



HAL
open science

PhaFiDyn: An explicit dynamic phase field damage model implementation

A. Barki, J. Zghal, L. Gallimard, I. Bruant, L. Davenne

► To cite this version:

A. Barki, J. Zghal, L. Gallimard, I. Bruant, L. Davenne. PhaFiDyn: An explicit dynamic phase field damage model implementation. *SoftwareX*, 2025, 31, pp.102227. <10.1016/j.softx.2025.102227>. <hal-05151392>

HAL Id: hal-05151392

<https://hal.science/hal-05151392v1>

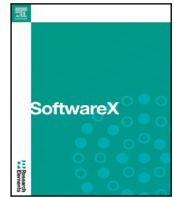
Submitted on 8 Jul 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License



Original software publication

PhaFiDyn: An explicit dynamic phase field damage model implementation

A. Barki *, J. Zghal , L. Gallimard, I. Bruant, L. Davenne

Laboratoire LEME, Université Paris Nanterre, 50 rue de Sèvres, 92410 Ville d'Avray, France

ARTICLE INFO

Keywords:

Dynamic loading
Fracture
Damage
Phase field
Crack path
FeniCS

ABSTRACT

Predicting the critical load that structures can sustain and the right crack path is an important scientific and practical issue. This reason is the trigger for the appearance of the damage mechanics. During these last decades, many models have shown their ability to predict both critical loads and crack paths. We can cite thick-level set models (TLS), phase field damage models, peridynamics, and more recently Lip-Field damage model. This work aims to show the capability of the PhaFiDyn software to predict crack paths with the unified formulation of the phase field damage model. PhaFiDyn is implemented on FeniCS, an open-source finite element library. Validation test is realized with numerical and experimental results from the literature. PhaFiDyn is easily modular, allowing for expansion of this implementation.

Code metadata

Current code version	v 1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00172
Permanent link to Reproducible Capsule	None
Legal Code License	GNU General Public License v3.0
Code versioning system used	None
Software code languages, tools, and services used	Python3, FeniCS, SymPy, NumPy, ParaView and Gmsh
Compilation requirements, operating environments and dependencies	Linux
If available Link to developer documentation/manual	https://github.com/jzghal/PhaFiDyn/blob/PhaFiDyn_main/README.md
Support email for questions	abarki@parisnanterre.fr , jzghal@parisnanterre.fr

1. Motivation and significance

Predicting the critical load that structures can withstand and determining the correct crack path are important scientific and practical challenges. A natural approach to addressing these issues is to search for an appropriate criterion. Strength and toughness criteria are well-suited for simple specimens having either no defect or an existing crack. However, when defects are present, the criterion is more complex to establish. Its origin is Griffith's criterion, which introduced the idea of two main energy hypotheses: surface energy proportional to the crack surface and crack propagation based on the principle of energy minimization. This criterion does not explain phenomena such as crack nucleation and propagation by fatigue. The ideas in Griffith's theory can only be applied to fairly simple cases. For more complex applications, other theoretical concepts have been developed, like the damage mechanics models. Over the past few decades, many models

have shown their ability to predict both critical loads and crack paths. We can cite thick level set models (TLS) [1] and phase field damage models [2].

The work presented here follows on from GPFniCS software [3] and PhaFiSta,¹ which deal with the unified formulation of the phase field damage model for quasi-static loading. This approach is extended here, in the dynamic loading cases, considering the implementation of the unified formulation of the phase field damage model. Given the complexity of setting up dynamic experimental tests, and the difficulty in understanding crack mechanisms, numerical simulations can be a valuable tool for better analysing and visualizing phenomena such as propagation, bifurcation, or branching of cracks and fragmentation. This work proposes a clear and accessible implementation, within the FeniCS environment, of the two main formulations of the phase field damage model for brittle and quasi-brittle fracture. The first is

* Corresponding author.

E-mail address: abarki@parisnanterre.fr (A. Barki).¹ <https://hal.science/hal-04832407v2>

the original problem, which involves solving a constrained variational inequality to enforce the irreversibility of damage. The second is a simplified problem [4] that solves an unconstrained variational equality, where irreversibility is approximated using a history variable that prevents the reduction of elastic potential. Moreover, the implementation of the phase field damage model for dynamic loading within the open-source FEniCS framework provides a reproducible and extensible tool for the mechanics community. Although phase field damage methods are widely studied, open-source implementations, particularly for dynamic problems, remain scarce.

An explicit time integration scheme is employed, enabling the simulation of crack propagation under dynamic loading without requiring iterative resolution at each time step. This scheme is chosen here for its advantageous properties in handling impacts and wave propagation. The implementation used in this work is based on the algorithms developed by [5]. The test cases are inspired by the works of [6].

To demonstrate the capabilities of the implementation, a numerical validation is presented. The results show good agreement with experimental data and highlight the effectiveness of the phase field damage model for dynamic fracture scenarios. Finally, the developed code is modular and well-documented, designed to support reproducibility, extensibility, and future developments in the field of damage mechanics.

In this paper, the focus is placed on the PhaFiDyn software that describes the phase field damage model implementation in the case of explicit dynamics. The outline of this work is as follows. In Section 1.1, the theory of the phase field damage model in the explicit dynamic case is presented. Its implementation is presented with an exhaustive description in Section 2. Then the validation of PhaFiDyn and an example of use are illustrated in Section 3. Finally, the main conclusions are summarized in the last section.

