DATA-DRIVEN MULTI-SCALE MODELING, ANALYSIS AND SIMULATION OF MATERIAL FAILURE

By

Eduardo Augusto Barros de Moraes

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy
Computational Mathematics, Science and Engineering – Dual Major

2022

ABSTRACT

DATA-DRIVEN MULTI-SCALE MODELING, ANALYSIS AND SIMULATION OF MATERIAL FAILURE

By

Eduardo Augusto Barros de Moraes

Material failure processes are inherently stochastic and anomalous, occurring across a wide span of length and time scales, from dislocation motion at the micro-scale, to formation of micro-cracks, up to crack propagation and aging mechanisms at the macro-scale and cascading failure at the system-level. Anomalies such as intermittent signals in Acoustic Energy experiments, power-law distribution of the energy spectrum, crackling noise, dislocation avalanches, among other indicators, occur even in standard, ordered, crystalline materials. Modeling and simulation of failure must take into account parametric and model-form uncertainties that propagate across the scales, when seemingly unimportant material properties or loading conditions could cause catastrophic failure at the component level. The pursuit of a unified framework for quantitative and qualitative failure prediction that can bridge the multiple scales while still incorporating the material's underlying stochastic processes is still a challenge, which requires a new modeling paradigm that incorporates such features with both robustness and simplicity.

In this work, we propose a data-driven methodology for multi-scale, statistically consistent modeling of anomalous failure processes. At the micro-scale, the goal is to study the dynamics of dislocations, which play a vital role in plasticity and crack nucleation mechanisms, and shows anomalous features across different time and length-scales. We start by investigating the dislocation mobility properties at the nano-scale and propose a surrogate model for dislocation motion based on a Kinetic Monte Carlo method, where the dislocation motion is emulated as a random-walk on a network following a Poisson process. The surrogate learns the rates of the corresponding Poisson process directly from high-fidelity, Molecular Dynamics (MD) simulations. The surrogate is capable of efficiently obtaining uncertainty measures for the mobility parameter, which can be then propagated to more complex simulations in upper scales. At the meso-scale, the collective behavior of dislocation dynamics leads to avalanche, strain bursts, intermittent energy spikes, and

nonlocal interactions. We develop a probabilistic model for dislocation motion constructed directly from trajectory data from Discrete Dislocation Dynamics (DDD). We obtain the corresponding Probability Density Function for the dislocation position, and propose a nonlocal transport model for the PDF. We use a bi-level Machine Learning framework to learn the parameters of the nonlocal operator and the coefficients of the PDF evolution equation, facilitating a continuum representation of the anomalous phenomena.

At the macro-scale, parametric material uncertainties substantially affect the predictability of failure at the component level. We develop an Uncertainty Quantification (UQ) and Sensitivity Analysis (SA) framework for propagation of parametric uncertainties in a stochastic phase-field model of damage and fatigue, and we use the Probabilistic Collocation Method (PCM) as a building block. A Global SA indicates the most influential parameters in solution uncertainty and shows that damage initiation is sensitive to parameters associated with classical free-energy potential definitions, providing another motivation to incorporate the heavy-tail processes as observed in the meso-scale. We extend the framework and develop a Machine Learning (ML) framework for failure prediction phase-field models for brittle materials. We combine a classification algorithm with a pattern recognition scheme using virtual nodes from the phase-field damage model to generate patterns of material softening at each time-step. The framework identifies the presence and location of cracks and is robust even under noisy data, whether from model, parametric, or experimental uncertainties.

I dedicate this work to my wife, Julia, and my children, Lucas and Olivia.

ACKNOWLEDGEMENTS

I would like to thank Dr. Mohsen Zayernouri, my PhD advisor, for the wonderful opportunity to join FMATH group and engage in such interesting and productive research. I am grateful for all the discussions, exchange of ideas, and thorough help during this journey. I extend my gratitude to the Committee members, Dr. Shanker Balasubramaniam, Dr. Tony Gao, Dr. Sara Roccabianca, Dr. Hui-Chia Yu for the valuable feedback and support for the completion of this PhD project.

I would like to thank the external collaborators that worked with me the past five years and helped this project make substantial advances, Dr. Mark Meerschaert, Dr. Hadi Salehi, and Dr. Marta D'Elia.

I would also like to thank all the current and former members of FMATH group with whom I had the privilege to share this amazing experience of being a Spartan. Thank you Dr. Ehzan Kharazmi, Dr. Mehdi Samiee, Dr. Jorge Suzuki, Pegah Varghaei, Ali Akhvan-Safaei, Hadi Seyedi, and Dr. Yongtao Zhou for the collaborations, discussions, and help through the years.

