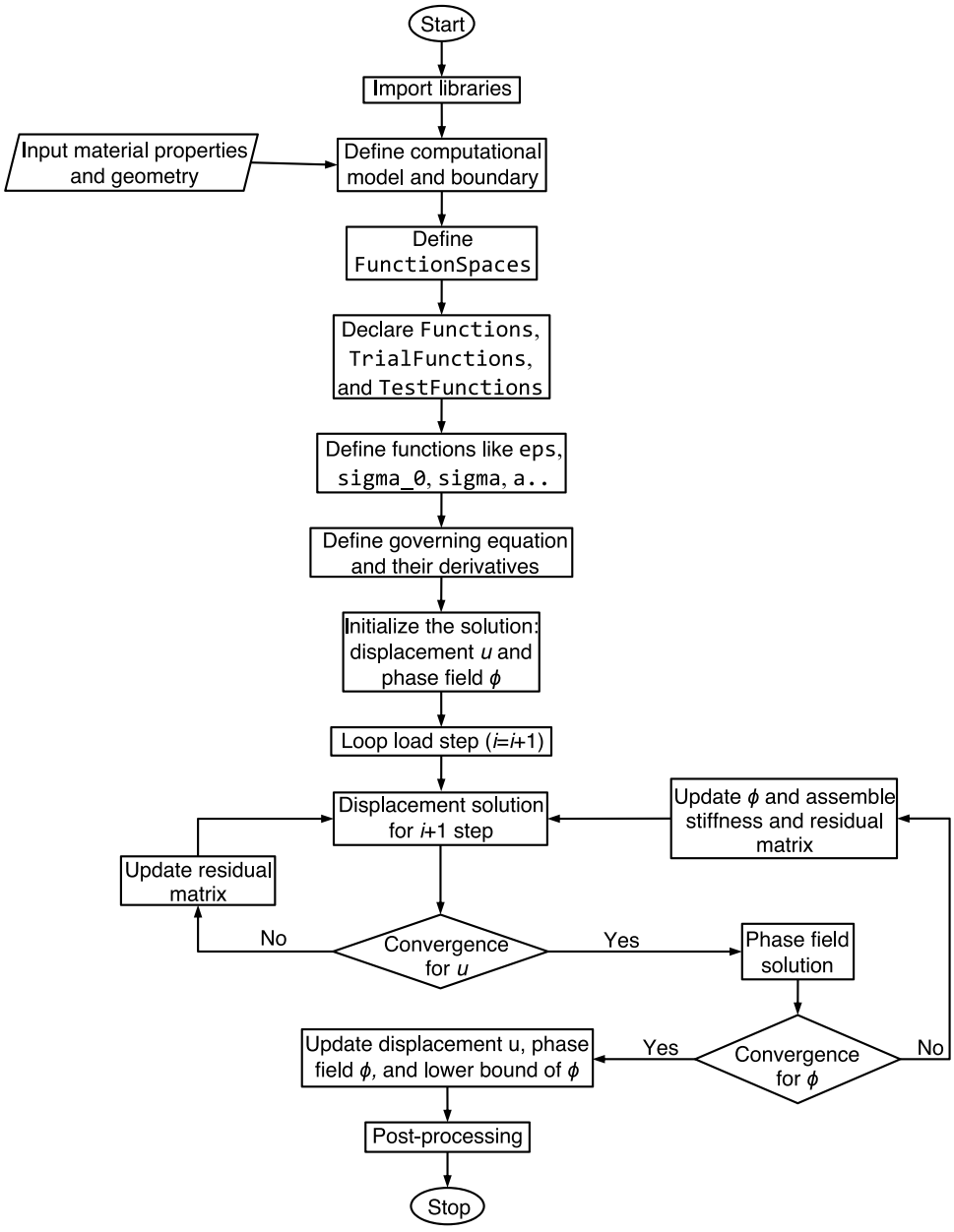

Fig. 1. Outline of key components and architecture of the GPFniCS.

graphical visualisation. The crack propagation path can be easily shown by the phase field contour plots. For plotting the load-vs-displacement, load and displacement values at the interested location are recorded in a text file. This load-displacement data can be used in any plotting tool as per the user's choice. The final step before the new load step is to update the lower limit of the phase field. In the formulation of PFM, the propagated cracks cannot heal, so this effect is enforced by imposing a phase field that never lowers its actual value, so the current phase field is assigned as the lower bound. The PETScTAOSolver *class provides the solution in the bounded limits. Updating the lower bound of the phase field removes the requirement for the history variable used in the available phase field codes.*