1.1. Phase field equations in elastodynamic

In general, the phase field approach to model systems with sharp interfaces is based on the introduction of a continuous field variable, which smoothly describes the transition between different physical phases of the system. In the context of damage mechanics, this variable represents a damage field, separating the safe phase from the fully damaged phase and thus approximating the crack discontinuity damaged zone. The evolution of this damage field, d , in response to the external loading conditions models the crack propagation process. The phase field damage model is a regularization of the variational formulation of Griffith's theory [7]. The phase field damage model relies on the definition of the following energies:

$$\text{Elastic energy : } \mathcal{E}_e(u, d) = \int_{\Omega} g(d)\psi_0(\epsilon(u)) d\Omega \quad (1)$$

$$\text{Fracture energy : } \mathcal{E}_d(d) = \int_{\Omega} \frac{G_c}{c\ell} (\alpha(d) + \ell^2 |\nabla d|^2) d\Omega \quad (2)$$

$$\text{Kinetic energy : } \mathcal{E}_c(\dot{u}) = \int_{\Omega} \frac{1}{2} \rho \dot{u}^2 d\Omega \quad (3)$$

$$\text{External energy : } \mathcal{E}_{ext}(u) = \int_{\Omega} b^* u d\Omega + \int_{\partial\Omega_t} t^* u dS \quad (4)$$

where:

- Ω represents the domain volume
- G_c is the toughness of the safe material
- ψ_0 is the elastic potential of safe material
- ρ is the mass density of material
- b^* is the distributed body force
- t^* is the distributed surface load over $\partial\Omega_t$
- u represents the displacement
- \dot{u} is the velocity associated with u
- $d \in [0, 1]$ is the damage variable
- $g(d)$ is the degradation function

ℓ is the phase field regularization length

$\alpha(d)$ is the crack geometrical function

c is the scaling parameter to recover the sharp crack surface
 $(c = 4 \int_0^1 \sqrt{\alpha(z)})$

The action integral over time interval $I = [t_i, t_{i+1}]$ is introduced, involving an augmented Lagrangian \mathcal{L} :

$$\mathcal{E}(u, d) = \int_I \mathcal{L}(u, \dot{u}, d) dt = \int_I (\mathcal{E}_e(u, d) + \mathcal{E}_d(d) - \mathcal{E}_c(\dot{u}) - \mathcal{E}_{ext}(u)) dt \quad (5)$$

At each step, the displacement and damage fields that minimize the action integral must be determined. The validity of this method is ensured by general physical principles such as the damage irreversibility, stability, and energy conservation [8–11].

According to the literature, we can distinguish two main approaches. The first one (called here the original problem) is based on the complete resolution of the phase field damage problem with updating the damage field to ensure the damage irreversibility condition. The second one (called simplified problem), proposed by [4], used a history variable defined as:

$$H(u, t) = \max_{0 \leq \tau \leq t} (\psi_0(\epsilon(u, \tau)), H_0) \quad (6)$$

where H_0 is the history energy threshold defined as a function of the chosen phase field damage model. This variable prevents the elastic potential from decreasing, which makes it possible to approximate the damage irreversibility condition of the original problem.

For these two approaches, the staggered algorithm proposed by [2] is used. It consists of an alternate minimization of the displacement and the damage problem until convergence for all loading steps. The strong formulation follows from the Gateaux derivative of the action integral in the direction of variations and integration by parts. The stress balance equation and a gradient-type evolution system for the damage field formulate this. These approaches can be summarized through the following strong equations:

- Elastodynamic problem

$$\begin{aligned} \text{div}(\boldsymbol{\sigma}) + b^* &= \rho \ddot{u} & \text{in } \Omega \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= t^* & \text{on } \partial\Omega_t \end{aligned} \quad (7)$$

- Damage problem

– Original problem

$$\begin{aligned} d &\geq 0 & \text{in } \Omega \\ g'(d)\psi_0(\boldsymbol{\epsilon}) + \frac{G_c}{c\ell} (\alpha'(d) - 2\ell^2 \Delta d) &\geq 0 & \text{in } \Omega \\ d \left[g'(d)\psi_0(\boldsymbol{\epsilon}) + \frac{G_c}{c\ell} (\alpha'(d) - 2\ell^2 \Delta d) \right] &= 0 & \text{in } \Omega \\ \nabla d \cdot \mathbf{n} &\geq 0 & \text{on } \partial\Omega \\ \dot{d} \nabla d \cdot \mathbf{n} &= 0 & \text{on } \partial\Omega \end{aligned} \quad (8)$$

– Simplified problem

$$\begin{aligned} H(u, t) &= \max_{0 \leq \tau \leq t} (\psi_0(\epsilon(u, \tau)), H_0) \\ g'(d)H(\boldsymbol{\epsilon}, t) + \frac{G_c}{c\ell} (\alpha'(d) - 2\ell^2 \Delta d) &= 0 & \text{in } \Omega \\ \nabla d \cdot \mathbf{n} &= 0 & \text{on } \partial\Omega \end{aligned} \quad (9)$$

The elastodynamic problems are usually solved by the Finite Element Method (FEM). The discrete evolution equation is:

$$\text{Find } \mathbf{u} \in v_u \text{ such as } \mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F} \quad (10)$$

where \mathbf{M} , \mathbf{K} and \mathbf{F} represent the mass, stiffness and load matrices. v_u is the admissible displacement field set.

To solve this equation, the scheme of centred differences – an explicit time integration scheme – is employed in this work. This scheme exhibits favourable properties in modelling impacts and wave propagation phenomena. This method is conditionally stable (Eq. (11)). As a result, the use of an iterative staggered algorithm is not required. The centred differences of velocity and acceleration fields are:

$$\dot{u}^t = \frac{u^{t+\Delta t} - u^{t-\Delta t}}{2\Delta t} \quad \text{and} \quad \ddot{u}^t = \frac{u^{t+\Delta t} - 2u^t + u^{t-\Delta t}}{\Delta t^2}$$

and the velocity field at half time step is:

$$\dot{u}^{t+\frac{1}{2}} = \frac{u^{t+\Delta t} - u^t}{\Delta t} \quad \text{and} \quad \dot{u}^{t-\frac{1}{2}} = \frac{u^t - u^{t-\Delta t}}{\Delta t}$$

The resolution steps of elastodynamic problem are [12]:

1. $\dot{u}^{k+\frac{1}{2}} = \dot{u}_k + \frac{\Delta t}{2} \ddot{u}_k$
2. $u^{k+1} = u^k + \Delta t \dot{u}^{k+\frac{1}{2}}$ and apply the Dirichlet boundary conditions
3. $\ddot{u}^{k+1} = \mathbf{M}^{-1}[\mathbf{F} - \mathbf{K}u^{k+1}]$
4. $\dot{u}^{k+1} = \dot{u}^{k+\frac{1}{2}} + \frac{\Delta t}{2} \ddot{u}^{k+1}$ and apply the Dirichlet boundary conditions

After the update, the Dirichlet boundary conditions are applied by directly prescribing the displacement values on the constrained degrees of freedom.

