



Article

Mesh-Based and Meshfree Reduced Order Phase-Field Models for Brittle Fracture: One Dimensional Problems

Ngoc-Hien Nguyen ^{1,2,*}, Vinh Phu Nguyen ^{3,*} , Jian-Ying Wu ⁴ and Thi-Hong-Hieu Le ⁵ and Yan Ding ² 

¹ Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

² Department of Mathematical Sciences, School of Science, RMIT University, Melbourne, Victoria 3000, Australia; yan.ding@rmit.edu.au

³ Department of Civil Engineering, Monash University, Clayton, Victoria 3800, Australia

⁴ State Key Laboratory of Subtropical Building Science, South China University of Technology, Guangzhou 510641, China; jywu@scut.edu.cn

⁵ Department of Aerospace Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh City 700000, Vietnam; honghieu.le@hcmut.edu.vn

* Correspondence: hien.nguyen@rmit.edu.au (N.-H.N.); phu.nguyen@monash.edu (V.P.N.); Tel.: +61-3-8805-6987 (N.-H.N.); +61-3-9905-5989 (V.P.N.)

Received: 17 May 2019; Accepted: 5 June 2019; Published: 8 June 2019



Abstract: Modelling brittle fracture by a phase-field fracture formulation has now been widely accepted. However, the full-order phase-field fracture model implemented using finite elements results in a nonlinear coupled system for which simulations are very computationally demanding, particularly for parametrized problems when the randomness and uncertainty of material properties are considered. To tackle this issue, we present two reduced-order phase-field models for parametrized brittle fracture problems in this work. The first one is a mesh-based Proper Orthogonal Decomposition (POD) method. Both the Discrete Empirical Interpolation Method (DEIM) and the Matrix Discrete Empirical Interpolation Method ((M)DEIM) are adopted to approximate the nonlinear vectors and matrices. The second one is a meshfree Kriging model. For one-dimensional problems, served as *proof-of-concept demonstrations*, in which Young's modulus and the fracture energy vary, the POD-based model can speed up the online computations eight-times, and for the Kriging model, the speed-up factor is 1100, albeit with a slightly lower accuracy. Another merit of the Kriging's model is its non-intrusive nature, as one does not need to modify the full-order model code.

Keywords: phase-field theory; brittle fracture; Reduced-Order Model (ROM); Kriging model; Proper Orthogonal Decomposition (POD)

1. Introduction

Fracture is one of the most commonly-encountered failure modes of engineering materials and structures. Prevention of cracking-induced failure is, therefore, a major constraint in engineering designs. As with many other physical phenomena, computational modelling of fracture constitutes an indispensable tool not only to predict the failure of cracked structures, for which full-scale experiments are either too costly or even impracticable, but also to shed light onto understanding the fracture processes of many materials such as concrete, rock, ceramics, metals, biological soft tissues, etc. Within the context of continuum modelling of brittle fracture, this paper presents mesh-based and mesh-free reduced-order phase-field models. The proposed models can be used for parameter sensitivity analysis for brittle fracture problems, parameter estimation for phase-field brittle fracture

models, and fracture constrained optimization problems; that is, all sorts of problems involving the repeated solution of differential equations that govern the phenomenon of brittle fracture [1].

Brittle fracture is herein modelled using a Phase-Field Model (PFM) [2–4], which is able to handle crack initiation, propagation, merging, and branching in two and three dimensions with a relatively simple implementation. The basic idea is to approximate a sharp crack by a diffuse band of finite width (characterized by a regularized length scale parameter b) via the introduction of a scalar damage-like phase-field. The crack patterns are the natural outcome of a system of two coupled partial differential equations obtained from the minimization of a potential energy that consists of a stored bulk energy, the work of external forces, and the surface energy. Furthermore, PFMs combine features of fracture mechanics (when the length scale parameter b approaches zero, phase-field solutions recover fracture mechanics predictions) and damage mechanics into one single theory, thus overcoming the aforementioned difficulties of many other approaches [4,5]. They have been applied to brittle fracture [3,4,6,7], quasi-brittle fracture [8–11], dynamic fracture [12–16], and multi-physics fracture [17–20]. We refer to the reviews of PFMs in [21–23] for an intensive list of references.

However, it is widely recognized that PFM simulations are very computationally demanding as fine meshes are required to resolve the highly-localized deformations within the damage band. Things can be even worse because of the “curse of dimensionality” when the system is parametrized and a higher parameter space (i.e., more varying parameters) is considered. The Reduced-Order Model (ROM) is proposed and applied in such situations. The ROM aims to reduce the dimension of the state-space system and hence to decrease computational expense, while retaining the characteristic dynamics of the original system and preserving the input-output relationship.

The general framework for model reduction, either parametric or non-parametric, is based on a projection technique. More specifically, a reduced-order model is obtained by projecting a large-scale system onto a low-dimensional subspace for which the basic vectors can be derived from the method of snapshots and Proper Orthogonal Decomposition (POD) [24,25]. For linear and non-parametric problems, the reduced-order matrices and vectors are first pre-computed, pre-projected, and remain constant, and the ROM can thus be efficiently evaluated without further reference to the original Full-Order Model (FOM) [26]. The situation becomes more complicated for nonlinear, parametric, and/or coupled systems for which the reduced-order matrices and vectors are parameter-dependent. The need for re-computing and re-projecting full-order matrices for each new parameter/newest solutions turns out to be more expensive than solving the FOM itself.

In order to address the above challenges, several approaches have been proposed, e.g., global-POD [24], POD-Greedy [27], missing point estimation [28], Gauss–Newton with approximated tensors [29], Empirical Interpolation Method (EIM) [30] and its discrete variants, the Discrete Empirical Interpolation Method (DEIM) [31,32], the Matrix Discrete Empirical Interpolation Method ((M)DEIM) [33–35], and matrix interpolation methods [36–40]. Thanks to these interpolation algorithms, especially the EIM/DEIM/(M)DEIM, model reduction for nonlinear problems has achieved significant progress in recent years.

Even though model order reduction has been applied in many fields such as fluid mechanics [24,41,42] and structural dynamics [43–45], there are only a few in fracture mechanics; see [46] for fatigue crack problems, [47] for softening viscoplasticity, [48,49] for the lattice model-based nonlinear fracture problems, and [50,51] for multiscale fracture problems. The aim of this paper is to present *reduced-order phase-field models for brittle fracture*, which can be used to solve parametrized brittle fracture problems efficiently, for example parameter sensitivity analysis for brittle fracture problems, parameter estimation for phase-field brittle fracture models, fracture constrained optimization problems [52], and when the randomness and uncertainty of material properties are considered. To this end, we selected the PFM of [2,3] and constructed the corresponding reduced-order models using POD-(M)DEIM and the Kriging method [53,54]. For large-scale systems, e.g., $n = 10^5$ mesh points, the offline POD-(M)DEIM phase cannot be performed since it is impossible to store 10^{10} -dimensional

vectors. The Kriging model, as a promising alternative approach, is a meshfree surrogate model that adopts statistical methods, which can provide deeper insight into the relationship between some outputs of interest and input design variables. Compared with POD-(M)DEIM, the advantage of the Kriging model is its fast online computations and lower computer storage.

In order to avoid all the complexities of PFMs so as to focus on the ROM itself, we have selected a one-dimensional (1D) PFM for quasi-static small strain brittle fracture. In this simple setting, one does not have to deal with strain energy splits, damage boundedness, irreversibility, and so on. We emphasize that the aims of this paper are not to solve the 1D parametrized brittle fracture problem by using a ROM, the solution of which can be found analytically [55], but to demonstrate how to build ROMs for PFMs and present preliminary 1D results. The proposed ROMs are inherently multi-dimensional. The fact that we have applied them to a very simple problem of a softening bar is due to the lack of computing resources to perform the intensive offline calculations (one might need to carry out 5000 2D/3D fracture simulations). However, we are not clear if ROMs can perform well for two- and three-dimensional PFMs.

Based on computations of a one-dimensional bar with varying the Young's modulus and fracture energy (thus, geometry, loadings, and boundary conditions are fixed even though they can also be parameters), the POD-(M)DEIM ROM can speed up the online computations eight-times, whereas for the Kriging model, the speed up factor is 1100, albeit with a slightly lower accuracy. Moreover, the Kriging model has the extra advantage of its non-intrusive nature in the sense that one does not need to modify the full-order model code. Needless to say, all these savings in the computational cost are achieved with extensive offline computations using the full model. These encouraging one-dimensional results are simply a proof-of-concept demonstration and serve as a platform to build ROMs for three-dimensional brittle fracture problems.

The remainder of this paper is structured as follows. Section 2 presents the formulation of the selected phase-field brittle fracture model, which includes the governing equations, the weak form, and the finite element solver. This is followed by Section 3, which is devoted to the presentation of the two reduced-order models: the POD-(M)DEIM ROM in Section 3.1 and the Kriging model in Section 3.2. Numerical examples are provided in Section 4 to assess the performance of these models. Conclusions and further works required to lift the limitations of the current work are given in Section 5. The POD algorithm is presented in Appendix A, and the analytical solution of the investigated model is given in Appendix B.

2. Phase-Field Model for Quasi-Brittle Fracture

This section briefly recalls the one-dimensional phase-field model for brittle fracture. Governing equations are given in Section 2.1, and the weak form and finite element discretization are presented in Section 2.2. We refer to [3,6,23] for details on various phase-field models for brittle and cohesive fracture.

2.1. Governing Equations

Let us consider a bar of length L , which is fixed at the left end ($x = 0$) and pulled at the right end ($x = L$). Without loss of generality, a unit cross-sectional area is assumed. The displacement and crack phase-field (or damage field) are represented by the functions $u := u(x, d; E_0, G_c, b, u^*)$ and $d := d(x, u; E_0, G_c, b)$, respectively. The following governing equations and boundary conditions hold [23]:

$$\frac{\partial \sigma}{\partial x} = 0, \quad (\text{equilibrium equation}) \quad (1)$$

$$\left(\frac{G_c}{b} + E_0 \epsilon^2\right) d - G_c b \frac{\partial^2 d}{\partial x^2} = E_0 \epsilon^2, \quad (\text{damage evolution equation}) \quad (2)$$

$$\sigma = (1 - d)^2 E_0 \epsilon, \quad (\text{stress-strain equation}) \quad (3)$$

$$\epsilon = \frac{\partial u}{\partial x}, \quad (\text{kinematics equation}) \quad (4)$$

$$u(x=0) = 0, \quad u(x=L) = u^*, \quad (\text{essential boundary condition for } u) \quad (5)$$

$$d(x=0, L) = 0, \quad (\text{essential boundary condition for } d) \quad (6)$$

where $\epsilon(x)$ and $\sigma(x)$ are the strain and stress fields, respectively; E_0 and G_c denote Young's modulus and the fracture energy of the material; b is the length scale introduced to approximate a sharp crack by a diffuse damage band; the loading is described via the imposed displacement u^* . The essential boundary conditions for d are to prevent damage from initiating at either end of the bar. For this particular phase-field model, the damage boundedness condition $0 \leq d \leq 1$ is automatically satisfied [6]. As only monolithic loadings are herein considered, damage irreversibility $\dot{d} \geq 0$ is also fulfilled without any special consideration. Note however that, if needed, it can be enforced quite straightforwardly using various techniques; see [23] for details. As can be seen, PFM involves the solution of two coupled partial differential equations: the equilibrium equation and the damage evolution equation. This usually leads to a misunderstanding that PFM is just another gradient-enhanced damage model developed by [56]. It has been computationally shown in [5,11] that when the length scale approaches zero, the PFM approaches a fracture mechanics model rather than a damage model. Note that body forces are omitted for simplicity. As can be seen from the non-homogeneous Dirichlet boundary condition $u(x=L) = u^*$, we adopt a displacement control to trace the entire equilibrium path as snap-back does not occur for problems considered in this work. If needed, arc-length control can be used; see, e.g., [57].