I would like to send a special thanks to my family and friends. To my parents, Ângela and Eduardo, and my sister Isabela, thank you for believing in me since the beginning and for giving me all the love and support that helped me reach this stage. To my in-laws Vicente and Eliane, and my sister-in-law Bruna, thank you for your support and trust you had in me through all the years.

This work could not be possible without the incredible support of my wife, Julia. Thank you for staying by my side through the most challenging times of our lives, thank you for caring and providing all the support that our family needs, and for giving me the strength to keep improving and going after new challenges. To my children, Lucas and Olivia, thank you for all your patience you showed in the most difficult times, the kindness and pure joy you give to those around you every day. May this work and all our efforts be a door for a lifetime of happiness that you deserve.

This work was supported by the department of Mechanical Engineering and the department of Computational Mathematics, Science and Engineering at Michigan State University. The high-performance computational resources were provided by the Institute for Cyber-Enabled Research

(ICER) at Michigan State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Failure as a Stochastic Anomalous Process

The initial works on fracture mechanics by Griffith [2] laid the ground of many years of development of failure analysis through the theory of Linear Elastic Fracture Mechanics (LEFM). Although descriptive, Griffith's theory relied heavily in the assumption of homogeneous materials which, consequently, originated smooth and continuous characteristics of propagating cracks [3]. However, in reality failure processes are not deterministic and fracture surfaces are naturally rough, thus limiting the extent of failure prediction based solely on LEFM models. Over the last three decades, a solid understanding of the nature of failure in heterogeneous media (a more realistic assumption for material models) has been developed and matured with the aid of fractography analysis and Acoustic Emission (AE) techniques.

1.1.1 Fractography Analysis

The onset of quantitative studies of failure in heterogeneous and disordered media can be traced back to Mandelbrot [4], where the analysis of fractured surfaces revealed an inherent fractal dimension. The fractal dimension in an index that characterizes the complexity of patterns by measuring the ratio of change in detail due to the change in scale. In their study, Mandelbrot et. al. showed that the same fracture dimension increments are found from computing the fractal dimension of fracture boundaries and surfaces.

Further fractography studies corroborated quantitatively the fractal characteristics of fracture surfaces [5–10]. In [5], experimental computation of fractal dimension of brittle fracture surfaces showed a direct correspondence between fractal dimension and toughness. Universality of fracture dimension of fractured surfaces was investigated in [6] (Fig. 1.1). After applying different heat treatments to aluminum alloys that lead to distinct fracture characteristics and fracture toughness,

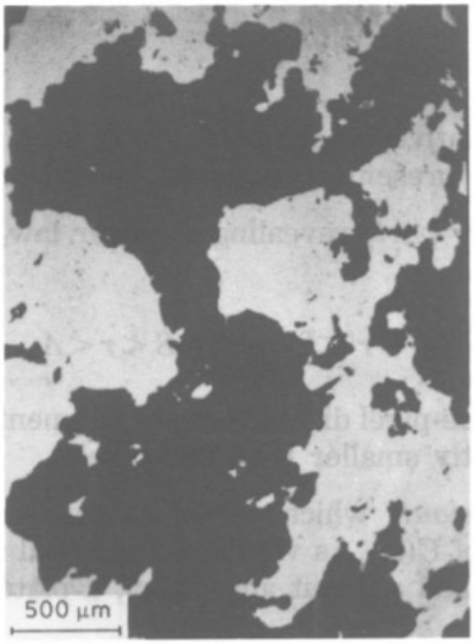


Figure 1.1 Micrograph of a fractured surface highlighting the different phases of aluminium and nickel, used in fractography analysis to compute the fractal dimension.

they showed within experimental error that fractal dimension was independent of the heat treatment or fracture mode.

1.1.2 Acoustic Emission

During an Acoustic Emission (AE) test, irreversible changes in the material's microstructure undergoing a failure process cause the emission of acoustic (elastic) waves that can be recorded through sensors. The use of AE in fracture experiments showed interesting phenomena that could not be accounted for in classical LEFM models, indicating an anomalous process was taking place inside the material. For instance, intermittent energy signals, power-law distribution of energy spectrum, crackling noise, can all be observed during an AE test, all of which are indicators that avalanches and anomalous dynamics are occurring. Such anomalous statistics can be traced back to two main contributors: dislocation motion, and fracture.