When using the explicit time integration scheme, the Courant–Friedrichs–Lewy (CFL) condition must be satisfied [12]:

$$\Delta t < \frac{h_{\min}}{\eta} \sqrt{\frac{\rho}{E}} \quad (11)$$

where h_{\min} represents the minimum distance between nodes within the discretized domain, η is a safety factor larger than 1, E is the Young modulus, and ρ is the mass density.

Therefore, the algorithm of the elastodynamic phase field minimization using the explicit time integration scheme can be described as:

Algorithm 1 Explicit dynamics phase field damage model

Given initial mechanical conditions: u^0, \dot{u}^0 and initial damage condition d^*

if Original problem then

$$d^0 = \operatorname{argmin}_{d, d^* \leq d} (\mathcal{E}(u^0, d))$$

else if Simplified problem then

$$H^0 = \max(\psi_0(u^0), H_0)$$

$$d^0 = \operatorname{argmin}_{d, H^0} (\mathcal{E}(u^0, d))$$

end if

Initialize $\ddot{u}^0 = \mathbf{M}^{-1}[\mathbf{F} - \mathbf{K}u^0]$

for each time step k do

$$u^{k+1} = u^k + \Delta t \dot{u}_k + \frac{\Delta t^2}{2} \ddot{u}_k$$

Apply the Dirichlet boundary conditions

if Original problem then

Update minimal bound

$$d_{\min} \leftarrow d^k$$

$$d^{k+1} = \operatorname{argmin}_{d, d^k \leq d} (\mathcal{E}(u^{k+1}, d))$$

else if Simplified problem then

Update history variable

$$H^{k+1} = \max(\psi_0(u^{k+1}), H^k, H_0)$$

$$d^{k+1} = \operatorname{argmin}_{d, H^{k+1}} (\mathcal{E}(u^{k+1}, d))$$

end if

$$\ddot{u}^{k+1} = \mathbf{M}^{-1}[\mathbf{F} - \mathbf{K}u^{k+1}]$$

$$\dot{u}^{k+1} = \dot{u}^k + \frac{\Delta t}{2} (\ddot{u}^k + \ddot{u}^{k+1})$$

Apply the Dirichlet boundary conditions

end for

1.2. Unified formulation of phase field damage model

Several models have been defined by the mechanical community. The two most used models in elastodynamic applications are the PF-AT2 model [13,14] and the PF-AT1 model [8]. These two models allow

Table 1

Parameters for the phase field models.

Model	Softening type	c	α	a_1	a_2	a_3	p
PF-AT2		2	d^2	2	-0.5	0	2
PF-AT1		8/3	d	2	-0.5	0	2
PF-CZM	Linear	π	$2d - d^2$	-	-0.5	0	2

Table 2

The threshold H_0 expressions.

	PF-AT2	PF-AT1	PF-CZM
H_0	0	$\frac{3G_c}{16\ell}$	$\frac{\sigma_c^2}{2E}$

for to description of the brittle fracture. Then, the first cohesive zone phase field models (PF-CZM) appear [15] and are developed by [16]. In the cohesive zone model, the failure strength is considered, trying to represent the material strength. This addition improves the phase field damage model. The cohesive models have demonstrated their ability to describe the cracking process in brittle and quasi-brittle materials. The crack process is linked to stresses and depends on the damage model chosen. Although many other models exist, the PF-AT1, PF-AT2 and PF-CZM models are focused here. The different brittle fracture models are characterized mainly by the type of geometric cracking function and degradation (or softening) function used. The unified formulation of the phase field damage model is defined through:

- the unified degradation function $g(d)$, defined as [16]:

$$g(d) = \frac{(1-d)^p}{(1-d)^p + a_1 d \cdot P(d)} \quad (12)$$

where

$$P(d) = 1 + a_2 d + a_3 d^2, \quad p \geq 2, \quad a_1 d \cdot P(d) > 0$$

$$\text{and } a_1 = \frac{4G_c E}{\pi \ell \sigma_c^2} > 0, \quad a_2 \text{ and } a_3 \text{ are constants.}$$

- the crack geometrical function $\alpha(d)$ and the scaling parameter.

The different parameters of the unified formulation of phase field models can be summarized in the following Table 1. Moreover, that for each phase field model, the threshold H_0 is defined in Table 2.

2. Software description

PhaFiDyn software models brittle isotropic elastic materials submitted to dynamic loading in two dimensions. The dynamics are represented by time discretization and spatial discretization using the finite element method. For that, PhaFiDyn is implemented on the FeniCS framework. FeniCS is an open-source library that is efficient, flexible, and designed to solve partial differential equations using finite element methods [17]. As the phase field damage model is based on a variational formulation, the use of FeniCS is perfectly adapted. Moreover, it facilitates the meshes management and the solving of linear or non-linear problems. The isotropic elasticity is considered for this initial implementation. However, PhaFiDyn can be extended to materials more complex, such as electromechanical or thermomechanical couplings. It is also possible to consider the anisotropic behaviour, both for rigidity and for damage. In short, the use of FeniCS for the phase field damage model problem gives flexibility and the possibility to evolve this implementation towards more complex materials.

PhaFiDyn uses mainly the FeniCS 2019.1.0 library, which is the latest stable release version. To download this version, follow the installation procedures for your system at <https://fenicsproject.org/download/archive/>. PhaFiDyn is implemented in the Python 3 environment. Moreover, the downloads of SymPy and NumPy libraries are also necessary. ParaView is used to display the mesh and the results. The

loaded format file is .xdmf (and .h5). To download ParaView, follow the installation procedures at <https://www.paraview.org/download/>. The mesh is loaded via Gmsh and uses the msh2 format. PhaFiDyn can support Tri3 finite elements, as FeniCS only handles this type of finite element.

2.1. Software architecture

The problem resolution of the phase field damage model consists of the research of admissible fields that minimize the action integral. Consequently, the phase field damage model problem is governed by the strong formulations ((7), (9), (8)). Two approaches for the phase field method have been presented in the previous section. Moreover, the consideration of unilateral contact is necessary to have the crack coherent, especially when we have different fracture modes. For that, the elastic potential decomposition in the positive part and the negative part is added. Two methods are used here: the volumetric/deviatoric decomposition from [18] and spectral decomposition [4].