2.2. Weak Forms and Finite Element Implementation

The weak form of the above governing equations is given by:

$$\int_0^L \frac{\partial u}{\partial x} (1-d)^2 E_0 \frac{\partial \delta u}{\partial x} dx = 0 \quad (7)$$

$$\int_0^L \left(\frac{G_c}{b} + E_0 \epsilon^2\right) d \delta d + G_c b \frac{\partial d}{\partial x} \frac{\partial \delta d}{\partial x} = \int_0^L E_0 \epsilon^2 \delta d dx \quad (8)$$

for test functions δu and δd .

In order to simplify the notation, let us denote the vector of state variables by $\mathbf{x} = (u, d)$ and the vector of model parameters by $\boldsymbol{\mu} = (L, E_0, G_c, b, u^*) \in \boldsymbol{\Omega}$. This weak form is discretized by using standard finite elements, resulting in the following discrete equations (see Remark 1):

$$\mathbf{K}^{aa}(\mathbf{x}; \boldsymbol{\mu}) \mathbf{a}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{0}, \quad (9)$$

$$\mathbf{K}^{dd}(\mathbf{x}; \boldsymbol{\mu}) \bar{\mathbf{a}}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}). \quad (10)$$

where $\mathbf{a}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^N$ and $\bar{\mathbf{a}}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^N$ are the nodal displacement vector and damage vector, respectively. $\mathbf{K}^{aa}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ and $\mathbf{K}^{dd}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ are the matrices, and $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^N$ is the (external) force vector (we admit this terminology is not entirely correct, but as there is only one vector in the system, we hope that it does not cause any confuse). Here, N is the number of grid points. Note that one has to solve the displacement equation with the constraint $a_N = u^*$ and the damage equation with $\bar{a}_1 = \bar{a}_N = 0$. The matrices and the force vector in the above are given by:

$$\mathbf{K}^{aa} = \int_0^L (\nabla \mathbf{N})^T E \nabla \mathbf{N} dx, \quad E = (1 - \bar{d})^2 E_0, \quad (11)$$

$$\mathbf{K}^{dd} = \int_0^L \left[\left(\frac{G_c}{b} + E_0 \bar{\epsilon}^2 \right) \mathbf{N}^T \mathbf{N} + G_c b (\nabla \mathbf{N})^T \nabla \mathbf{N} \right] dx, \quad (12)$$

$$\mathbf{f} = \int_0^L E_0 \bar{\epsilon}^2 \mathbf{N}^T dx, \quad (13)$$

where quantities with a bar indicate fixed values (as a staggered solver is used, which will be shortly discussed); \mathbf{N} is the row vector of the shape functions; and $\nabla \mathbf{N}$ denotes the row vector of the first derivatives of the shape functions. The symbol \square^T denotes the transpose operator.

Remark 1. Due to the non-convexity of the energy functional in terms of both displacements and damage, monolithic solvers are not very robust. That is why staggered solvers or alternating minimization solvers, where the off-diagonal coupling matrices are not needed, are popular in phase-field fracture [3,6].

The system of Equations (9)–(10) is solved using a staggered solver, also known as the Alternating Minimization (AM) solver. That is, one fixes the damage in the equilibrium equation and solves for the displacement. Next, the updated (latest) displacement field is used to calculate the driving force and substituted into the damage evolution equation to solve for the damage field. The process is repeated until convergence. These steps are summarized in Algorithm given in Box 1, where the notation \mathbf{a}_{n+1}^k signifies the nodal displacement vector at load increment $n + 1$ and AM iteration k ; and \mathbf{a}_{n+1} denotes the converged displacement vector at step $n + 1$. Basically, there are two computational bottlenecks:

1. the solution of two $N \times N$ systems for each AM iteration k and
2. the evaluation of the force vector \mathbf{f} and matrices \mathbf{K}^{aa} and \mathbf{K}^{dd} .

which render PFM computations time-consuming and may not be feasible in situations where they have to be repeatedly executed a large number of times. Reduced-Order Models (ROM) can be applied for such problems to obtain a lower order efficient model. They are introduced subsequently in Section 3. Note that we do not focus on the cost of the robust-but-slow AM solver and refer to [23] for a discussion on efficient solvers for phase-field models. We refer to Remark 2 for extension to 2D/3D problems.

Box 1. Quasi-static brittle PFM: AM solver for load step $n + 1$.

1. **Initialization:** $(\mathbf{a}_{n+1}^0, \bar{\mathbf{a}}_{n+1}^0) = (\mathbf{a}_n, \bar{\mathbf{a}}_n)$, $k = 1$
2. Do AM iterations: while $|\bar{\mathbf{a}}_{n+1}^k - \bar{\mathbf{a}}_{n+1}^{k-1}| > \eta$ ($\eta = 10^{-5}$ is the precision)
 - (a) **Displacement sub-problem:** solve for \mathbf{a}_{n+1}^k with fixed $\bar{\mathbf{a}}_{n+1}^{k-1}$

$$\mathbf{K}^{aa} \mathbf{a}_{n+1}^k = \mathbf{0} \quad \text{subject to } \mathbf{a}_{n+1}^k[N] = u_{n+1}^*$$
 - (b) **Phase-field sub-problem:** solve for $\bar{\mathbf{a}}_{n+1}^k$ with fixed \mathbf{a}_{n+1}^k

$$\mathbf{K}^{dd} \bar{\mathbf{a}}_{n+1}^k = \mathbf{f} \quad \text{subject to } \bar{\mathbf{a}}_{n+1}^k[1] = \bar{\mathbf{a}}_{n+1}^k[N] = 0$$
 - (c) Set $k = k + 1$
3. Update nodal unknowns: $(\mathbf{a}_{n+1}, \bar{\mathbf{a}}_{n+1}) = (\mathbf{a}_{n+1}^k, \bar{\mathbf{a}}_{n+1}^k)$

Remark 2. For hybrid isotropic brittle fracture PFMs (see, e.g., [22]), the displacement and damage sub-problems are linear. For two- (2D) or three- (3D) dimensional problems, one needs to solve an $(N \times nsd) \times (N \times nsd)$ system for the displacements and an $N \times N$ system for the damage where nsd is the number of spatial dimensions. Therefore, the proposed ROMs, presented in the next section, can be equally applied to 2D and 3D problems when a hybrid brittle fracture PFM is used. For tension/compression asymmetric anisotropic PFMs [6,58], the displacement

sub-problem becomes nonlinear, and thus, one has to slightly modify the POD-(M)DEIM ROM as discussed in Remark 3.

3. Reduced-Order Modelling

This section presents two reduced-order models, one based on the mesh-based POD method presented in Section 3.1 and the other based on a meshfree approach (cf. Section 3.2).

3.1. Mesh-Based Approach

Essentially, a ROM is carried out in two phases: a computationally-expensive *offline phase* and a computationally-efficient *online phase*. In the offline phase, a set of samples is collected from a standard analysis of the full-order model (in this context, the PFM simulation). This information is employed to construct a reduced-order model that is used in the online phase. In practice, the POD is applied to a set of samples collected from a full-order PFM analysis to compute a set of basis vectors (Section 3.1.1). These basis vectors are later used in the POD-DEIM to build the approximation for the force vector \mathbf{f} (cf. Section 3.1.2) and in the POD-(M)DEIM for the matrices \mathbf{K}^{aa} and \mathbf{K}^{dd} (Section 3.1.3).

3.1.1. Parameterized and Nonlinear ROM Based on the Projection Framework

Consider the system of equations in Equations (9) and (10). An ROM of this system can be obtained by approximating the full-state vectors \mathbf{a} and $\bar{\mathbf{a}}$ as a linear combination of m and \bar{m} basis vectors as follows,

$$\mathbf{a} = \mathbf{V}\mathbf{a}_r, \quad \bar{\mathbf{a}} = \bar{\mathbf{V}}\bar{\mathbf{a}}_r, \quad (14)$$

where $\mathbf{a}_r \in \mathbb{R}^m$ and $\bar{\mathbf{a}}_r \in \mathbb{R}^{\bar{m}}$ are the reduced-order versions of the displacement and damage field, respectively. $\mathbf{V} = [v_1^u \ v_2^u \ \dots \ v_m^u] \in \mathbb{R}^{N \times m}$ and $\bar{\mathbf{V}} = [v_1^d \ v_2^d \ \dots \ v_{\bar{m}}^d] \in \mathbb{R}^{N \times \bar{m}}$ are the orthonormal bases. Now, at every iteration, step k in Box 1, projecting the system of Equations (9) and (10) onto the reduced spaces formed by these bases yields the lower order model as follows:

$$\mathbf{K}_r^{aa}(\mathbf{x}^k; \mu) \mathbf{a}_r = \mathbf{0}, \quad (15)$$

$$\mathbf{K}_r^{dd}(\mathbf{x}^k; \mu) \bar{\mathbf{a}}_r = \mathbf{f}_r(\mathbf{x}^k; \mu), \quad (16)$$

where the reduced-order matrices and vector are given by:

$$\mathbf{K}_r^{aa}(\mathbf{x}^k; \mu) = \mathbf{V}^T \mathbf{K}^{aa}(\bar{\mathbf{x}}^k; \mu) \mathbf{V}, \quad (17)$$

$$\mathbf{K}_r^{dd}(\mathbf{x}^k; \mu) = \bar{\mathbf{V}}^T \mathbf{K}^{dd}(\bar{\mathbf{x}}^k; \mu) \bar{\mathbf{V}}, \quad (18)$$

$$\mathbf{f}_r(\mathbf{x}^k; \mu) = \bar{\mathbf{V}}^T \mathbf{f}(\bar{\mathbf{x}}^k; \mu). \quad (19)$$

The ROM task is to find the bases \mathbf{V} and $\bar{\mathbf{V}}$ so that $m \ll N$ and $\bar{m} \ll N$, then to solve the system of Equations (15)–(19) using Box 1. How to determine these bases is subsequently discussed in Box 3. This task is simple and straightforward when the original system is an affine and linear system; and it would be complex for nonlinear and coupled systems. The construction and solution of the system (15)–(19) over previous solutions, i.e., $(\bar{\mathbf{x}}^k = (\mathbf{V}\mathbf{a}_r^k, \bar{\mathbf{V}}\bar{\mathbf{a}}_r^k))$, and the parameter space are nontrivial. For instance, at each AM iteration k , the ROM requires firstly the re-construction of the full-order system matrices \mathbf{K}^{aa} , \mathbf{K}^{dd} and vector \mathbf{f} corresponding to the parameters and previous solutions, which still depend on the dimension of the full model, and secondly, the projection of these matrices/vectors on the reduced spaces to obtain the corresponding reduced-order matrices and vector. This pure POD model may result in a longer and much more complicated computation than the original FOM. In this study, we implemented DEIM and (M)DEIM [31,35] in the offline stage to get rid of the full-dimension dependence of the ROM matrices and vector. Precisely, DEIM was used to approximate the nonlinear external force, that is the right-hand side of the damage

sub-problem equation (Section 3.1.2), and (M)DEIM was adopted to approximate the nonlinear matrices (Section 3.1.3).