Evidence of dislocation avalanches in crystal plasticity has been widely studied through AE experiments [11–15]. In [11], creep experiments in single ice crystals led to intermittent energy

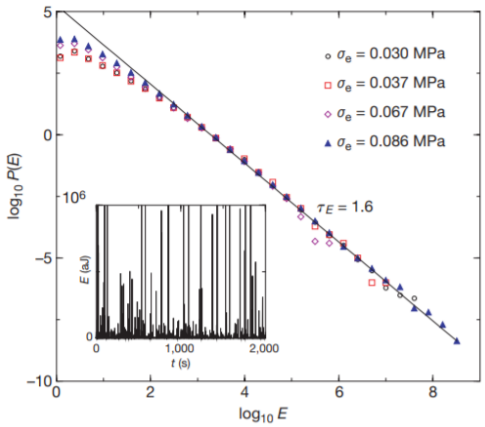


Figure 1.2 Acoustic emission signals from single crystal ice under creep. The energy bursts caused by the collective dislocation motion appear in the form of intermittent signals that scale as a power-law of exponent $\tau = 1.6$, independent of the applied stress.

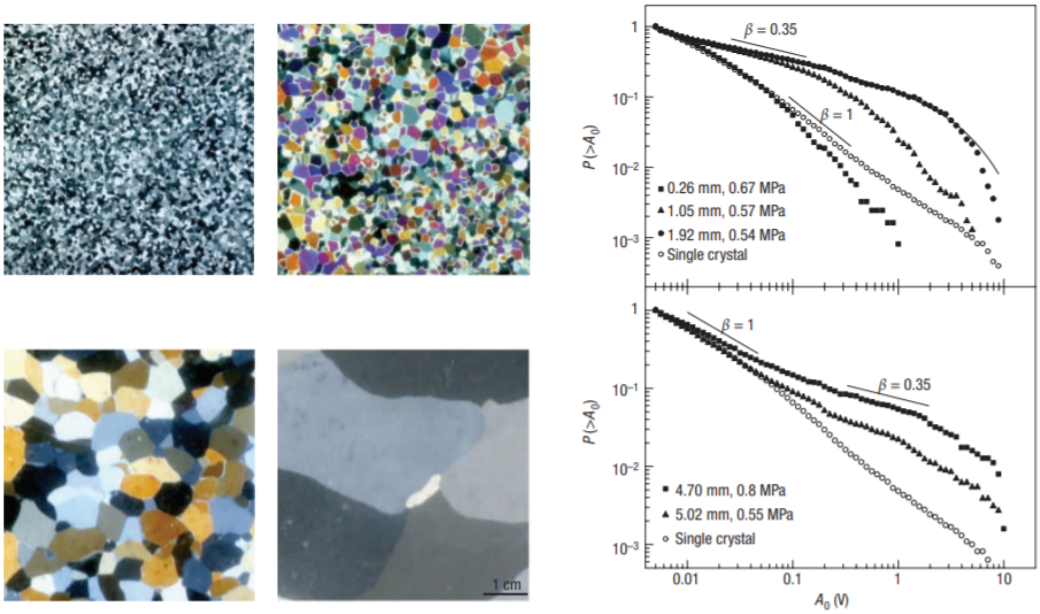
signals that follow a power-law scaling due to the collective motion of dislocation populations (Fig. 1.2). They compared the statistical results from a numerical model of dislocation dynamics with the experimental acoustic data. Similar power-law decay of dislocation activity has been observed through phase-field models of dislocation dynamics [12].

Dislocation avalanches in polycrystalline materials was studied extensively in [13]. They show that avalanches in intermittency is also present in polycrystals, where avalanche sizes are bounded by the grain sizes (Fig 1.3). Yet, they propose that the accumulation of internal stresses near the grain boundaries due to dislocation activity triggers further avalanches in neighboring grains.

From a different perspective, AE tests also showed anomalous features in brittle fracture [16–20]. encourages the search for a unified model that can effectively incorporate the heavy-tail statistics into a predictive model. Ref. [16] showed that energy bursts from AE tests of microfracture follow a power-law scaling, corroborating the observations also made in the time correlation function.

Bonamy et. al. [18] studied crack growth in heterogeneous materials through a stochastic LEFM model to account for the material heterogeneity. They have observed that crack propagation velocity is intermittent, with activity burst duration and size scaling as power-laws (Fig. 1.4). The jerky dynamics observed in the simulations are referred to as the *crackling noise* from experiments.