The implementation strategy is presented in Fig. 1. To begin, the geometry and the material properties are defined, and the phase field damage model is chosen. The mesh is converted into .xml format. Then, the boundary conditions are defined using the FeniCS functions and the mesh. In FeniCS, several finite element spaces are available. In this tool, the used element's spaces are Continuous Galerkin type. These spaces are defined in considering the vectorial shape of displacement, velocity and acceleration fields as well as the scalar shape of the damage field. An additional space of Discontinuous Galerkin is added for the simplified problem to define the history variable at the previous step. The trial and test fields are expressed using these spaces by the two FeniCS functions: TrialFunction() and TestFunction(). The fields u , v , a represent the displacement, velocity, and acceleration fields at the previous step, while u_{new} , v_{new} , a_{new} correspond to these fields at the new step. The damage fields are also defined, with $pold$ to the previous step and phi to the new step. For the simplified problem, the field representing the history variable at the previous step, $Hold$, is given. Then, according to the considered model (PF-AT2, PF-AT1, PF-CZM), the weak formulation of this problem is expressed, either by deriving the action integral with the FeniCS function, derivative(), or by defining it manually. For the original problem, the constrained minimization is realized with the initialization bounds for the damage field: $d_{min}=0$ and $d_{max}=1$. In the case of the simplified problem, as there is a history variable, the minimization is unconstrained. The behaviour law, the consideration (or not) of unilateral contact, the parameters defining the phase field damage model are described in several functions (Psi0(u,d), sigma_0(u), sigma(u,d), PFparameters(E, nu, Gc, sigc, Model), ...). Once the weak formulation is established, the steps of algorithm 1 are followed. The fields phi and a are initialized. Before starting the phase field damage problem resolution, the mass matrix is computed. The solvers are chosen according to the selected approach. For the original problem, which implies solving an inequality variational, a solver from the library SNES, such as `vinewtonssls` or `vinewtonrsls` is used. In the case of the simplified problem, an equality variational problem is solved. For the models PF-AT2 and PF-AT1, a linear solver is sufficient. While for the PF-CZM model, a non-linear solver is necessary, `newtonls` or `newtontr` is used from the library SNES [19–21].

Then, the incremental loop is launched, considering the mechanical loading and the time discretization. At each increment, if the boundary conditions depend on time, they are updated. u_{new} is computed, and the associated boundary conditions are applied to this field. Then phi is determined with u_{new} , and a_{new} is computed from u_{new} and phi . Finally, v_{new} is obtained with a_{new} . At the end of the incremental step, the fields from the previous step are updated with the new fields. To ensure the damage irreversibility with the original problem, d_{min} is updated with phi . For the simplified problem,

Table 3
Physical values of results in data.

Increment	Time	ElasticEnergy	KineticEnergy	FractureEnergy
ExternalEnergy	TotalEnergy	dmin	dmax	x_dmax
	y_dmax			

$Hold$ is updated with the history variable computed using the field u_{new} .

Then the datas are saved with `postprocessing()` and `paraviewResults()` functions. `postprocessing()` function allows the creation of a results file in .data format. The results are reported in a table with the following header (see Table 3).

Simulation results are exported to ParaView format. The `ParaviewResults()` function exports several fields, including displacement, velocity, acceleration, damage, stress, and strain, to a ParaView file for visualization and analysis. In the end, a cartography of these fields is obtained.

To solve the phase field damage model, the minimization algorithm 1 was implemented in FeniCS 2019 following the described strategy Fig. 1.

2.2. Software functionalities

PhaFiDyn gives a large flexibility and allows various modelling. The material properties and the file geometry in .geo format are the inputs. The mesh can be realized with specific meshing tools and considered as an input file in .msh format. Otherwise, the software can make the mesh in the same format. It also allows us to specify important parameters, such as the tolerance and iterations number of solvers, the choice between the plane strain and the plane stress hypothesis, the type of decomposition, and the phase field damage model. Moreover, The `Nout` variable can be used to control the output frequency, so that ParaView files are saved every `Nout` time steps instead of every step. The boundary conditions and the crack tip tracking can be adjusted as a function of the studied problem. Finally, to execute the software, it could launch the following command in your terminal: `python filename.py` and the computation is launched. The results file in .data format can be used to plot the physical values wanted.

3. Illustrative examples

To illustrate the PhaFiDyn performances, two numerical simulations are conducted. Both tests are common in the dynamic phase field damage models. The first one is a validation test, called the Kalthoff experiment and conducted by Kalthoff and Winkler [22]. The second simulation involves dynamic branching, which is documented in the literature [6]. For these numerical simulations, the original and simplified problems are considered.

Reproducibility and implementation details

To facilitate reproducibility, all simulations were performed using the open-source finite element framework FeniCS 2019.1.0, in a Python 3 environment. The nonlinear solvers are based on the PETSc SNES library. The meshes were generated with Gmsh and the results were post-processed via ParaView, using the formats .xdmf and .h5.

All the code, including the geometry files, solver settings and execution instructions, is publicly available at the following address : <https://github.com/jzghal/PhaFiDyn>

The implementation was validated by comparison with :

- the reference numerical results of Borden et al. [6];
- the experimental data from Kalthoff and Winkler [22].

These comparisons confirm the consistency and reliability of the numerical approach developed in this work.