3.1.2. DEIM

The DEIM was applied to approximate the nonlinear function of the external force in Equations (19) and (13). The idea is to select sampling of the nonlinear terms combined with interpolation among these samples to recover an approximate nonlinear evaluation, as follows:

$$\mathbf{f}_r(\mathbf{x}^k; \mu) = \bar{\mathbf{V}}^T \mathbf{f}(\mathbf{V} \mathbf{a}_r^k; \mu) \approx \bar{\mathbf{V}}^T \Phi (\mathbf{P}^T \Phi)^{-1} \mathbf{P}^T \mathbf{f}(\mathbf{V} \mathbf{a}_r^k; \mu), \quad (20)$$

where matrix Φ is the POD basis vectors of the nonlinear snapshots obtained from the FOM, where $\Phi = [\Phi_1, \dots, \Phi_{n_f}] \in \mathbb{R}^{n \times n_f}$. Matrix $\mathbf{P} = [\mathbf{e}_1, \dots, \mathbf{e}_{n_f}] \in \mathbb{R}^{n \times n_f}$ is the n_f -indices matrix provided by DEIM, which we used here as the original proposed in [31], where $\mathbf{e}_i = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ is the i^{th} column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, n_f$. Note that Φ is acting as a projector of the basis on vector \mathbf{f} (Equation (21)), while \mathbf{P} is acting as a filter matrix that returns the non-zero components of \mathbf{f} (Equation (22)). In other words, it is used to define the reduced mesh (see Section 3.1.3).

Let us define:

$$\mathbf{D} := \bar{\mathbf{V}}^T \Phi (\mathbf{P}^T \Phi)^{-1} \in \mathbb{R}^{\bar{m} \times n_f}, \quad (21)$$

$$\mathbf{F}_r(\mathbf{a}_r^k; \mu) := \mathbf{P}^T \mathbf{f}(\mathbf{V} \mathbf{a}_r^k; \mu) \in \mathbb{R}^{n_f}. \quad (22)$$

The POD-DEIM reduced-order of the external force in (19) now has the form:

$$\mathbf{f}_r(\mathbf{x}^k; \mu) = \mathbf{D} \mathbf{F}_r(\mathbf{a}_r^k; \mu), \quad (23)$$

The complexity of the evaluation of \mathbf{f}_r is now reduced to the evaluation of \mathbf{F}_r and a matrix multiplication (note that the computation of \mathbf{D} was carried out in the offline phase). The POD-DEIM basis vectors were obtained using the algorithm in Box 2 where N_s represents the number of sampling points (the number of points that discretize the parameter space), and N_T , in this paper, denotes the number of load increments. We refer to Box A1 for the algorithm of $\text{POD}(\mathbf{X}_f, \epsilon_f)$. Note that the DEIM was used in the offline phase (see Box 3) to build the reduced matrices and vectors for the approximation of the FE external force and matrices.

Box 2. DEIM algorithm.

```

input :  $\mathbf{X}_f = [\mathbf{f}(\mathbf{x}^1; \mu_1), \dots, \mathbf{f}(\mathbf{x}^{N_T}; \mu_1)], \dots, [\mathbf{f}(\mathbf{x}^1; \mu_{N_s}), \dots, \mathbf{f}(\mathbf{x}^{N_T}; \mu_{N_s})] \in \mathbb{R}^{N \times (N_s \times N_T)}, \epsilon_f$ 
output: Basis  $\Phi \in \mathbb{R}^{N \times n_f}$ , set of indices  $\mathbf{P} \in \mathbb{R}^{n_f}$ 
 $[\mathbf{f}_1, \dots, \mathbf{f}_{n_f}] = \text{POD}(\mathbf{X}_f, \epsilon_f)$ 
 $[\rho, \Phi_1] = \max |\mathbf{f}_1|^a$ 
 $\mathbf{U} = [\mathbf{f}_1], \quad \mathbf{P} = [\mathbf{e}_{\Phi_1}]^b, \quad \Phi = [\Phi_1]$ 
for  $i = 2 : n_f$  do
    Solve  $(\mathbf{P}^T \mathbf{U}) \mathbf{c} = \mathbf{P}^T \mathbf{f}_i$  for  $\mathbf{c}$  and  $\mathbf{res} = \mathbf{f}_i - \mathbf{U} \mathbf{c}$ 
     $[\rho, \Phi_i] = \max |\mathbf{res}|$ 
     $\mathbf{U} = [\mathbf{U} \mathbf{f}_i], \quad \mathbf{P} = [\mathbf{P} \mathbf{e}_{\Phi_i}], \quad \Phi = \begin{bmatrix} \Phi \\ \Phi_i \end{bmatrix}$ 
end for

```

^a This max function returns the maximum value ρ at location Φ_1 .

^b \mathbf{e}_i is the i^{th} column of an identity matrix.

3.1.3. (M)DEIM

The (M)DEIM was applied to approximate the nonlinear matrices in Equations (17) and (18). The idea is to express the nonlinear matrices (i.e., $\mathbf{K}^{dd}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})$) in vector format, then apply the DEIM to approximate it. Without loss of generality, let us consider $\mathbf{K}^{dd}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})$ (the matrix of the damage sub-problem) and define the vector version of it:

$$\mathbf{k}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu}) := \text{vec}(\mathbf{K}^{dd}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})) \in \mathbb{R}^{N^2}, \quad (24)$$

that is, \mathbf{k} is obtained by stacking the columns of \mathbf{K}^{dd} . Then, this vector is approximated as:

$$\mathbf{k}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu}) \approx \mathbf{k}(\mathbf{x}_r^k; \boldsymbol{\mu}) = \boldsymbol{\Phi}^{\bar{m}} \boldsymbol{\theta}^{\bar{m}}(\mathbf{x}^k; \boldsymbol{\mu}). \quad (25)$$

Here, $\boldsymbol{\Phi}^{\bar{m}} \in \mathbb{R}^{N^2 \times m_a}$ is a nonlinear-parameter-independent basis and $\boldsymbol{\theta}^{\bar{m}}(\mathbf{x}^k; \boldsymbol{\mu}) \in \mathbb{R}^{m_a}$ is the coefficient vector. Apply DEIM in Box 2 to a set of snapshots $\mathbf{X}_K = [\mathbf{k}(\mathbf{x}^1; \boldsymbol{\mu}_1), \dots, \mathbf{k}(\mathbf{x}^{N_T}; \boldsymbol{\mu}_1), \dots, \mathbf{k}(\mathbf{x}^{N_T}; \boldsymbol{\mu}_{N_s})]$ to obtain the basis $\boldsymbol{\Phi}^{\bar{m}}$ and the interpolation indices $\mathcal{I} \in \mathbb{R}^{N^2}$. The reduced vector is obtained by the following projection:

$$\begin{aligned} \mathbf{k}_r(\mathbf{x}_r^k; \boldsymbol{\mu}) &= \text{vec}(\bar{\mathbf{V}}^T \mathbf{K}^{dd}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu}) \bar{\mathbf{V}}) \\ &= (\bar{\mathbf{V}}^T \otimes \bar{\mathbf{V}}^T) \boldsymbol{\Phi}^{\bar{m}} \boldsymbol{\theta}^{\bar{m}}(\mathbf{x}^k; \boldsymbol{\mu}) \in \mathbb{R}^{m_a}. \end{aligned} \quad (26)$$

Let us define $\mathbf{D}^{\bar{m}} := (\bar{\mathbf{V}}^T \otimes \bar{\mathbf{V}}^T) \boldsymbol{\Phi}^{\bar{m}}$, which is a precomputed matrix; the online computation of the reduced vector becomes:

$$\mathbf{k}_r(\mathbf{x}_r^k; \boldsymbol{\mu}) = \mathbf{D}^{\bar{m}} \boldsymbol{\theta}^{\bar{m}}(\mathbf{x}^k; \boldsymbol{\mu}), \quad (27)$$

where:

$$\boldsymbol{\theta}^{\bar{m}}(\mathbf{x}^k; \boldsymbol{\mu}) = (\boldsymbol{\Phi}_{\mathcal{I}}^{\bar{m}})^{-1} \mathbf{k}_{\mathcal{I}}(\mathbf{x}; \boldsymbol{\mu}). \quad (28)$$

Then, the (M)DEIM approximation matrix $\mathbf{K}^{dd}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})$ is obtained by reversing the $\text{vec}(\cdot)$ operation. The crucial step in the online evaluation of \mathbf{k}_r is the computation of $\mathbf{k}_{\mathcal{I}}(\mathbf{x}; \boldsymbol{\mu})$. Within the framework of the finite element method, a reduced mesh concept, also called the reduced integration domain, can be used to evaluate $\mathbf{k}_{\mathcal{I}}(\mathbf{x}; \boldsymbol{\mu})$ efficiently. The basic idea is to loop over only elements belonging to \mathcal{I} that contribute to the stiffness matrix. For more details of this technique, we refer to [35,59]. The exact same algorithm applies for $\mathbf{K}^{aa}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})$ by defining $\mathbf{k}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu}) := \text{vec}(\mathbf{K}^{aa}(\tilde{\mathbf{x}}^k; \boldsymbol{\mu})) \in \mathbb{R}^{N^2}$ and replacing \bar{m} by m and $\bar{\mathbf{V}}$ by \mathbf{V} .

In summary, the offline POD-(M)DEIM is presented in Box 3. The procedure is as follows. First, N_s sampling points in the parameter space Ω are generated, and for every point $\boldsymbol{\mu}^i$, solve the corresponding FOM for $u^* \in [0, u_{\max}]$, i.e., the entire loading path. Snapshots of the nodal displacement vector, nodal damage vector, external force, and global matrices \mathbf{K}^{aa} and \mathbf{K}^{dd} for each load increment (designated by t_j) are stored. These snapshots are next used to build the bases for the POD, i.e., \mathbf{V} and $\bar{\mathbf{V}}$, and $\boldsymbol{\Phi}^f, \boldsymbol{\Phi}^m, \boldsymbol{\Phi}^{\bar{m}}$ for POD-(M)DEIM. Finally, the DEIM (Box 2) is applied to $\boldsymbol{\Phi}^f, \boldsymbol{\Phi}^m, \boldsymbol{\Phi}^{\bar{m}}$ to obtain the reduced matrices and vectors and the reduced mesh. Recall that $m_f, m_a, m_{\bar{a}}$ are the number of basis vectors for the force vector, matrices \mathbf{K}^{aa} and \mathbf{K}^{dd} , respectively.

Once all the bases are obtained and stored, online simulations of the brittle fracture problem for any $\boldsymbol{\mu} \in \Omega$ can be performed using the procedure given in Box 4. We used $\mathbf{a}_{r,n+1}^k$ to denote the reduced-order nodal displacement vector at load increment $n+1$ of AM iteration k . This notation also applies to the reduced nodal damage vector. As can be seen that one has to modify the FOM code to have a POD-(M)DEIM ROM code. The meshfree Kriging model, presented in the next section, is a non-intrusive technique where one does not modify the FOM code.

Remark 3. For anisotropic PFMs, the displacement sub-problem is nonlinear and most often solved with the Newton–Raphson (NR) method. For each NR iteration, one has to solve $\mathbf{K}_T^{aa} \delta \mathbf{a} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{a})$ for the

displacement corrections. Therefore, in the POD-(M)DEIM, one simply applies the DEIM to the internal force vector $\mathbf{f}^{\text{int}}(\mathbf{a})$ and MDEIM to the tangent matrix \mathbf{K}_T^{aa} . This constitutes a small modification to our proposed POD algorithm. Thus, our POD ROM is inherently multi-dimensional. See [47] for an application of POD to nonlinear finite elements.

Box 3. Offline POD-(M)DEIM calculation.