2.2. Software functionalities

The discretised geometry and material properties are primary inputs for GPFniCS*. The displacement boundary conditions (Neumann and Dirichlet) and phase field boundary conditions must change according to the problem. In the terminal, use the command:* python3 filename.py *to execute the code. In the provided version of* GPFniCS*, a variable load step is implemented to speed up the computation in loading regimes where the fracture process is not supposed to occur. For instance, linear elastic behaviour is expected during the initial loading time, which is obtained via large load steps, while near the peak and softening regime, small load steps are used to capture nonlinear effects accurately. Fields like stress or strain can also be calculated by defining the tensor function space (*TensorFunctionSpace*) with one degree less than the displacement function space. Before saving the stress/strain .pvd file, it is required to transfer it to tensor function space using project operator. The computed fields can be exported as .pvd, as mentioned in the previous subsection. This Python script to plot the force-displacement curve is also provided with the* GPFniCS *library.*

3. Illustrative examples

The developed code GPFniCS *is validated with the one-dimensional and two-dimensional mixed mode fracture specimens of linear isotropic elastic material subjected to monotonic loading. A uniaxial bar subjected to tensile*

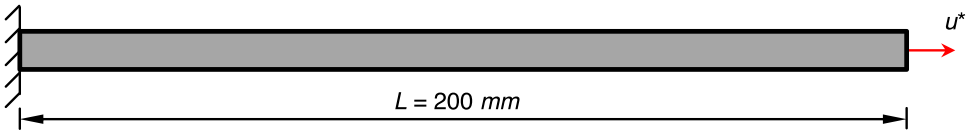

Fig. 2. Schematic illustration of a uniaxial bar in tension with boundary conditions.

load is analysed for linear and Cornelissen softening. A three-point bend and L-shaped specimens are examined and compared to the experimental observations to validate the software. The details of these studies are as follows:

3.1. Uniaxial bar in tension

A one-dimensional uniform bar of unit cross-sectional area and 200 mm length is simulated using the developed GPFniCS *to obtain the load-displacement behaviour. The vertical left-hand-side edge of the bar is fixed, and a displacement is applied on the right end of the bar, as shown in Fig. 2. The material properties of the bar are taken as Young's modulus (E_0) = 30.0 GPa, Poisson ratio (ν_0) = 0.2, fracture strength (f_t) = 3.0 MPa and fracture energy (G_f) = 0.12 N/mm. Two types of softening (Linear and Cornelissen) are simulated for various values of characteristic length (b) and mesh element size (h), whose details are reported in Tables 1 and 2. A displacement of 0.120 mm and 0.216 mm is applied in the case of linear and Cornelissen softening, respectively. The simulated linear softening and Cornelissen softening results are shown in Figs. 3 and 4. The numerical results show a similar trend as reported in the literature [9].*

3.2. Three-point bend specimen

A three-point bend specimen of 450 mm length is taken to validate the GPFniCS, *as shown in Fig. 5(a). The dimensions, boundary conditions, and the considered physical notch in the specimen are also provided in Fig. 5(a). The material properties are considered as E_0 = 20.0 GPa, ν_0 = 0.2, G_f = 0.113 N/mm and f_t = 2.4 MPa. The mesh density in the probable failure region is taken high so that the size of the elements is of the same order of characteristic length. The geometry of the three-point bend specimen is discretised into 12,974 three-noded linear triangular elements (T3), and the element's size in the potential failure area is 0.25 mm. A displacement of 1.0 mm is applied in a ramped fashion. The simulations for two different characteristic lengths (b = 1.5 mm and 2.5 mm) are performed, and the load vs displacement response and crack growth path for the Cornelissen softening model are shown in Fig. 5(b) and (c), respectively. The numerical results of load vs displacement behaviour for maximum load and dissipation within experimental bounds and show an even better match than the literature [9].*

Table 1
Mesh size and characteristic length used for linear softening simulation.

Case	b/h	b	h	number of elements	a_1	a_2	a_3
L_10_5	10	5	0.5	400	320/π	−0.5	0
L_20_5	20	5	0.25	800	320/π	−0.5	0
L_50_5	50	5	0.1	2000	320/π	−0.5	0
L_100_5	100	5	0.05	4000	320/π	−0.5	0
L_200_5	200	5	0.025	8000	320/π	−0.5	0
L_10_10	10	10	1.0	200	160/π	−0.5	0
L_20_10	20	10	0.5	400	160/π	−0.5	0
L_50_10	50	10	0.2	1000	160/π	−0.5	0
L_100_10	100	10	0.1	2000	160/π	−0.5	0
L_200_10	200	10	0.05	4000	160/π	−0.5	0
L_10_20	10	20	2.0	100	80/π	−0.5	0
L_20_20	20	20	1.0	200	80/π	−0.5	0
L_50_20	50	20	0.4	500	80/π	−0.5	0
L_100_20	100	20	0.2	1000	80/π	−0.5	0
L_200_20	200	20	0.1	2000	80/π	−0.5	0
L_10_50	10	50	5	40	32/π	−0.5	0
L_20_50	20	50	2.5	80	32/π	−0.5	0
L_50_50	50	50	1	200	32/π	−0.5	0
L_100_50	100	50	0.5	400	32/π	−0.5	0
L_200_50	200	50	0.25	800	32/π	−0.5	0