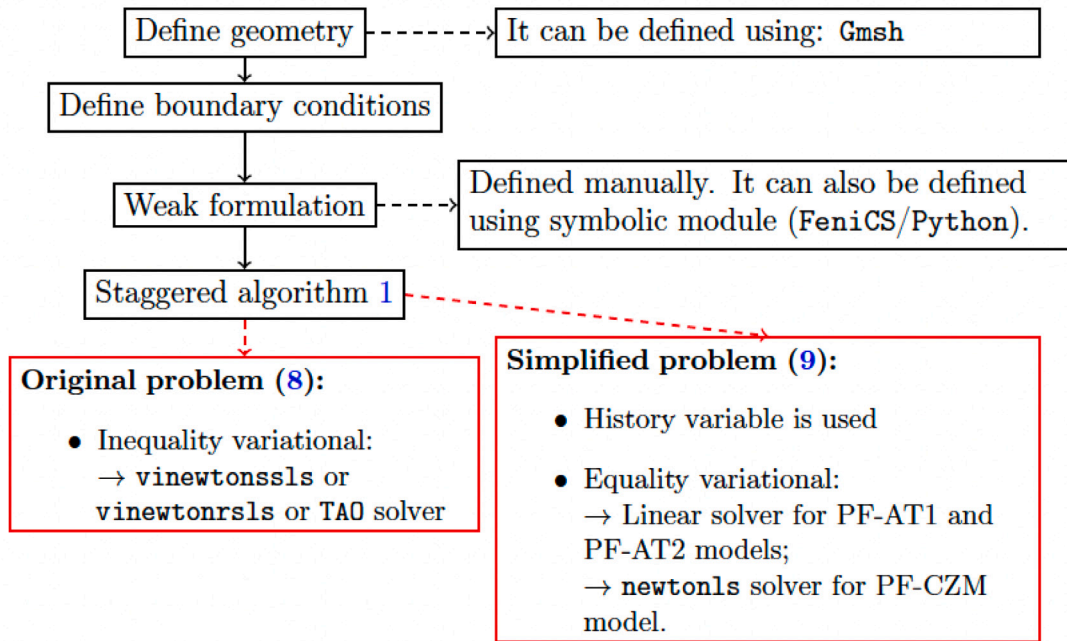


Fig. 1. Implementation strategy on FeniCS 2019 with PETsc library [19–21].

Table 4
Material properties from [6].

Mass density ρ [kg/m ³]	8000
Young's modulus E [Pa]	190×10^9
Poisson's ratio ν	0.3
Toughness G_c [Pa.m]	2.213×10^4

Table 5
Mechanical properties from [6].

Mass density ρ [kg/m ³]	2450
Young's Modulus E [Pa]	32×10^9
Poisson's Ratio ν	0.3
Toughness G_c [Pa.m]	3
Failure Strength σ_c [Pa]	6.4×10^6

3.1. Kalthoff test

A double-notched rectangular plate, subjected to a dynamic shear, is simulated. PF-AT2 and PF-AT1 models are used. The results from the PF-AT2 model are compared with the numerical results from [6]. The geometry and boundary are illustrated in Fig. 2. A symmetric condition is considered to reduce the time of computation. The impact is simulated by the application of velocity represented Fig. 3, with a maximum velocity of 16.5 m/s. The material properties are defined in the Table 4. The hypothesis of plane strain is considered. The length regularization is $\ell = 1.95 \times 10^{-4}$ m. The structure is discretized with a non-uniform mesh refined in the crack zone. The mesh size in the crack zone is $h = 9.7 \times 10^{-5}$. The finite elements total number is 240 532. The time step is $\Delta t = 2.65 \times 10^{-9}$ s to respect the CFL condition (11). The spectral decomposition is used.

In Fig. 4, the elastic and fracture energies are compared with the numerical results from [6] to validate our implementation. The results are quite similar. The slight difference can be explained by the integration scheme, as the Generalized- α method is used in [6].

Then, in Fig. 5, the crack, the crack length as a function of time and the velocity crack normalized by the Rayleigh velocity as a function of time are plotted. Tracking the crack tip enabled us to plot these curves. The numerical results from the original and simplified problems are almost identical. The angle of crack initiation is 70.8°. The crack velocity is below 60% of the Rayleigh velocity.

The results for the PF-AT1 model are represented in Fig. 6. As for the PF-AT2 model, the original and simplified problems have almost identical results. The angle of crack initiation is 71.1°. The crack velocity is slightly above 60% of the Rayleigh velocity.

From the experimental results in [22], the angle of the crack initiation is 70°. The numerical results, from the PF-AT2 and PF-AT1 models, are therefore consistent with the experimental results published in [22].

3.2. Dynamic branching

To illustrate the performance of PhaFiDyn, a dynamic branching test is considered here.

The tensile loading on a notched plate, from [6], is considered. The linear PF-CZM model is used. The geometry and boundary conditions are illustrated in Fig. 7. A tensile load is applied to the top and bottom surfaces, as shown in Fig. 8. The material properties are defined in Table 5. The plane strain hypothesis is considered. The length regularization is $\ell = 5 \times 10^{-4}$ m. A uniform mesh is used, with a mesh size of $h = 1.25 \times 10^{-4}$ m, resulting in 441 490 finite elements. The time step is set to $\Delta t = 4.9 \times 10^{-9}$ s to satisfy the CFL condition. The volumetric/deviatoric decomposition is used.

In this dynamic branching case, there are two cracks. Both cracks are distinctly named: the upper crack and the lower crack. In Fig. 9, both the cracks and their velocities are illustrated. After the branching, the crack propagation is almost identical between the upper and lower cracks.

In agreement with the observations of [6], the numerical results show a dynamic branching. The analysis of reflected waves explains this mechanism. Using PhaFiDyn, we successfully replicated this phenomenon, validating the relevance of our numerical approach.

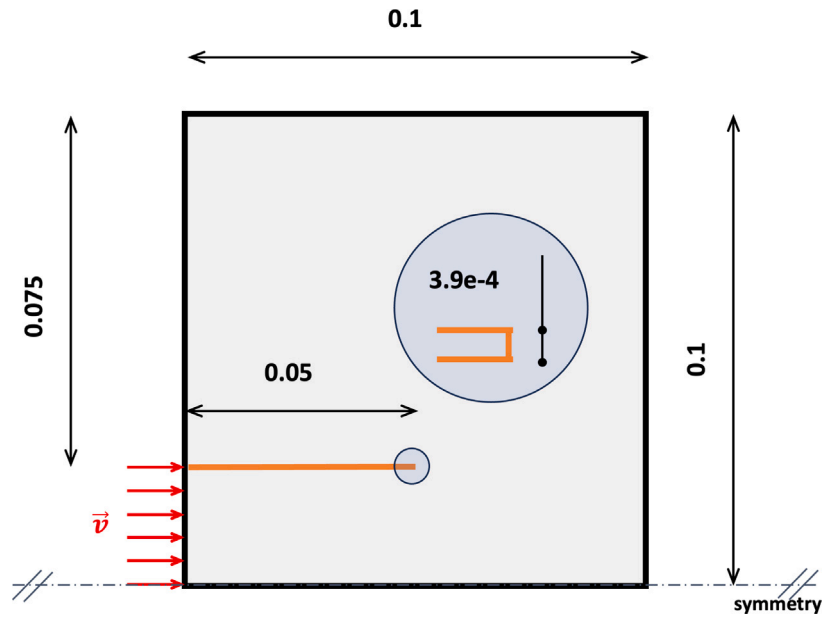


Fig. 2. Geometry definition (dimensions are in meter).