Set: $m, \bar{m}, m_f, m_a, m_{\bar{a}}, \mu^s \in \Omega^{d \times N_s}$, number of load increments N_T

Initialize: $\mathbf{X}_a = []; \mathbf{X}_{\bar{a}} = []; \mathbf{X}_f = []; \mathbf{X}_{K^{aa}} = []; \mathbf{X}_{K^{dd}} = []$

Step 1: Collect the solution, matrix, vector snapshots

for $i = 1: N_s$ **do**

Solve the FOM (see Box 1), and store the data for every load increment

$$\mathbf{a}_{\text{local}}^i = [\mathbf{a}(t_1; \mu^i), \dots, \mathbf{a}(t_{N_T}; \mu^i)]$$

$$\bar{\mathbf{a}}_{\text{local}}^i = [\bar{\mathbf{a}}(t_1; \mu^i), \dots, \bar{\mathbf{a}}(t_{N_T}; \mu^i)]$$

$$\mathbf{f}_{\text{local}}^i = [\mathbf{f}(t_1; \mu^i), \dots, \mathbf{f}(t_{N_T}; \mu^i)]$$

$$\mathbf{K}(\mathbf{aa})_{\text{local}}^i = [\text{vec}(\mathbf{K}^{aa})(t_1; \mu^i), \dots, \text{vec}(\mathbf{K}^{aa})(t_{N_T}; \mu^i)]$$

$$\mathbf{K}(\mathbf{dd})_{\text{local}}^i = [\text{vec}(\mathbf{K}^{dd})(t_1; \mu^i), \dots, \text{vec}(\mathbf{K}^{dd})(t_{N_T}; \mu^i)]$$

Compress the local snapshots to the global ones:

$$\mathbf{X}_a = [\mathbf{X}_a \ \mathbf{a}_{\text{local}}^i], \quad \mathbf{X}_{\bar{a}} = [\mathbf{X}_{\bar{a}} \ \bar{\mathbf{a}}_{\text{local}}^i], \quad \mathbf{X}_f = [\mathbf{X}_f \ \mathbf{f}_{\text{local}}^i]$$

$$\mathbf{X}_{K^{aa}} = [\mathbf{X}_{K^{aa}} \ \mathbf{K}(\mathbf{aa})_{\text{local}}^i], \quad \mathbf{X}_{K^{dd}} = [\mathbf{X}_{K^{dd}} \ \mathbf{K}(\mathbf{dd})_{\text{local}}^i]$$

end for

Step 2: Extract the global bases using POD given in Box A1:

$$\mathbf{V} = \text{POD}(\mathbf{X}_a, m), \quad \bar{\mathbf{V}} = \text{POD}(\mathbf{X}_{\bar{a}}, \bar{m})$$

$$\Phi^f = \text{POD}(\mathbf{X}_f, m_f), \quad \Phi^m = \text{POD}(\mathbf{X}_{K^{aa}}, m_a), \quad \Phi^{\bar{m}} = \text{POD}(\mathbf{X}_{K^{dd}}, m_{\bar{a}})$$

Step 3: Perform the DEIM (Box 2) on $\Phi^f, \Phi^m, \Phi^{\bar{m}}$ to obtain the reduced matrices and vectors and the reduced mesh based on the reduced bases \mathbf{V} and $\bar{\mathbf{V}}$.

Box 4. Phase-field ROM for quasi-static brittle fracture: online phase.

1. **Initialization:** $(\mathbf{a}_{r,n+1}^0, \bar{\mathbf{a}}_{r,n+1}^0) = (\mathbf{a}_{r,n}, \bar{\mathbf{a}}_{r,n}), k = 1$
2. Do AM iterations

- (a) **Displacement sub-problem:** solve for $\mathbf{a}_{r,n+1}^k$ with fixed $\bar{\mathbf{a}}_{r,n+1}^{k-1}$
 - (i) Solve Equation (28) on the reduced mesh to obtain $\theta^m(\bar{\mathbf{a}}_{r,n+1}^{k-1}; \mu)$ (replace \bar{m} by m).
 - (ii) Reconstruct the reduce matrix $\mathbf{k}_r^{aa}(\bar{\mathbf{a}}_{r,n+1}^{k-1}; \mu)$ using Equation (27) (replace \bar{m} by m).
 - (iii) Obtain $\mathbf{K}_r^{aa}(\bar{\mathbf{a}}_{r,n+1}^{k-1}; \mu)$ by reversing the $\text{vec}()$ operation: $\mathbf{K}_r^{aa} = \text{reverseVec}(\mathbf{k}_r^{aa})$.
 - (iv) Solve $\mathbf{K}_r^{aa} \mathbf{a}_{r,n+1}^k = \mathbf{0}$
- (b) **Phase-field sub-problem:** solve for $\bar{\mathbf{a}}_{r,n+1}^k$ with fixed $\mathbf{a}_{r,n+1}^k$
 - (i) Solve Equation (28) on the reduced mesh to obtain $\theta^{\bar{m}}(\mathbf{a}_{r,n+1}^k; \mu)$.
 - (ii) Reconstruct the reduce matrix $\mathbf{k}_r^{dd}(\mathbf{a}_{r,n+1}^k; \mu)$ using Equation (27).
 - (iii) Obtain $\mathbf{K}_r^{dd}(\mathbf{a}_{r,n+1}^k; \mu)$ by $\mathbf{K}_r^{dd} = \text{reverseVec}(\mathbf{k}_r^{dd})$.
 - (iv) Obtain $\mathbf{f}_r(\mathbf{x}^k; \mu)$ using Equation (23).
 - (v) Solve $\mathbf{K}_r^{dd} \bar{\mathbf{a}}_{r,n+1}^k = \mathbf{f}_r$
- (c) **Set** $k = k + 1$

3. Update nodal unknowns: $(\mathbf{a}_{r,n+1}, \bar{\mathbf{a}}_{r,n+1}) = (\mathbf{a}_{r,n+1}^k, \bar{\mathbf{a}}_{r,n+1}^k)$

3.2. Meshfree Kriging Method

This section presents the Kriging model that we utilized, for which more details can be found in [53,54,60]. Generally, in order to apply the Kriging prediction model, one follows the algorithm in Figure 1. Basically, it also consists of two phases: the offline phase where data are collected by running the FOM and the predictor is built. In the online phase, the predictor is used to get the outputs for any given input. Note that this phase does not use any phase-field finite element code, resulting in a non-intrusive model (see Remark 4).

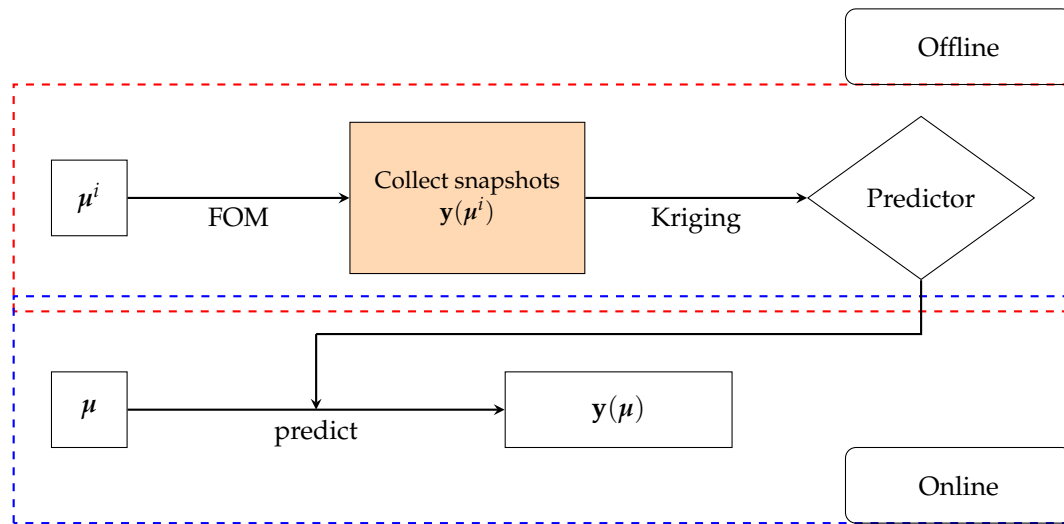


Figure 1. The online and offline procedures of the Kriging approach. FOM, Full-Order Model.

Consider an output of interest $y = \{y(\mathbf{x}; \boldsymbol{\mu}) | \boldsymbol{\mu} \in \Omega\}$ that varies within a parameter space Ω . Kriging models can be obtained by assuming that output $y(\mathbf{x}; \boldsymbol{\mu})$ can be described as a linear combination of a regression model and a stochastic process as follows [60]:

$$Y(\mathbf{x}; \boldsymbol{\mu}) = m(\mathbf{x}; \boldsymbol{\mu}) + Z(\mathbf{x}; \boldsymbol{\mu}), \quad (29)$$

where m is a regression model known as the deterministic trend, which globally approximates the parameter space. The regression model is a linear combination of p chosen functions f_j ,

$$m(\mathbf{x}; \boldsymbol{\mu}) = \sum_{j=1}^p \beta_j f_j(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu})^T \boldsymbol{\beta}, \quad (30)$$

where β_j are regression parameters and f_j are regressor functions. The stochastic process Z , which creates the localized deviation of the parameter space, is assumed to have zero mean and variance function:

$$\mathbb{E}[Z(\mathbf{x}; \boldsymbol{\mu}_i) \times Z(\mathbf{x}; \boldsymbol{\mu}_j)] = \sigma^2 \mathcal{R}(\theta, \boldsymbol{\mu}_i, \boldsymbol{\mu}_j), \quad (31)$$

where σ^2 is variance and $\mathcal{R}()$ is the correlation model with parameters θ .

We now consider a set of N_s design points denoted by $\boldsymbol{\mu}_d = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{N_s}\}$ to which the corresponding output are $\mathbf{y} = \{y(\mathbf{x}; \boldsymbol{\mu}_1), \dots, y(\mathbf{x}; \boldsymbol{\mu}_{N_s})\}$. The regressor \mathbf{F} with $F_{ij} = f_j(\mathbf{x}; \boldsymbol{\mu}_i)$ is given by a vector of p regressor functions:

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}_1) \dots \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}_{N_s})]^T. \quad (32)$$

By defining the correlation matrix $R(\theta)$ as the matrix of the stochastic process correlation between the i^{th} and j^{th} design points, we have:

$$R_{ij}(\theta) = \mathcal{R}(\theta, \mu_i, \mu_j), \quad i, j = 1, \dots, N_s. \quad (33)$$

The vector of correlations between an untried point, μ , and the design points is defined as:

$$\mathbf{r}(\mathbf{x}; \mu) = [\mathcal{R}(\theta, \mu_1, \mu), \dots, \mathcal{R}(\theta, \mu_{N_s}, \mu)]^T. \quad (34)$$

The predicted estimates, $\hat{\mathbf{y}}(\mathbf{x}; \mu)$, of the response $\mathbf{y}(\mathbf{x}; \mu)$ at an untried point μ , are given by:

$$\hat{\mathbf{y}}(\mathbf{x}; \mu) = \mathbf{f}(\mathbf{x}; \mu)^T \hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x}; \mu)^T R(\theta)^{-1} (\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\beta}}), \quad (35)$$

where $\hat{\boldsymbol{\beta}}$ is estimated from the data, using least squares regression:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^T R(\theta)^{-1} \mathbf{F})^{-1} \mathbf{F}^T R(\theta)^{-1} \mathbf{y}. \quad (36)$$

Once the regression model and the covariance function of a stochastic process have been determined, the prediction of \mathbf{y} can be done by using Kriging models. According to the Design and Analysis of Computer Experiments (DACE) [60], the correlations are restricted in the form:

$$\mathcal{R}(\theta, \mu_i, \mu_j) = \prod_{k=1}^d \mathcal{R}_k(\theta, \mu_i^k - \mu_j^k), \quad (37)$$

where d is the dimension of the parameter space. The correlation parameters can be determined by minimizing the log-likelihood for θ as:

$$\hat{\theta} = \arg \min_{\theta} [n \log \sigma^2 + \log(|R(\theta)|)], \quad (38)$$

where $|R(\theta)|$ is the determinant of the correlation matrix corresponding to the design points. Assuming a Gaussian process, the optimal coefficients $\hat{\boldsymbol{\beta}}$ of the correlation function solve:

$$\hat{\theta} = \min_{\theta} |R(\theta)|^{\frac{1}{n}} \sigma^2. \quad (39)$$

The corresponding maximum likelihood estimate of the variance, σ^2 , is:

$$\sigma^2 = \frac{1}{n} (\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\beta}})^T (\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\beta}}). \quad (40)$$

The model template in the DACE toolbox (Design and Analysis of Computer Experiments) has been discussed in [61].