3.3. L-shaped specimen

The L-shaped specimen is considered to validate the mixed mode capabilities of GPFniCS. *The dimensions and the boundary conditions used in the simulation are presented in Fig. 6(a), along with the expected failure region for reference. The material properties are considered as E_0 = 25.85 GPa, ν_0 = 0.18, G_f = 0.09 N/mm and f_t = 2.7 MPa. To study the effect of mesh density, sizes of mesh elements 1 mm and 0.5 mm are considered in the potential area of failure. Therefore, the specimen's geometry is discretised into 123,059 and 36,119 T3 elements for 0.5 mm and 1.0 mm mesh sizes. The effect of characteristic length is also studied with different characteristic lengths of 5 mm and 10 mm for the mesh of 0.5 mm. A displacement of 0.84 mm is applied for all the cases. The load* vs *displacement response and crack growth path for the Cornelissen softening model are shown in Fig. 6(b) and (c), respectively. The numerical results are very close to the numerical studies in the literature [9].*

A deviation is observed from the experimental bounds with the material properties used in the simulations. In literature [22], some studies are performed with reduced Young's modulus for the same problem to match initial stiffness, which is not considered in the presented study. The numerically computed crack propagation path lies in the expected region predicted by Ambati [7] and is depicted in orange in Fig. 6(a).

4. Impact

GPFniCS *is a novel and robust software to perform numerical simulations for fracture failures. The capabilities of the* GPFniCS *have the following impact and possibilities for further improvements:*

Table 2
Mesh size and characteristic length used for Cornelissen softening simulation.

case	b/h	b	H	number of elements	a_1	a_2	a_3
C_10_5	10	5	0.5	400	320/π	1.3868	0.6567
C_20_5	20	5	0.25	800	320/π	1.3868	0.6567
C_50_5	50	5	0.1	2000	320/π	1.3868	0.6567
C_100_5	100	5	0.05	4000	320/π	1.3868	0.6567
C_200_5	200	5	0.025	8000	320/π	1.3868	0.6567
C_10_10	10	10	1.0	200	160/π	1.3868	0.6567
C_20_10	20	10	0.5	400	160/π	1.3868	0.6567
C_50_10	50	10	0.2	1000	160/π	1.3868	0.6567
C_100_10	100	10	0.1	2000	160/π	1.3868	0.6567
C_200_10	200	10	0.05	4000	160/π	1.3868	0.6567
C_10_20	10	20	2.0	100	80/π	1.3868	0.6567
C_20_20	20	20	1.0	200	80/π	1.3868	0.6567
C_50_20	50	20	0.4	500	80/π	1.3868	0.6567
C_100_20	100	20	0.2	1000	80/π	1.3868	0.6567
C_200_20	200	20	0.1	2000	80/π	1.3868	0.6567
C_10_50	10	50	5	40	32/π	1.3868	0.6567
C_20_50	20	50	2.5	80	32/π	1.3868	0.6567
C_50_50	50	50	1	200	32/π	1.3868	0.6567
C_100_50	100	50	0.5	400	32/π	1.3868	0.6567
C_200_50	200	50	0.25	800	32/π	1.3868	0.6567