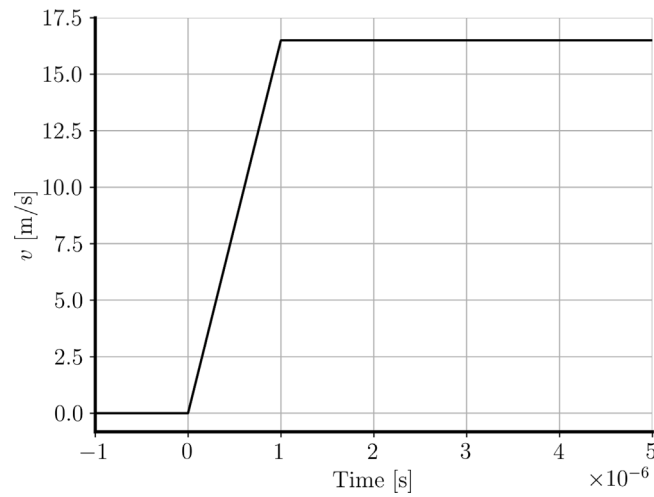


Fig. 3. Kalthoff test: applied velocity.

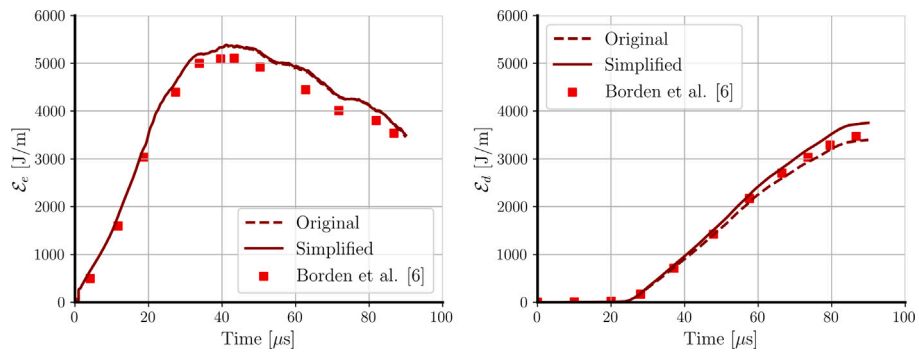


Fig. 4. Kalthoff test: PF-AT2 model - ϵ_e and ϵ_d vs Time.

4. Conclusions

PhaFiDyn is an additional tool for PhaFiSta. PhaFiDyn offers public access to dynamic fracture analysis using the PF-AT2, PF-AT1, and PF-CZM models in two dimensions. These different models allow to model

the brittle and quasi-brittle fracture. Both the most classical decompositions are also implemented: the volumetric/deviatoric decomposition from [18] and the spectral decomposition [4]. This software has been validated with the numerical results from [6] and experimental results from [22]. Now, PhaFiDyn is easily modular to consider :

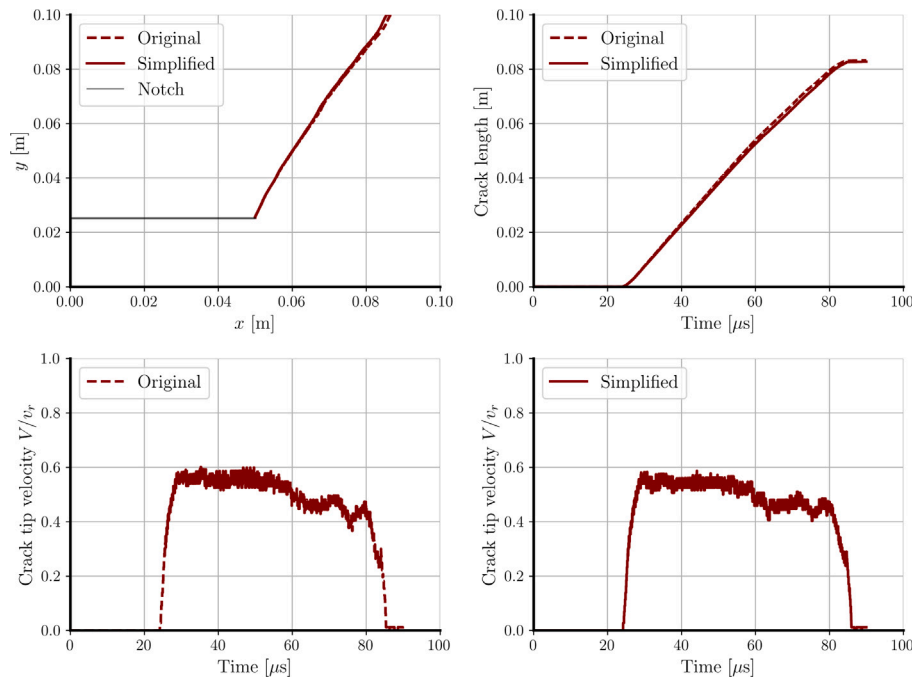


Fig. 5. Kalthoff test: PF-AT2 model results.