Remark 4. As the Kriging model is non-intrusive in the sense that one can just use a PFM code as a black box to obtain the training data, it applies to any phase-field models, namely 2D/3D and isotropic/anisotropic models.

3.3. A Posteriori Error Estimations

The relative L_2 -norm and Root Mean Squared Error (RMSE) used to evaluate our reduced-order models are written as:

$$L_{2a} = \frac{\|\mathbf{a} - \hat{\mathbf{a}}\|_{L_2}}{\|\mathbf{a}\|_{L_2}}, \quad (41)$$

and:

$$rmse_{\mathbf{a}} = \sqrt{\frac{\sum_{i=1}^{N_{\text{test}}} (\hat{\mathbf{a}}_i - \mathbf{a}_i)^2}{N_{\text{test}}}}, \quad (42)$$

where $\hat{\mathbf{a}}$ is the approximate ROM solutions and \mathbf{a} represents the true solutions (precisely the FOM solutions). Note that $\hat{\mathbf{a}} = \mathbf{V}\mathbf{a}_r$, so both \mathbf{a} and $\hat{\mathbf{a}}$ have the same length.

4. Numerical Examples

In this section, the performances of the two proposed ROMs are evaluated for the 1D problem stated in Section 2. Even though one can, in general, consider the variation of the geometry (L), material parameters (E_0, G_c, b), and loadings (u^*), we present, for simplicity, results for $\boldsymbol{\mu} = (E_0, G_c) \in \boldsymbol{\Omega} \subset \mathbb{R}^2$. That is, we consider a traction bar of a fixed length $L = 25$, $b = L/100$, unit cross-section (units are deliberately left out here, given that they can be consistently chosen in any system), and that is subjected to the maximum load of $u_{\max} = 37.5$. Material constants E and G_c are varying parameters. Previous studies have shown that such a length scale is small enough [23] (see Remark 5). The bar is uniformly discretized with 1000 linear elements (element size $h < b/2$), resulting in $N = 1001$ grid points, which produce converged results; see Appendix B for a convergence analysis. The load is applied via prescribed displacement $u^* \in [0, u_{\max}]$ at the right-most node. The entire loading process is divided into N_T equal steps. The quantities of interests are (i) the displacement field, strain field, and damage field along the bar at u_{\max} and (ii) the stress-strain curve, which is obtained from the load-displacement curve where load is the reaction force at the right-most node and displacement is the applied displacement u^* . These quantities computed using an FOM serve as the output of interest $y = y(\mathbf{x}; \boldsymbol{\mu})$ used to build the Kriging predictor. All simulations were carried out using an in-house code written in MATLAB.

Concretely, we considered the parameter space $E_0 \times G_c = [1, 10] \times [1, 10]$. The offline calculation can face the “curse of dimensionality” if we generate the samples using a full-factorial experiment. For example, let $N_{E_0} = N_{G_c} = 50$ be the sample numbers of each variable; the computations require $N_{E_0} \times N_{G_c} = 2500$ runs for evaluating the solution for every possible combination of every possible design value. Therefore, in this paper, we have used the Latin Hypercube Sampling method (LHS) [62] to generate statistically-optimal sampling points. The typical behaviour of this traction softening bar is as follows; see Figures 5 and 6. When the load is small, i.e., smaller than $0.8u_{\max}$, the bar is still in the elastic regime, albeit with a small damage due to the lack of an elastic domain of the chosen PFM. A further increment in loading leads to the initiation of a crack (a point in this 1D case) in the centre of the bar. Note that we have not introduced any imperfections in the bar to trigger damage localization; see, e.g., [55]. The fact that the point of damage localization is the centre of the bar is due to the perfect symmetry of the problem. The strain and damage are now localized in a small region centred at this cracking point.

Remark 5. We note that for the selected phase-field model (Section 2), the solution depends on the length scale b (see Appendix B), and one should consider its variation. However, if one adopts our length scale-insensitive PFM presented in [7], the result is independent of b .

Remark 6. In order to make sure the ROMs can capture strain localization (or crack initiation), we have applied a large displacement of $u_{\max} = 37.5$ to make sure that for all considered sampling points (i.e., all pairs of (E_0, G_c) used in the offline phase), damage localization happens at least once.

We first present the ROM results in Section 4.1. That is, we generated sampling points, built various ROMs (with different accuracies), and selected the best ROM, by evaluating the ROMs with the FOM for a given random (E_0, G_c) , to be used in online computations. The construction of the Kriging’s model was discussed in Section 4.2. Finally, Section 4.3 presents the comparison between

the POD ROM and Kriging model by evaluating them against the FOM for some random values $(E_0, G_c) \in \Omega$.

4.1. ROM Results

To illustrate the performance of the POD-(M)DEIM approach, we generated $N_s = 60$ sample points in Ω using the LHS, and built the ROM using the algorithm given in Box 3. This number of sampling points was based on our experiences with ROM; see for instance [42]. Precisely five ROMs, cf. Table 1, have been built to obtain the (M)DEIM bases of \mathbf{K}^{aa} , \mathbf{K}^{dd} , and \mathbf{f} . Figure 2 shows the snapshot spectrum and the cut-off lines corresponding to the $\epsilon_{\text{POD}} = 10^{-9}$ of these matrices and vector. Concretely, we have used Step 1 in the algorithm in Box A1 (Appendix A) to get the snapshot spectrum and Step 2 to get the number of required basis vectors. Table 1 presents the details of the (M)DEIM bases of each ROM. The number of M(DEIM) bases varied depending on the captured energy. More energy required a higher number of bases.

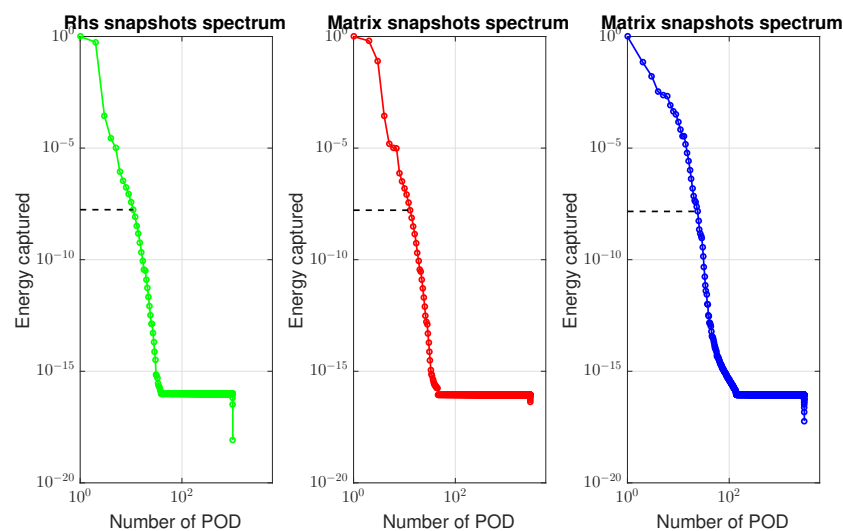


Figure 2. Vector and matrices' snapshots spectrum. Left is the vector \mathbf{f} , middle the matrix \mathbf{K}^{dd} , and right the matrix \mathbf{K}^{aa} .

Table 1. ROM characteristics.

ϵ_{POD}	10^{-6} (ROM ₁)	10^{-7} (ROM ₂)	10^{-8} (ROM ₃)	10^{-9} (ROM ₄)	10^{-10} (ROM ₅)
$m_a - m_{\bar{a}} - m_f$	17 – 6 – 5	22 – 12 – 10	23 – 12 – 10	27 – 15 – 13	28 – 16 – 14

Now, for each (M)DEIM in ROM_{*i*}, we built the POD basis of the solution's snapshot (displacement and damage fields) with the assumption that the number of POD bases for the displacement and that for the damage field were equal, i.e., $m = \bar{m}$. A number of cases corresponding to $m \in \{2, 100\}$ were considered. The POD-(M)DEIM was then validated against the FOM and pure POD approach for a random pair of (E_0, G_c) . Figure 3 presents the L_2 -norm relative error (L_{2f} for the reaction force) of ROM_{*i*}, with $i = 1, \dots, 5$ and pure POD in comparison with FOM. In terms of accuracy, the pure POD performed excellently in the range of (20 – 40) POD basis. When more bases were used, the accuracy was reduced due to the fact that more noise and error were added into the model. Meanwhile, the accuracy of ROM_{*i*} increased when the captured energy increased. However, the accuracy was not much different when ϵ_{POD} was in the range of 10^{-9} and 10^{-10} . Furthermore, the performance of ROM_{*i*} was much faster than the pure POD. We note that the performance (all of the computations were performed on a PC with Intel(R) Core(TM) i7-6820HQ CPU @ 2.70 GHz 2.7 GHz, RAM 8.0 GB (64-bit operating system, x64-based processor)), in terms of both accuracy and speed, of ROM₄ and ROM₅ was better than ROM₁, ROM₂, and ROM₃. The reason here is that the AM solver (cf. Box 3)

required iterations until convergence was satisfied. For the lower number of POD-(M)DEIM bases (meaning less accuracy of the model), it required more iterations to get convergence. For this reason, we selected ROM₅ with $m = 20$, $\bar{m} = 20$, and $m_a - m_{\bar{a}} - m_f = 28 - 16 - 14$ as the best ROM to be used in other online calculations.

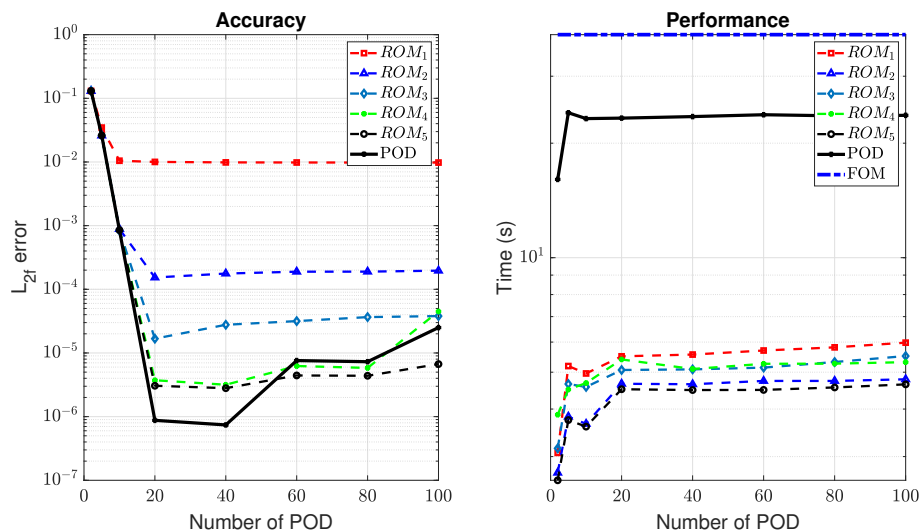


Figure 3. Performance of the POD-Matrix Discrete Empirical Interpolation Method ((M)DEIM) vs. pure POD. The number of POD indicates the POD of the solution vectors (m and \bar{m}).