A random-fuse model was used in [19] to study the transition of percolation, to avalanche,



(a) (b)

Figure 1.3 (a) Polycrystalline microstructure with different grain sizes, averages of $\langle d \rangle = 0.26\ mm$ (top-left), 0.87 *mm* (top-right), 1.92 *mm* (bottom-left), and 5.02 *mm* (bottom-right). (b) Distribution of avalanche size from dislocation motion in polycrystals for crystals with different grain sizes. A tempered power-law fit of the form $P(> A_0) = A_0^{-\beta} \exp(-A_0/A_c)$ is used to estimate the cut-off amplitude A_c. The power-law exponent across all samples is similar. At the bottom, the relation for coarse-grained samples.

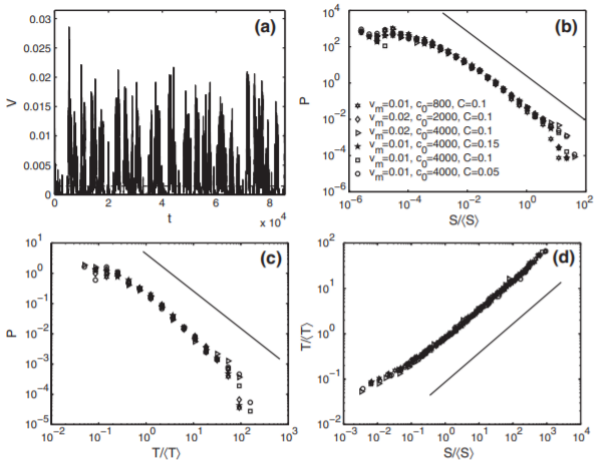


Figure 1.4 Time-series of spatially averaged crack-front velocity (a), and the respective scalings of burst duration distribution (b), size distribution (c), and their mutual scaling (d).

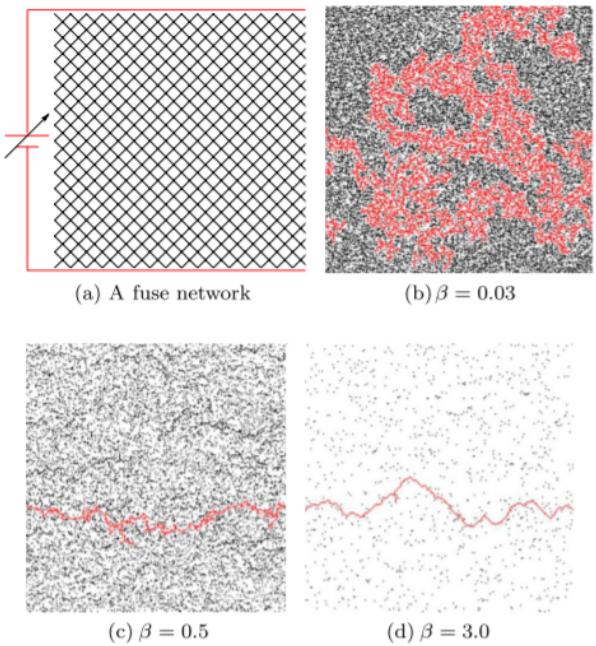
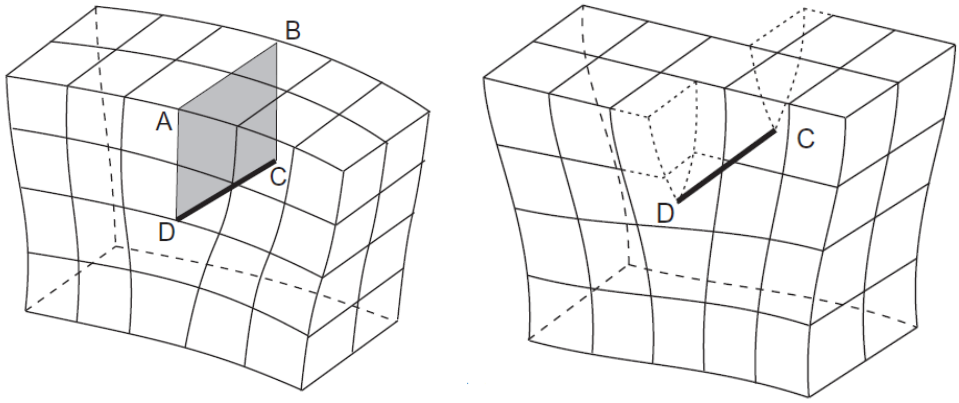


Figure 1.5 Random-fuse model (a), and the fractured samples under different values of β, for percolated, disordered media (b), critical dynamics (c), and finally leading to nucleated cracks for large β (d).

to brittle failure in disordered media. The fuse model was parameterized by specifying failure resistance following a power-law probability $F(x) = x^{-\beta}$, where β is the disorder parameter. The systems transitions from completely disordered (small β) subject to percolation, to a critical avalanche regime, up to a brittle failure region under low disorder (high β) and larger system size (Fig. 1.5). They point that critical phenomena leads to nucleation in the limit of long length scales.