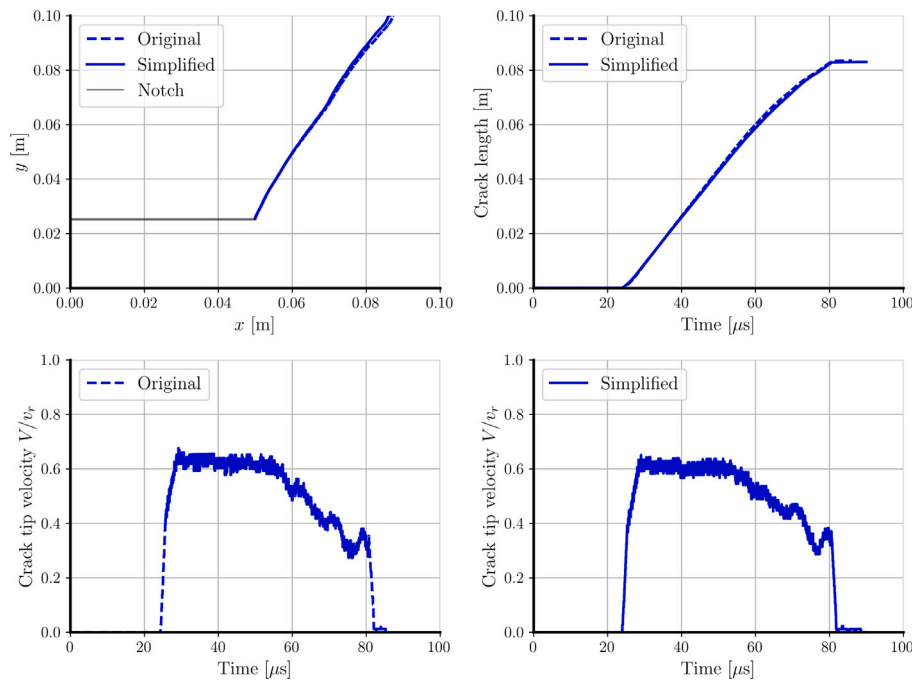


Fig. 6. Kalthoff test: PF-AT1 model.

- other behaviour laws (for example, thermo-mechanical or electro-mechanical couplings),
- other decompositions of the elastic potential,
- different phase field cohesive zone models,
- anisotropic behaviours for rigidity and damage.

Finally, it can be generalized to three-dimensional simulations.

CRedit authorship contribution statement

A. Barki: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization. **J. Zghal:** Writing

- review & editing, Writing – original draft, Supervision, Software, Methodology, Investigation, Conceptualization. **L. Gallimard:** Writing – review & editing, Conceptualization. **I. Bruant:** Writing – review & editing, Conceptualization. **L. Davenne:** Writing – review & editing, Conceptualization.

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.

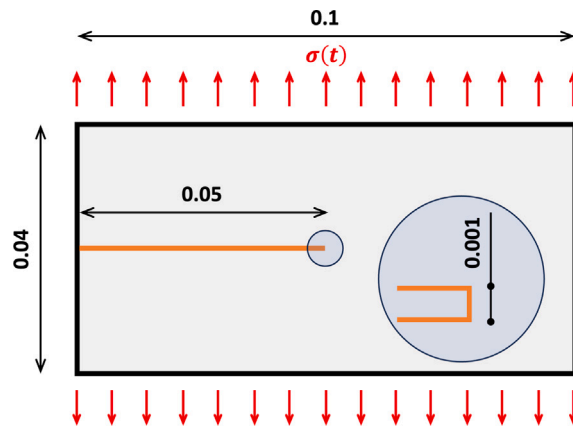


Fig. 7. Geometry definition (dimensions are in meter).

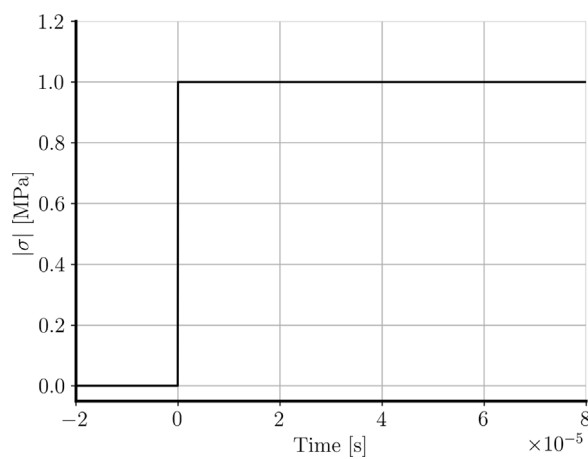


Fig. 8. Dynamic branching: applied load.

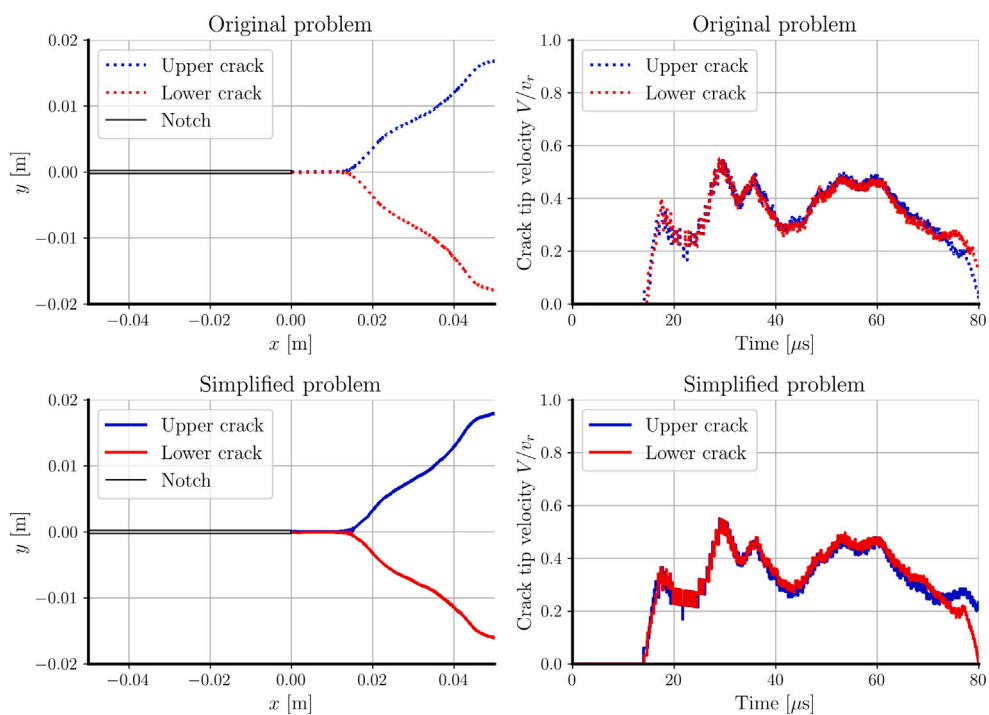


Fig. 9. PF-CZM linear model.