4.2. Kriging Results

In order to verify the Kriging model against the sampling numbers, we generated several samples using LHS with N_s from 10–5000, ran the FOM, and collected the output of interest, then built the Kriging model as discussed in Section 3.2; see also Figure 1. The second-order polynomial was chosen as a regression model, while the Gaussian function served as a correlation model. After that, a random pair of (E_0, G_c) was selected to test the Kriging’s predictor. Figure 4 shows the relative L_2 -norm and RMSE of the reaction force, the construction, and the prediction time, respectively. The relative error reduced when the number of samples increased (see Remark 7 for an elaboration on this error behaviour). We note here that the L_2 -norm error and the RMSE produced similar values. The prediction time was generally relatively constant; however, the model’s construction time was increased when the sampling number increased, for example, $t_{build} = 0.1025$ s for $N_s = 50$ samples; however, it increased rapidly to $t_{build} = 1.17 \times 10^3$ s for $N_s = 4000$. From here, to avoid over-fitting, we can use $N_s = 4000$ (instead of $N_s = 1000$) as our Kriging model to compare with ROM. Please note here that although t_{build} was considered large for $N_s = 4000$, it was computed one time at the offline phase, so it did not affect the performance of the prediction.

Remark 7. Actually, the relative error in Figure 4 reduced significantly in the interval $[1, 1000]$ of samples, and then, it fluctuated, but did not go down to zero. From the data analysis viewpoint, the more data, the greater the accuracy of the model. However, it seemed to not be the case here. This was probably due to the discontinuity of the data (damage localization), and the Kriging model could not improve the accuracy around the discontinuity, although more data were introduced.

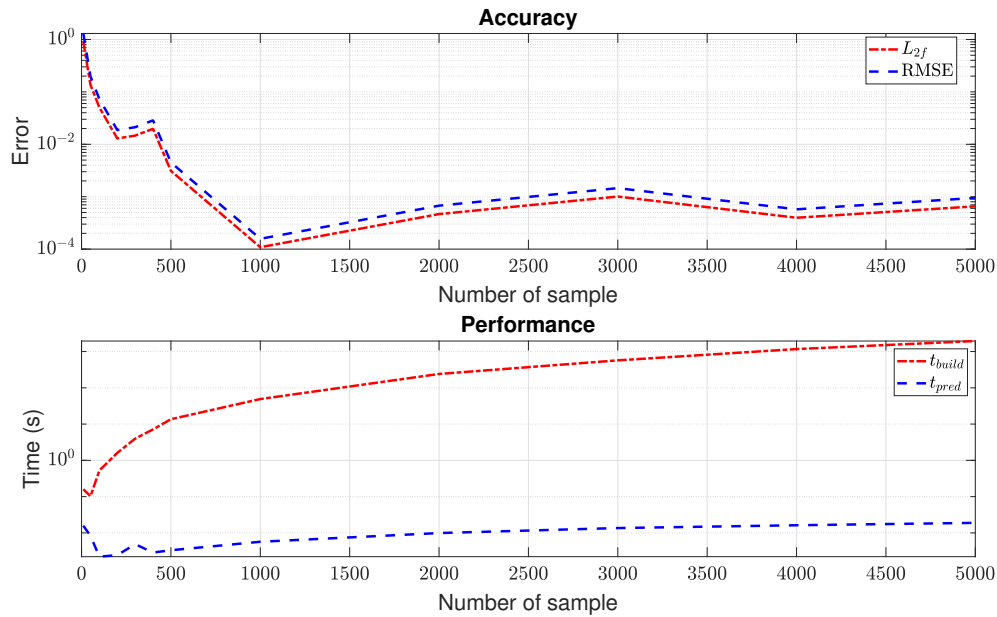


Figure 4. Relative errors and computational time of the Kriging model for different sampling numbers.

4.3. ROM vs. Kriging

A random sample test with $N_{test} = 10$ was generated to compute the relative errors of the ROM and Kriging prediction in comparison with the FOM. The averaged errors and computational time of each model are recorded and given in Table 2. Although the ROM provided more accurate solutions than Kriging (for example, relative error L_{2f} of 10^{-4} vs. 10^{-3}), the ROM online phase was much slower than the Kriging model. Concretely, ROM's speedup factor, compared with FOM, was approximately eight-times, while the Kriging's was approximately 1100.

Table 2. FOM, ROM, and KRI (Kriging) comparisons. The designation 20-20-28-16-14 refers to the ROM₅ with $m = \bar{m} = 20$ and $m_a - m_{\bar{a}} - m_f = 28 - 16 - 14$. Notations L_{2f}, L_{2u}, L_{2d} represent the L_2 error for the reaction force at the right-most node, the nodal displacement, and nodal damage vector, respectively. The time in this table indicates the online computation time.

Parameter	FOM	ROM	KRI
N	1001	20-20-28-16-14	1
t_{CPU} (s)	64.1	8.1	5.61×10^{-3}
L_{2f}		1.74×10^{-4}	9.63×10^{-3}
L_{2u}		4.45×10^{-3}	6.41×10^{-3}
L_{2d}		2.01×10^{-3}	8.21×10^{-3}

We now present the mechanical behaviour of this traction bar for the case $(E_0, G_c) = (3.705, 4.332)$. The response of the bar, obtained using the FOM, in terms of the load-displacement curve is given in Figure 5, where F is the reaction force at the right-most node. As this phase-field model lacked an elastic domain, the pre-peak behaviour was not linear, as damage was non-zero immediately upon load application. When the peak was reached, the bar was suddenly broken into two parts reflected by a sharp drop in the load-displacement curve. The evolution in time of the displacement, damage, and strain field is shown in Figure 6 for three time instances: $0.4u_{max}$, $0.8u_{max}$, and $0.9u_{max}$ as marked in Figure 5. Evidently, there was a strong localization of damage and strain at the middle of the bar. It is interesting to see whether ROM solutions can capture this strain localization phenomenon.

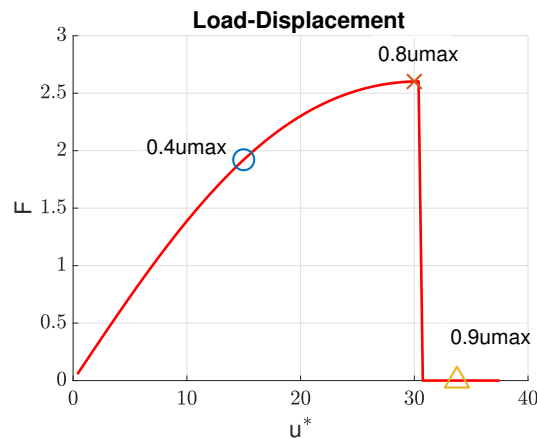


Figure 5. Load-displacement response of the bar for $(E_0, G_c) = (3.705, 4.332)$.

Figures 7–10 present the output of interest: (i) the displacement field, strain field, and damage field along the bar at u_{\max} and (ii) the stress-strain curve for two sets of $\mu = (E_0, G_c)$. Here, $\mu_1 = (3.705, 4.332)$ and $\mu_2 = (3.535, 9.639)$. In general, the mesh-based and meshfree approaches provided solutions in very good agreement with the full-order model. For μ_1 , the bar was completely broken, and thus, there was a jump in the displacement field (Figure 7a), as well as strain localizations; see Figures 8a and 9a. The reason that the ROMs can capture strain localization is explained in Remark 6. For μ_2 , the damage was small, and the bar was still in the elastic regime. It can be seen that the Kriging model can capture the displacement jump (Figure 7a) and damage localization (Figure 8a), but not strain localizations, cf. Figure 9a, which involves a sudden jump of a very large magnitude.

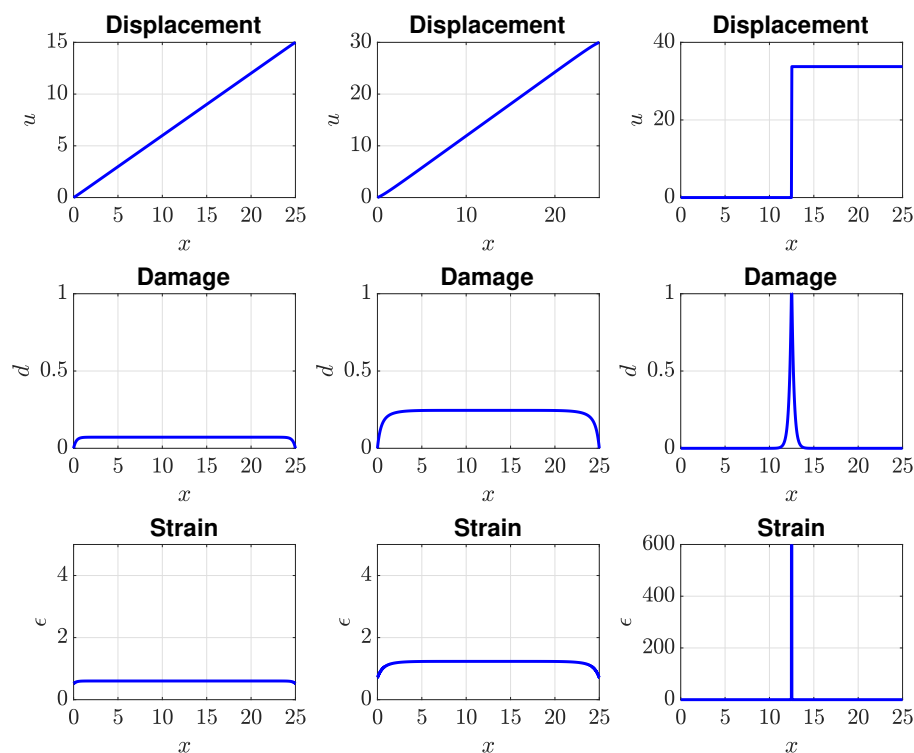


Figure 6. Evolution of the displacement, damage, and strain field for μ_1 : 40% u_{\max} (left), 80% u_{\max} (middle), 90% u_{\max} (right).

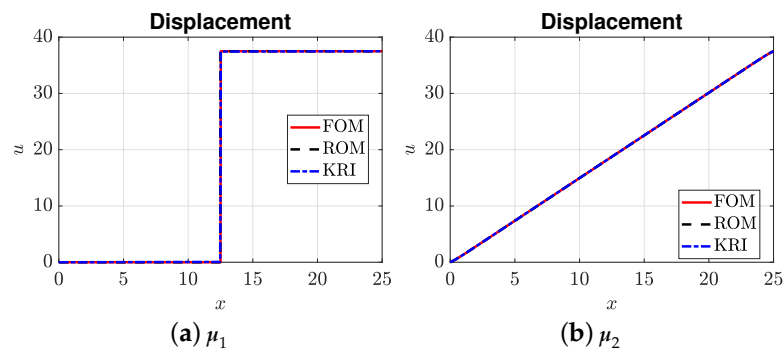


Figure 7. Displacement fields: for μ_1 (corresponding to a completely broken bar) and μ_2 .

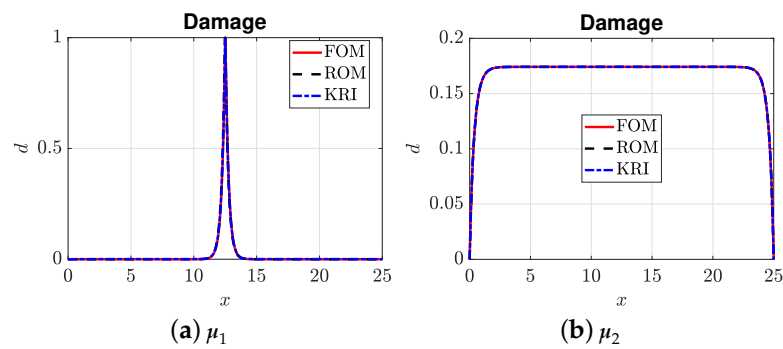


Figure 8. Damage fields: μ_1 (corresponding to a completely broken bar) and μ_2 .