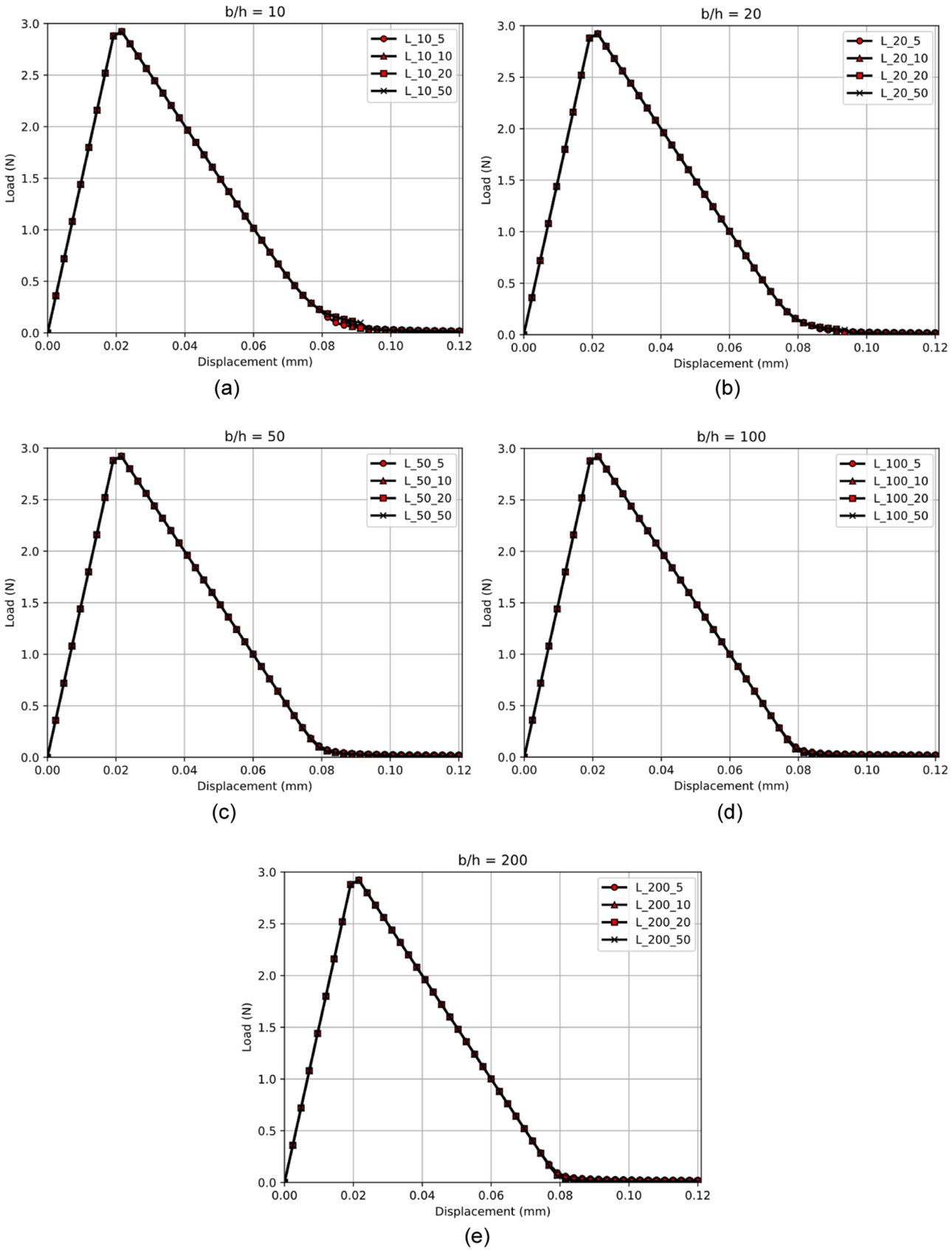

Fig. 3. Load vs displacement plot for a uniaxial bar for linear softening (a) $b/h = 10$, (b) $b/h = 20$, (c) $b/h = 50$, (d) $b/h = 100$, and (e) $b/h = 200$.