References

- [1] Moës N, Stolz C, Bernard P-E, Chevaugeon N. A level set based model for damage growth: The thick level set approach. *Internat J Numer Methods Engrg* 2011;86(3):358–80. <http://dx.doi.org/10.1002/nme.3069>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3069>.
- [2] Bourdin Blaise, Francfort Gilles A, Marigo Jean-Jacques. The variational approach to fracture. *J Elasticity* 2008/04/01;91(1):5–148. <http://dx.doi.org/10.1007/s10659-007-9107-3>.
- [3] Kumar Manish, Alessi Roberto, Salvati Enrico. GPFniCS: A generalised phase field method to model fracture. *SoftwareX* 2023;24:101594. <http://dx.doi.org/10.1016/j.softx.2023.101594>, URL <https://www.sciencedirect.com/science/article/pii/S235271102300290X>.
- [4] Miehe Christian, Hofacker Martina, Welschinger Fabian. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Comput Methods Appl Mech Engrg* 2010;199(45):2765–78. <http://dx.doi.org/10.1016/j.cma.2010.04.011>, URL <https://www.sciencedirect.com/science/article/pii/S0045782510001283>.
- [5] Li Tianyi. Analyse de la rupture dynamique fragile via les modèles d'endommagement à gradient : principes variationnels et simulations numériques (Ph.D. thesis), (2016SACLX042). Université Paris Saclay (COmUE); 2016, URL <https://pastel.hal.science/tel-01487449>.
- [6] Borden Michael J, Verhoosel Clemens V, Scott Michael A, Hughes Thomas JR, Landis Chad M. A phase-field description of dynamic brittle fracture. *Comput Methods Appl Mech Engrg* 2012;217–220:77–95. <http://dx.doi.org/10.1016/j.cma.2012.01.008>, URL <https://www.sciencedirect.com/science/article/pii/S0045782512000199>.
- [7] Francfort GA, Marigo J-J. Revisiting brittle fracture as an energy minimization problem. *J Mech Phys Solids* 1998;46(8):1319–42. [http://dx.doi.org/10.1016/S0022-5096\(98\)00034-9](http://dx.doi.org/10.1016/S0022-5096(98)00034-9), URL <https://www.sciencedirect.com/science/article/pii/S0022509698000349>.
- [8] Pham Kim, Amor Hanen, Marigo Jean-Jacques, Maurini Corrado. Gradient damage models and their use to approximate brittle fracture. *Int J Damage Mech* 2011;20(4):618–52. <http://dx.doi.org/10.1177/1056789510386852>.
- [9] Pham Kim, Marigo Jean-Jacques. Approche variationnelle de l'endommagement : II. Les modèles à gradient. *Comptes Rendus Mécanique* 2010;338(4):199–206. <http://dx.doi.org/10.1016/j.crme.2010.03.012>, URL <https://www.sciencedirect.com/science/article/pii/S1631072110000446>.
- [10] Pham Kim, Marigo Jean-Jacques. Approche variationnelle de l'endommagement : I. Les concepts fondamentaux. *Comptes Rendus Mécanique* 2010;338(4):191–8. <http://dx.doi.org/10.1016/j.crme.2010.03.009>, URL <https://www.sciencedirect.com/science/article/pii/S1631072110000410>.
- [11] Marigo Jean-Jacques, Maurini Corrado, Pham Kim. An overview of the modelling of fracture by gradient damage models. *Meccanica* 2016/12/01;51(12):3107–28. <http://dx.doi.org/10.1007/s11012-016-0538-4>.
- [12] Belytschko T, Liu WK, Moran B, Elkhodary K. *Nonlinear Finite Elements for Continua and Structures*. Wiley; 2014, URL <https://books.google.fr/books?id=BQpfAQAAQBAJ>.
- [13] Ambrosio Luigi, Tortorelli Vincenzo Maria. Approximation of functional depending on jumps by elliptic functional via Γ -convergence. *Comm Pure Appl Math* 1990;43(8):999–1036. <http://dx.doi.org/10.1002/cpa.3160430805>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160430805>.
- [14] Ambrosio Luigi, Tortorelli VM. On the approximation of free discontinuity problems. *Boll. Unione Mat. Ital., VII. Ser B* 1992;6(1):105–23.
- [15] Lorentz Eric, Cuvilliez S, Kazymyrenko K. Convergence of a gradient damage model toward a cohesive zone model. *Comptes Rendus Mécanique* 2011;339(1):20–6. <http://dx.doi.org/10.1016/j.crme.2010.10.010>, URL <https://www.sciencedirect.com/science/article/pii/S1631072110001671>.
- [16] Wu Jian-Ying. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. *J Mech Phys Solids* 2017;103:72–99. <http://dx.doi.org/10.1016/j.jmps.2017.03.015>, URL <https://www.sciencedirect.com/science/article/pii/S0022509616308341>.
- [17] Alnæs Martin, Blechta Jan, Hake Johan, Johansson August, Kehlet Benjamin, Logg Anders, et al. The FEniCS project version 1.5. *Arch Numer Softw* 2015;3(100). <http://dx.doi.org/10.11588/ans.2015.100.20553>.
- [18] Amor Hanen, Marigo Jean-Jacques, Maurini Corrado. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. *J Mech Phys Solids* 2009;57(8):1209–29. <http://dx.doi.org/10.1016/j.jmps.2009.04.011>, URL <https://www.sciencedirect.com/science/article/pii/S0022509609000659>.
- [19] Balay Satish, Abhyankar Shirang, Adams Mark F, Benson Steven, Brown Jed, Brune Peter, et al. PETSc web page. 2025, URL <https://petsc.org/>.
- [20] Balay Satish, Abhyankar Shirang, Adams Mark F, Benson Steven, Brown Jed, Brune Peter, et al. PETSc/TAO Users Manual. Technical Repor ANL-21/39 - Revision 3.23, Argonne National Laboratory; 2025, <http://dx.doi.org/10.2172/2476320>.
- [21] Balay Satish, Gropp William D, McInnes Lois Curfman, Smith Barry F. Efficient management of parallelism in object oriented numerical software libraries. In: Arge E, Bruaset AM, Langtangen HP, editors. *Modern software tools in scientific computing*. Birkhäuser Press; 1997, p. 163–202.
- [22] Kalthoff Joerg F. Modes of dynamic shear failure in solids. *Int J Fract* 2000/01/01;101(1):1–31. <http://dx.doi.org/10.1023/A:1007647800529>.