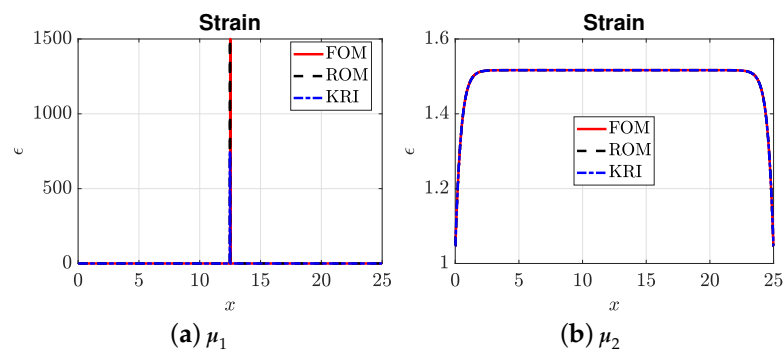


Figure 9. Strain field. Note that the Kriging model provided a maximum strain of only around 750 for μ_1 .

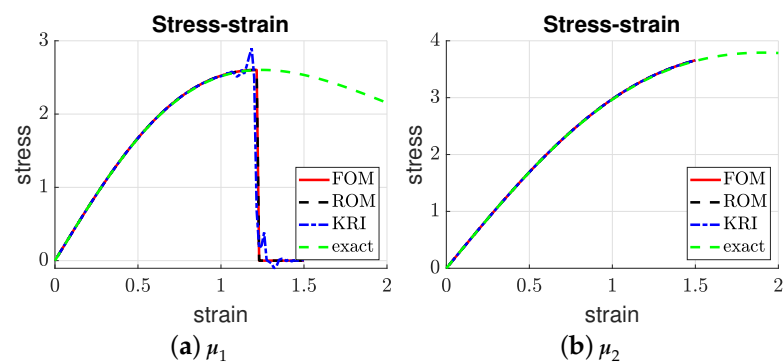


Figure 10. Stress-strain curve. We note the oscillations around the localization point for the Kriging model (a). The exact homogeneous stress-strain (valid up to crack initiation at a strain of 1.25) is given in Appendix B.

5. Conclusions

Within the context of parametrized brittle fracture mechanics, we have presented two classes of reduced-order phase-field models. They can be used to carry out a very large number of computations required in different situations efficiently, ranging from parameter sensitivity analysis for brittle fracture problems, parameter estimation for phase-field brittle fracture models, to fracture constrained optimization problems. Our first ROM was a Proper Orthogonal Decomposition (POD)-based projection method that utilized the Discrete Empirical Interpolation Method (DEIM) and the Matrix Discrete Empirical Interpolation Method ((M)DEIM) to approximate the nonlinear vectors and matrices. The second was a meshfree Kriging model. For one-dimensional problems where Young's modulus and fracture energy vary, the POD-(M)DEIM ROM can speed up the online computations eight-times, whereas the Kriging model's speed up factor was 1100, albeit with a slightly lower accuracy. The greatest difference between the POD-(M)DEIM and Kriging model was the non-intrusive nature of the latter in the sense that one does not have to modify the full-order model code. Needless to say, all these computational savings were obtained with extensive offline computations using the full model.

Even though our reduced-order models were applied for a one-dimensional bar, we have shown that they are inherently multi-dimensional in nature. However, further works are required for 2D/3D fracture problems. It is possible that one might need to use more sampling points to capture different crack patterns. Additionally, the proposed models suffered from the following limitations

- They cannot be used for extrapolation, i.e., when the parameters are out of the bounds of the considered parameter space;
- The load has not been parametrized. That is, the maximum prescribed displacement u_{\max} is fixed.
- The Kriging model resulted in oscillations around the damage localization point.

We note that the first limitation is inherent to any interpolation-based methods and might be tackled using machine learning methods. It is straightforward to overcome the second issue by just building a POD for u_{\max} . It is, however, very difficult to parametrize loadings in 2D and 3D. As far as the third issue is concerned, we anticipate that deep learning techniques might be helpful. For higher dimensional problems, it can become difficult. We are pursuing these paths and hope to publish them in the near future.

Author Contributions: Conceptualization, N.-H.N. and V.P.N.; Software, N.-H.N.; Supervision, V.P.N.; Writing—original draft, N.-H.N.; Writing—review & editing, V.P.N., J.-Y.W., T.-H.-H.L. and Y.D.

Funding: This research was funded by Australian Research Council via DECRAProject DE160100577 and the National Natural Science Foundation of China (51678246), the State Key Laboratory of Subtropical Building Science (2016KB12), the Fundamental Research Funds for the Central University (2015ZZ078), and the Scientific/Technological Project of Guangzhou (201607020005).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proper Orthogonal Decomposition Algorithm

For completeness, the POD algorithm to find the basis vectors directly from the snapshot matrix $\mathbf{X} \in \mathbb{R}^{n \times n_s}$ is given in Box A1. One can input either the desired number of basic vectors (m) or the tolerance ϵ_{POD} . Note that n can be N for the displacement/damage/force snapshot or N^2 for the matrix snapshots ($\mathbf{K}^{aa}, \mathbf{K}^{dd}$). n_s is the number of snapshots. The total energy and capturing energy are denoted by a and b , respectively.

Box A1. POD algorithm.

```

input : Snapshots  $\mathbf{X} \in \mathbb{R}^{n \times n_s}$ ,  $m$ , or  $\epsilon_{\text{POD}}$ 
output: Orthogonal basis  $\mathbf{V} \in \mathbb{R}^{n \times m}$ 
1.  $[\mathbf{V}, \Sigma] = \text{svd}(\mathbf{X})^a$ ,  $\Sigma = [\sigma_1, \dots, \sigma_{n_s}]$ 
2. Extract POD
if  $\epsilon_{\text{POD}}$  then
     $a = \sum_{i=1}^{n_s} \sigma_i^2$ ;  $b = 0$ ;  $i = 1$ 
    while  $((b/a)*100 < \epsilon_{\text{POD}})$  do
         $b = b + \sigma_i^2$ 
         $i = i + 1$ 
    end while
     $m = i$ 
     $\mathbf{V} = \mathbf{V}(:, 1:m)^b$ 
     $\Sigma = \Sigma(1:m)^c$ 
end if
if  $m$  then
    Basis matrix  $\mathbf{V} = \mathbf{V}(:, 1:m)$ 
     $\Sigma = \Sigma(1:m)$ 
end if

```

^a Here, svd is the MATLAB function for singular value decomposition; $\sigma_1 \geq \sigma_2 \dots \sigma_{n_s} \geq 0$.

^b This is MATLAB syntax to get the first m columns of the matrix \mathbf{V} .

^c This is to get the first m eigenvalues of Σ .

Appendix B. Exact Solutions

For the model given in Section 2, one can derive the exact solution for the homogeneous stress state, which is given by [23,55]:

$$\sigma = \left(\frac{G_c}{G_c + bE_0\epsilon^2} \right)^2 E_0\epsilon, \quad \sigma_c = \frac{3}{16} \sqrt{\frac{3E_0G_c}{b}} \quad (\text{A1})$$

where σ_c denotes the maximum stress, which depends on b . Note that the homogeneous solution is valid up to the point of damage localization or the peak load in the load-displacement curve.

We present mesh convergence analysis to justify the utilization of 1000 elements (or 1001 nodes) in all of our analyses. Three meshes consisting of 100, 500, and 1000 elements corresponding to the cases $h = b$, $h = b/5$, and $h = b/10$ were considered, where h denotes the element size. The results for the stress-strain are given in Figure A1a and the results for the damage profile in Figure A1b. As can be seen, 500 elements and 1000 elements yielded identical stress-strain data (thus, the same peak load), and 1000 elements were required to capture the cusp of the damage profile $d(x) = \exp(-|x|/b)$ [6]. Therefore, we denote the results obtained with 1000 elements' converged solutions and using this mesh to build the ROMs. Doing so, the error is due to ROM only, and thus, it is easier to understand the ROM behaviour.

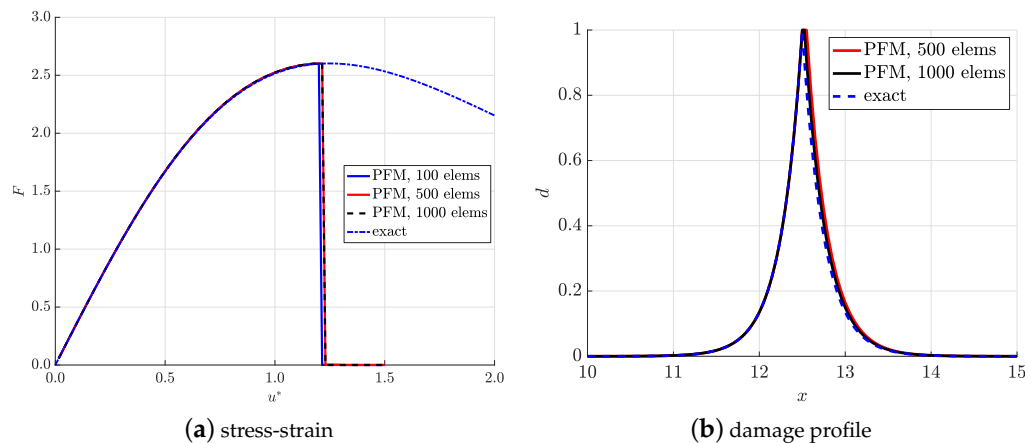


Figure A1. Mesh convergence analysis. PFM, Phase-Field Model.