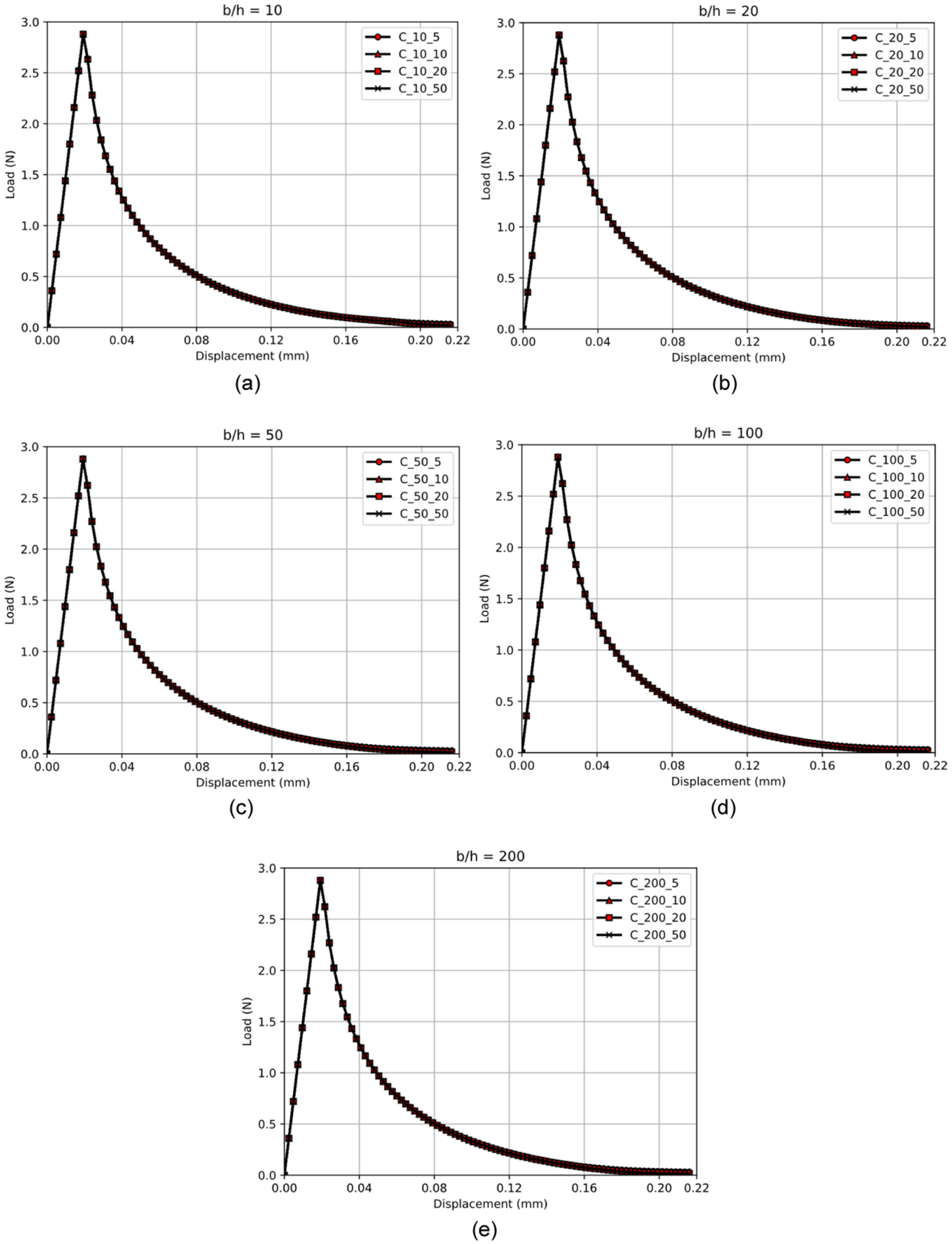

Fig. 4. Load vs displacement plot for a uniaxial bar for Cornelissen softening (a) $b/h = 10$, (b) $b/h = 20$, (c) $b/h = 50$, (d) $b/h = 100$, and (e) $b/h = 200$.