References

- Quarteroni, A.; Manzoni, A.; Negri, F. *Reduced Basic Methods for Partial Differential Equations*; Springer: Berlin, Germany, 2016.
- Francfort, G.; Marigo, J. Revisiting brittle fracture as an energy minimization problem. *J. Mech. Phys. Solids* **1998**, *46*, 1319–1342. [\[CrossRef\]](#)
- Bourdin, B.; Francfort, G.; Marigo, J.J. Numerical experiments in revisited brittle fracture. *J. Mech. Phys. Solids* **2000**, *48*, 797–826. [\[CrossRef\]](#)
- Wu, J.Y. A unified phase-field theory for the mechanics of damage and quasi-brittle failure in solids. *J. Mech. Phys. Solids* **2017**, *103*, 72–99. [\[CrossRef\]](#)
- Wu, J.Y.; Qiu, J.F.; Nguyen, V.P.; Mandal, T.K.; Zhuang, L.J. Computational modelling of localized failure in solids: XFEM vs. PF-CZM. *Comput. Methods Appl. Mech. Eng.* **2018**, *345*, 618–643. [\[CrossRef\]](#)
- Miehe, C.; Welschinger, F.; Hofacker, M. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. *Int. J. Numer. Meth. Engng.* **2010**, *83*, 1273–1311. [\[CrossRef\]](#)
- Wu, J.Y.; Nguyen, V.P. A length scale insensitive phase-field damage model for brittle fracture. *J. Mech. Phys. Solids* **2018**, *119*, 20–42. [\[CrossRef\]](#)
- May, S.; Vignollet, J.; de Borst, R. A numerical assessment of phase-field models for brittle and cohesive fracture: Γ -Convergence and stress oscillations. *Eur. J. Mech. A/Solids* **2015**, *52*, 72–84. [\[CrossRef\]](#)
- Wu, J.Y. A geometrically regularized gradient-damage model with energetic equivalence. *Comput. Methods Appl. Mech. Eng.* **2018**, *328*, 612–637. [\[CrossRef\]](#)
- Feng, D.C.; Wu, J.Y. Phase-field regularized cohesive zone model (CZM) and size effect of concrete. *Eng. Fract. Mech.* **2018**, *197*, 66–79. [\[CrossRef\]](#)
- Mandal, T.K.; Nguyen, V.P.; Heidarpour, A. Phase field and gradient enhanced damage models for quasi-brittle failure: A numerical comparative study. *Eng. Fract. Mech.* **2019**, *207*, 48–67. [\[CrossRef\]](#)
- Larsen, C.J.; Ortner, C.; Sali, E. Existence of Solutions to A Regularized Model of Dynamic Fracture. *Math. Models Methods Appl. Sci.* **2010**, *20*, 1021–1048. [\[CrossRef\]](#)
- Bourdin, B.; Larsen, C.J.; Richardson, C.L. A time-discrete model for dynamic fracture based on crack regularization. *Int. J. Fract.* **2011**, *168*, 133–143. [\[CrossRef\]](#)
- Schlüter, A.; Willenbücher, A.; Kuhn, C.; Müller, R. Phase field approximation of dynamic brittle fracture. *Comput. Mech.* **2014**, *54*, 1141–1161. [\[CrossRef\]](#)
- Borden, M.J.; Verhoosel, C.V.; Scott, M.A.; Hughes, T.J.; Landis, C.M. A phase-field description of dynamic brittle fracture. *Comput. Methods Appl. Mech. Eng.* **2012**, *217–220*, 77–95. [\[CrossRef\]](#)
- Nguyen, V.P.; Wu, J. Modelling dynamic fracture of solids with a phase-field regularized cohesive zone model. *Comput. Methods Appl. Mech. Eng.* **2018**, *340*, 1000–1022. [\[CrossRef\]](#)
- Lee, S.; Wheeler, M.F.; Wick, T. Pressure and fluid-driven fracture propagation in porous media using an adaptive finite element phase-field model. *Comput. Methods Appl. Mech. Eng.* **2016**, *305*, 111–132. [\[CrossRef\]](#)

18. Miehe, C.; Mauthe, S. Phase field modelling of fracture in multi-physics problems. Part III. Crack driving forces in hydro-poro-elasticity and hydraulic fracturing of fluid-saturated porous media. *Comput. Methods Appl. Mech. Eng.* **2016**, *304*, 619–655. [\[CrossRef\]](#)
19. Badnava, H.; Msekh, M.A.; Etemadi, E.; Rabczuk, T. An h-adaptive thermo-mechanical phase-field model for fracture. *Finite Elem. Anal. Des.* **2018**, *138*, 31–47. [\[CrossRef\]](#)
20. Zhou, S.; Zhuang, X.; Rabczuk, T. A phase-field modelling approach of fracture propagation in poroelastic media. *Eng. Geol.* **2018**, *240*, 189–203. [\[CrossRef\]](#)
21. Bourdin, B.; Francfort, G.; Marigo, J.J. *The Variational Approach to Fracture*; Springer: Berlin, Germany, 2008.
22. Ambati, M.; Gerasimov, T.; de Lorenzis, L. A review on phase-field models for brittle fracture and a new fast hybrid formulation. *Comput. Mech.* **2015**, *55*, 383–405. [\[CrossRef\]](#)
23. Wu, J.Y.; Nguyen, V.P.; Nguyen, C.T.; Sutulas, D.; Sinaie, S.; Bordas, S.P.A. Phase field model for fracture. *Adv. Appl. Mech. Fract. Mech. Recent Dev. Trends.* **2019**, submitted for publication.
24. Sirovich, L. Turbulence and the dynamics of coherrent structures. Part 1: Coherent structures. *Q. Appl. Math.* **1987**, *45*, 561–571. [\[CrossRef\]](#)
25. Holmes, P.; Lumley, J.; Berkooz, G. *Tubulence, Cohenrent Structures, Dynamical Systems and Symmetry*; Cambridge University Press: Cambridge, UK, 1996.
26. Antoulas, A.C.; Sorensen, D.C.; Gugercin, S. A survey of model reduction methods for large-scale systems. *Contemp. Math.* **2001**, *280*, 193–219.
27. Haasdonk, B.; Ohlberger, M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math. Model. Numer. Anal.* **2008**, *42*, 277–302. [\[CrossRef\]](#)
28. Astrid, P.; Weiland, S.; Willcox, K.; Backx, T. Missing Point Estimation in Models Described by Proper Orthogonal Decomposition. *IEEE Trans. Autom. Control* **2008**, *53*, 2237–2251. [\[CrossRef\]](#)
29. Carlberg, K.; Farhat, C.; Cortial, J.; Amsallem, D. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.* **2013**, *242*, 623–647. [\[CrossRef\]](#)
30. Barrault, M.; Maday, Y.; Nguyen, N.C.; Patera, A.T. An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Math.* **2004**, *339*, 667–672. [\[CrossRef\]](#)
31. Chaturantabut, S.; Sorensen, D.C. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764. [\[CrossRef\]](#)
32. Radermacher, A.; Reese, S. POD-based model reduction with empirical interpolation applied to nonlinear elasticity. *Int. J. Numer. Methods Eng.* **2016**, *107*, 477–495. [\[CrossRef\]](#)
33. Benner, P.; Gugercin, S.; Willcox, K. *A Survey of Model Reduction Methods for Parametric Systems*; Preprint MPIMD/13–14; Max Planck Institute Magdeburg: Magdeburg, Germany, 2013.
34. Wirtz, D.; Sorensen, D.C.; Haasdonk, B. A Posteriori Error Estimation for DEIM Reduced Nonlinear Dynamical Systems. *SIAM J. Sci. Comput.* **2014**, *36*, A311–A338. [\[CrossRef\]](#)
35. Negri, F.; Manzoni, A.; Amsallem, D. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comput. Phys.* **2015**, *303*, 431–454. [\[CrossRef\]](#)
36. Amsallem, D.; Cortial, J.; Carlberg, K.; Farhat, C. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Methods Eng.* **2009**, *80*, 1241–1258. [\[CrossRef\]](#)
37. Panzer, H.; Mohring, J.; Eid, R.; Lohmann, B. Parametric Model Order Reduction by Matrix Interpolation. *at-Automatisierungstechnik* **2010**, *58*, 475–484. [\[CrossRef\]](#)
38. Degroote, J.; Vierendeels, J.; Willcox, K. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int. J. Numer. Methods Fluids* **2010**, *63*, 207–230. [\[CrossRef\]](#)
39. Amsallem, D.; Farhat, C. An Online Method for Interpolating Linear Parametric Reduced-Order Models. *SIAM J. Sci. Comput.* **2011**, *33*, 2169–2198. [\[CrossRef\]](#)
40. Nguyen, N.H.; Le, T.H.H.; Khoo, B.C. Model reduction for parametric and nonlinear PDEs N by matrix interpolation. In Proceedings of the 2015 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 14–16 October 2015; pp. 105–110. [\[CrossRef\]](#)
41. Nguyen, N.C.; Patera, A.T.; Peraire, J. A ‘best points’ interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Methods Eng.* **2008**, *73*, 521–543. [\[CrossRef\]](#)

42. Nguyen, N.H. A rapid simulation of nano-particle transport in a two-dimensional human airway using POD/Galerkin reduced-order models. *Int. J. Numer. Methods Eng.* **2016**, *105*, 514–531. [[CrossRef](#)]
43. Rixen, D.J. A dual Craig–Bampton method for dynamic substructuring. *J. Comput. Appl. Math.* **2004**, *168*, 383–391. [[CrossRef](#)]
44. Markovic, D.; Park, K.; Ibrahimbegovic, A. Reduction of substructural interface degrees of freedom in flexibility-based component mode synthesis. *Int. J. Numer. Methods Eng.* **2007**, *70*, 163–180. [[CrossRef](#)]
45. Tiso, P.; Rixen, D.J. Discrete empirical interpolation method for finite element structural dynamics. In *Topics in Nonlinear Dynamics, Volume 1*; Springer: Berlin, Germany, 2013; pp. 203–212.
46. Galland, F.; Gravouil, A.; Malvesin, E.; Rochette, M. A global model reduction approach for 3D fatigue crack growth with confined plasticity. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 699–716. [[CrossRef](#)]
47. Ghavamian, F.; Tiso, P.; Simone, A. POD–DEIM model order reduction for strain-softening viscoplasticity. *Comput. Methods Appl. Mech. Eng.* **2017**, *317*, 458–479. [[CrossRef](#)]
48. Kerfriden, P.; Passieux, J.C.; Bordas, S.P.A. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. *Int. J. Numer. Methods Eng.* **2012**, *89*, 154–179. [[CrossRef](#)]
49. Kerfriden, P.; Goury, O.; Rabczuk, T.; Bordas, S.P.A. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Comput. Methods Appl. Mech. Eng.* **2013**, *256*, 169–188. [[CrossRef](#)] [[PubMed](#)]
50. Goury, O.; Amsallem, D.; Bordas, S.P.A.; Liu, W.K.; Kerfriden, P. Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: From dissipation-driven random selection to Bayesian optimization. *Comput. Mech.* **2016**, *58*, 213–234. [[CrossRef](#)]
51. Oliver, J.; Caicedo, M.; Huespe, A.E.; Hernández, J.; Roubin, E. Reduced order modelling strategies for computational multiscale fracture. *Comput. Methods Appl. Mech. Eng.* **2017**, *313*, 560–595. [[CrossRef](#)]
52. Xia, L.; Da, D.; Yvonnet, J. Topology optimization for maximizing the fracture resistance of quasi-brittle composites. *Comput. Methods Appl. Mech. Eng.* **2018**, *332*, 234–254. [[CrossRef](#)]
53. Krige, D.G. A Statistical Approach to Some Mine Valuations and Allied Problems at the Witwatersrand. Master’s Thesis, The University of Witwatersrand, Johannesburg, South Africa, 1951.
54. Cressie, N.A.C. The Origins of Kriging. *Math. Geol.* **1990**, *22*, 239–252. [[CrossRef](#)]
55. Pham, K.; Amor, H.; Marigo, J.J.; Maurini, C. Gradient damage models and their use to approximate brittle fracture. *Int. J. Damage Mech.* **2011**, *20*, 618–652. [[CrossRef](#)]
56. Peerlings, R.; de Borst, R.; Brekelmans, W.; de Vree, J. Gradient-enhanced damage for quasi-brittle materials. *Int. J. Numer. Methods Engng.* **1996**, *39*, 3391–3403. [[CrossRef](#)]
57. Wu, J.Y. Robust numerical implementation of non-standard phase-field damage models for failure in solids. *Comput. Methods Appl. Mech. Eng.* **2018**, *340*, 767–797. [[CrossRef](#)]
58. Amor, H.; Marigo, J.; Maurini, C. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. *J. Mech. Phys. Solids* **2009**, *57*, 1209–1229. [[CrossRef](#)]
59. Ryckelynck, D. Hyper-reduction of mechanical models involving internal variables. *Int. J. Numer. Methods Eng.* **2009**, *77*, 75–89. [[CrossRef](#)]
60. Sacks, J.; Welch, W.J.; Mitchell, T.J.; Wynn, H.P. Design and Analysis of Computer Experiments. *Stat. Sci.* **1989**, *4*, 409–435. [[CrossRef](#)]
61. Lophaven, S.N.; Nielsen, H.B.; Sondergaard, J. *Aspects of the MATLAB Toolbox DACE*; Technical Report IMM-REP-2002-13; Technical University of Denmark: Lyngby, Denmark, 2002.
62. McKay, M.D.; Beckman, R.J.; Conover, W.J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **1979**, *21*, 239–245.