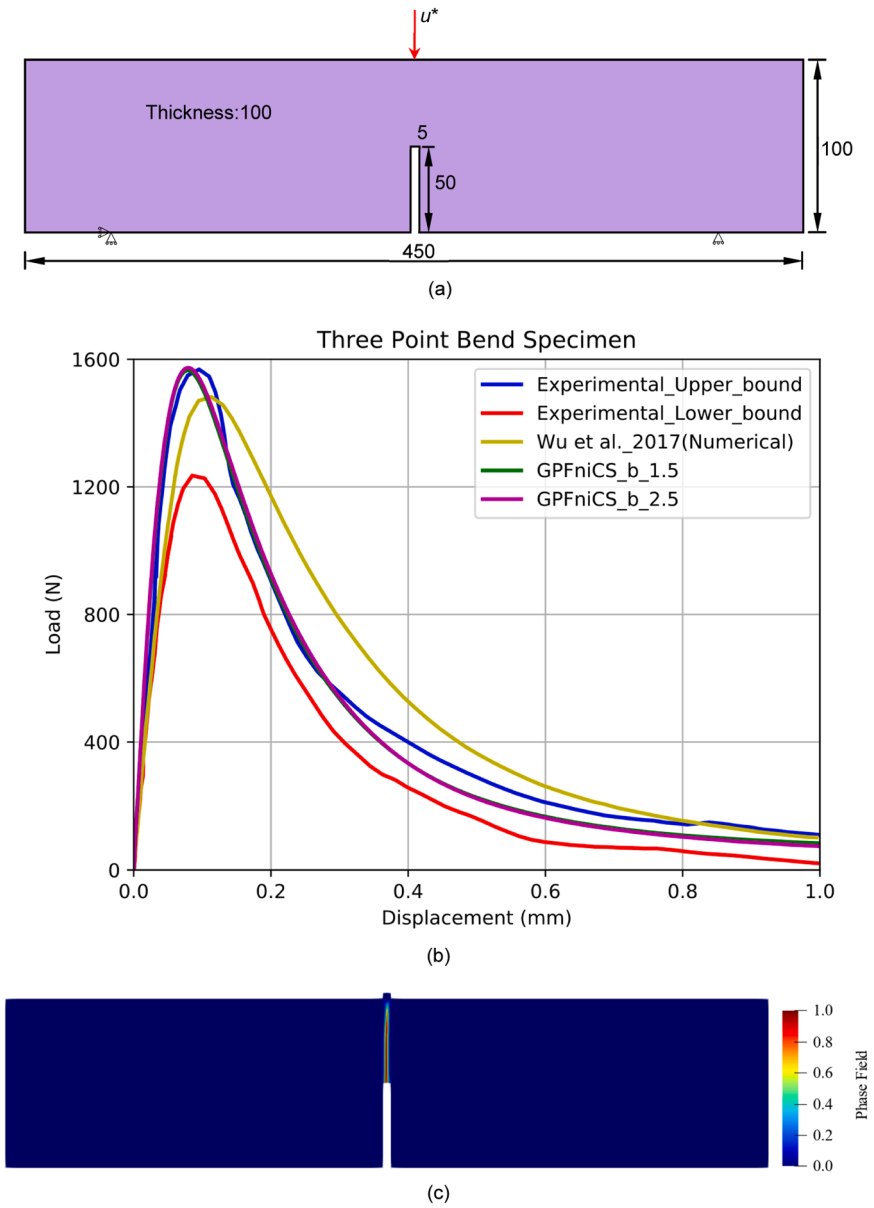

Fig. 5. Three-point bent specimen (a) Specimen with boundary conditions (dimensions in mm) (b) Load vs displacement plot for Cornelissen softening (c) Numerical computed crack growth path.

- GPFniCS *provides public access to software to perform the fracture failures analysis with PFM/CZ-PFM [7,9], which was not available till now.*
- *With* GPFniCS*, it is possible to model brittle and quasi-brittle fractures with general types of softening like linear, exponential, and hyperbolic [9-11].*
- *It will work as the basic code for the user to perform fracture failures subjected to monotonic loading and can be further extended to include the effect of another loading like thermal [23], fatigue [19, 24], creep [25], impact, and combination of these.*
- *The material's non-linearity and other physics like shape memory alloy [26] and solidification can be incorporated into the* GPFniCS *to increase the users of the developed software.*
- GPFniCS *will encourage the material developer to use PFM/CZ-PFM based simulation to understand fracture mechanisms better and improve the material design.*

5. Conclusions

A FEniCS-based efficient software is developed to perform the fracture failure analyses using PFM/CZ-PFM. It can model brittle and quasi-brittle fractures with general types of softening like linear, exponential, and hyperbolic. The spectral decomposition of strain is implemented to prevent crack propagation while applying compressive stresses. The validation of one and two-dimensional mixed mode cases is presented in the article to show the capabilities of the GPFniCS*. The general mesh file* .xml *is used as input for the analysis, and* .pvd *files are extracted for the post-processing in the ParaView.* GPFniCS *can be potentially extended to account for the effect of various types of loading like thermal, fatigue, creep, and physics like material non-linearity, solidification, and many more phenomena.*

CRediT authorship contribution statement

Manish Kumar: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Roberto Alessi:** Conceptualization, Investigation, Methodology, Supervision, Validation, Writing – review & editing. **Enrico Salvati:** Conceptualization, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

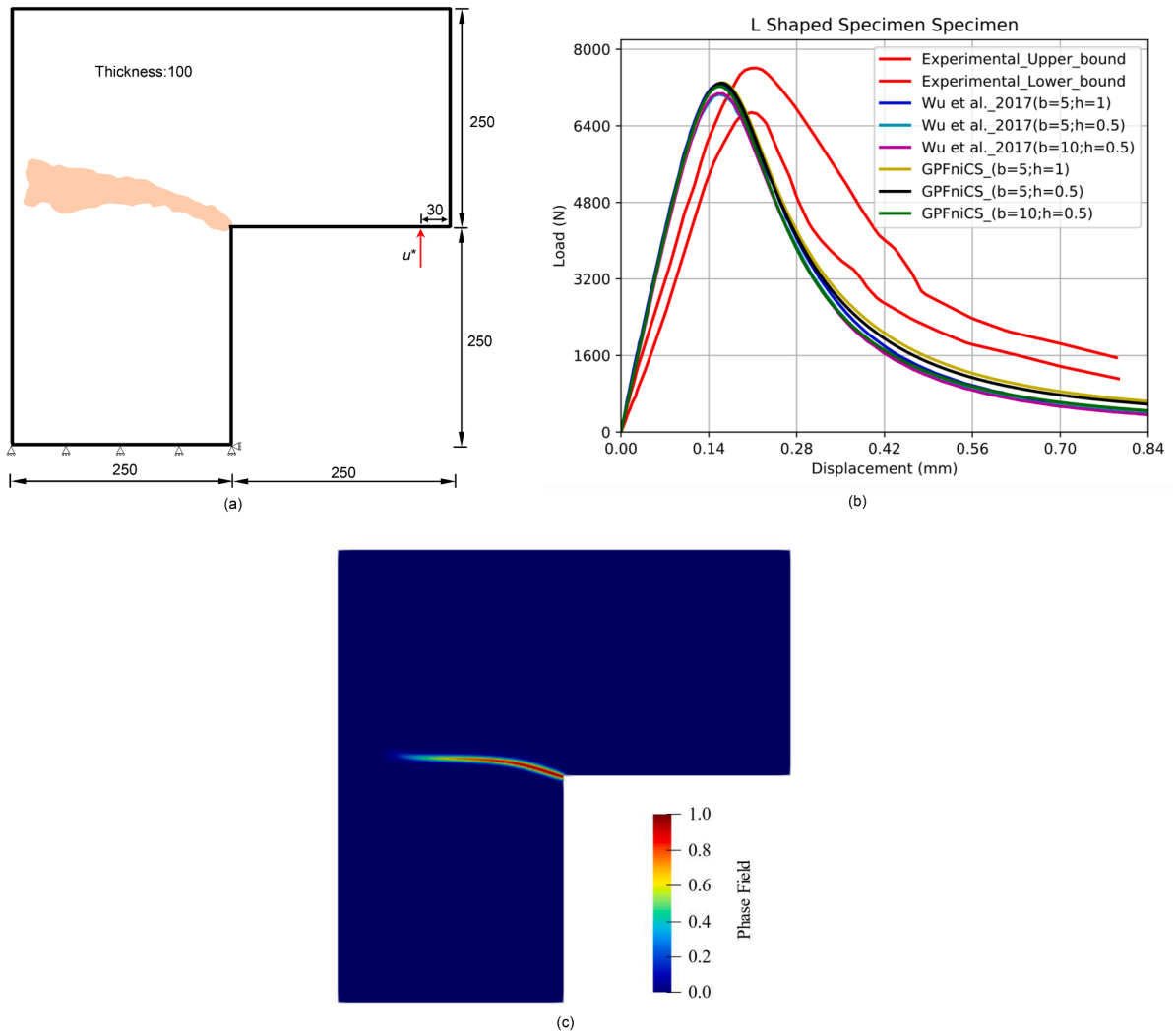

Fig. 6. L-shaped specimen (a) Specimen with boundary conditions (dimensions in mm) (b) Load vs displacement plot for Cornelissen softening (c) Numerical computed crack growth path.

Declaration of Competing Interest

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

Data availability

Data is available on the following link of Google Drive. https://drive.google.com/drive/folders/1g1CX7jl_jyyuMiuMYrb-fnNHsFEiliyu?usp=drive_link

The data and codes are shared on git-hub and google drive. The links are provided in manuscript.

Funding support

This work has been supported by the project "CONCERTO - Multi-scale modelling/characterisation and fabrication of nanocomposite ceramics with improved toughness" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2020- grant 2020BN5ZW9.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.softx.2023.101594.

References

[1] Patil RU, Mishra BK, Singh IV. An adaptive multiscale phase field method for brittle fracture. Comput Methods Appl Mech Eng 2018;329:254–88. https://doi.org/10.1016/j.cma.2017.09.021.
[2] Kumar M, Singh IV. Numerical investigation of creep crack growth in plastically graded materials using C(t) and XFEM. Eng Fract Mech 2020;226:106820. https://doi.org/10.1016/j.engfracmech.2019.106820.
[3] Patil RU, Mishra BK, Singh IV. A multiscale framework based on phase field method and XFEM to simulate fracture in highly heterogeneous materials. Theor Appl Fract Mech 2019;100:390–415. https://doi.org/10.1016/j.tafmec.2019.02.002.
[4] Hansen-Dörr AC, Dammaß F, de Borst R, Kästner M. Phase-field modeling of crack branching and deflection in heterogeneous media. Eng Fract Mech 2020;232:107004. https://doi.org/10.1016/j.engfracmech.2020.107004.
[5] Zhou S, Zhuang X, Zhu H, Rabczuk T. Phase field modelling of crack propagation, branching and coalescence in rocks. Theor Appl Fract Mech 2018;96:174–92. https://doi.org/10.1016/j.tafmec.2018.04.011.
[6] Farrahi GH, Javanbakht M, Jafarzadeh H. On the phase field modeling of crack growth and analytical treatment on the parameters. Contin Mech Thermodyn 2020;32(3):589–606. https://doi.org/10.1007/s00161-018-0685-z.
[7] Ambati M, Gerasimov T, De Lorenzis L. A review on phase-field models of brittle fracture and a new fast hybrid formulation. Comput Mech 2015;55(2):383–405. https://doi.org/10.1007/s00466-014-1109-y.

[8] Freddi F, Mingazzi L. Mesh refinement procedures for the phase field approach to brittle fracture. Comput Methods Appl Mech Eng 2022;388:114214. https://doi.org/10.1016/j.cma.2021.114214.
[9] Wu JY. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. J Mech Phys Solids 2017;103:72–99. https://doi.org/10.1016/j.jmps.2017.03.015.
[10] Wu JY. A geometrically regularized gradient-damage model with energetic equivalence. Comput Methods Appl Mech Eng 2018;328:612–37. https://doi.org/10.1016/j.cma.2017.09.027.
[11] Chen WX, Wu JY. Phase-field cohesive zone modeling of multi-physical fracture in solids and the open-source implementation in Comsol Multiphysics. Theor Appl Fract Mech 2022;117:103153. https://doi.org/10.1016/j.tafmec.2021.103153.
[12] Langtangen H.P., Logg A. Solving PDEs in Python: The FEniCS Tutorial I. Springer Nature. https://doi.org/10.1007/978-3-319-52462-7.
[13] Préve D, Lenarda P, Maskery I, Paggi M. A comprehensive characterization of fracture in unit cell open foams generated from Triply Periodic Minimal Surfaces. Eng Fract Mech 2023;277:108949. https://doi.org/10.1016/j.engfracmech.2022.108949.
[14] Alessi R, Vidoli S, De Lorenzis L. A phenomenological approach to fatigue with a variational phase-field model: the one-dimensional case. Eng Fract Mech 2018;190:53–73. https://doi.org/10.1016/j.engfracmech.2017.11.036.
[15] Ulloa J, Wambacq J, Alessi R, Degrande G, François S. Phase-field modeling of fatigue coupled to cyclic plasticity in an energetic formulation. Comput Methods Appl Mech Eng 2021;373:113473. https://doi.org/10.1016/j.cma.2020.113473.
[16] Salvati E. Residual stress as a fracture toughening mechanism: a Phase-Field study on a brittle material. Theor Appl Fract Mech 2021;114:103021. https://doi.org/10.1016/j.tafmec.2021.103021.
[17] Salvati E, Menegatti F, Kumar M, Pelegatti M, Tognan A. On the significance of diffuse crack width self-evolution in the phase-field model for residually stressed brittle materials. Mater Des Process Commun 2021;3(5):e261. https://doi.org/10.1002/mdp2.261.
[18] Kumar M, Salvati E. Advanced numerical methods for fracture assessment. Reference module in materials science and materials engineering. Elsevier; 2023. https://doi.org/10.1016/B978-0-323-90646-3.00010-1.
[19] Carrara P, Ambati M, Alessi R, De Lorenzis L. A framework to model the fatigue behavior of brittle materials based on a variational phase-field approach. Comput Methods Appl Mech Eng 2020;361:112731. https://doi.org/10.1016/j.cma.2019.112731.
[20] Kienle D, Aldakheel F, Keip MA. A finite-strain phase-field approach to ductile failure of frictional materials. Int J Solids Struct 2019;172:147–62. https://doi.org/10.1016/j.ijsolstr.2019.02.006.
[21] Zhang T, Yu T, Xing C, Bui TQ. Implementation of the adaptive phase-field method with variable-node elements for cohesive fracture. Adv Eng Softw 2023;186:103554. https://doi.org/10.1016/j.advengsoft.2023.103554.
[22] Zamani A, Gracie R, Reza Eslami M. Cohesive and non-cohesive fracture by higher-order enrichment of XFEM. Int J Numer Methods Eng 2012;90(4):452–83. https://doi.org/10.1002/nme.3329.
[23] Crosby T, Ghoniem N. Phase-field modeling of thermomechanical damage in tungsten under severe plasma transients. Comput Mech 2012;50(2):159–68. https://doi.org/10.1007/s00466-012-0733-7.
[24] Kumar M, Bhuwal AS, Singh IV, Mishra BK, Ahmad S, Rao AV, Kumar V. Nonlinear fatigue crack growth simulations using J-integral decomposition and XFEM. Procedia Eng 2017;173:1209–14. https://doi.org/10.1016/j.proeng.2016.12.126.
[25] Kumar M, Singh IV, Mishra BK, Ahmad S, Rao AV, Kumar V. Mixed mode crack growth in elasto-plastic-creeping solids using XFEM. Eng Fract Mech 2018;199:489–517. https://doi.org/10.1016/j.engfracmech.2018.05.014.
[26] Simoes M, Martínez-Pañeda E. Phase field modelling of fracture and fatigue in Shape Memory Alloys. Comput Methods Appl Mech Eng 2021;373:113504. https://doi.org/10.1016/j.cma.2020.113504.