1.2 Multi-scale Modeling of Material Failure

Multi-scale modeling of failure and fracture has been a topic of interest for many decades. Several complementary methodologies have been presented for the coupling between the scales, given the wide range of time and length scales involved during the failure process. A major difficulty is to properly address the stochastic nature of failure and to account for anomalous phenomena, such as nonlocality, in the upper scales without losing the physics from the microstructure. We briefly describe major advances in the literature on the multi-scale modeling of material failure, starting from the basic definitions of dislocation theory.



(a) Edge dislocation. (b) Screw dislocation.

Figure 1.6 An edge dislocation characterized by an extra half-plane of atoms (a), and a screw dislocation obtained by a twisted displacement (b) [1].

1.2.1 Fundamentals of Dislocation Theory

We present some definitions based on [1]. There are two canonical types of dislocations:

- **Edge dislocations**: can be seen as a half-plane of atoms being removed from the crystalline structure, displacing the atoms along the missing plane. This creates a linear defect, called an edge dislocation, with stronger deflection and stresses being closer to the dislocation line.
- **Screw dislocations**: screw dislocations are obtained by sliding the two halves of the crystal in opposite directions with respect to the center.

Fig. 1.6 illustrates the concepts of edge and screw dislocations. In practice, dislocations inside crystalline materials have mixed character between the edge and screw types.

An important characterization of dislocations are in the form of the Burgers vector, $\boldsymbol{b}$. The Burgers circuit in a crystal is an atom-to-atom path forming a closed loop. However, around dislocations the same path will not close the loop, and the $\boldsymbol{b}$ characterizes the missing vector needed to complete the Burgers circuit around the dislocation. The magnitude b of burgers vectors in cubic crystals is measured with respect to the atomic spacing a, such that

$$b = \frac{a\sqrt{3}}{2}. \tag{1.1}$$

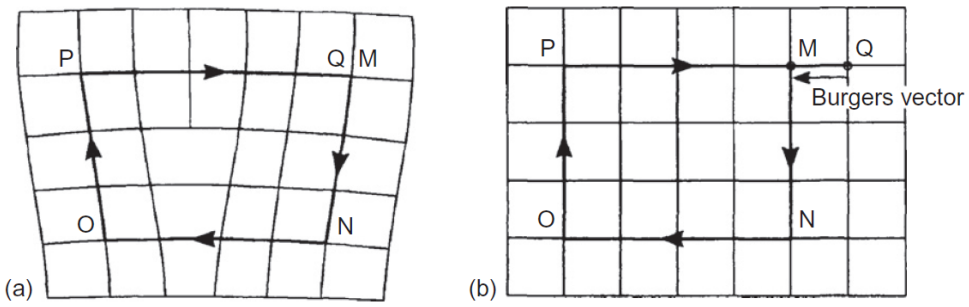


Figure 1.7 Example of construction of Burgers circuit, and Burgers vector definition for an edge dislocation [1].

Dislocations must always form a close loop, end at a free surface, or merge with other dislocations, but never end inside the crystal. This condition leads to Burgers vector conservation such that $\sum_i \boldsymbol{b}_i = 0$. The Burgers vector of an edge dislocation is illustrated in Fig. 1.7. For edge dislocations, the Burgers vectors is perpendicular to the dislocation line tangent vector, while they are parallel in screw dislocations.

Finally, the forces acting on a dislocation line with direction ξ tangent to the dislocation, under a stress field $\boldsymbol{\sigma}$ is called a Peach-Koehler force $\boldsymbol{f}^{PK}$, given by the expression

$$\boldsymbol{f}^{PK} = (\boldsymbol{\sigma} \cdot \boldsymbol{b}) \times \xi. \tag{1.2}$$

1.2.2 From Nano to Meso-scale

At the nano-scale, atomistic simulations are usually carried out through Molecular Dynamics (MD) simulations. Specifically concerning failure problems, MD simulations have been used to simulate dislocation motion and interaction, and fracture problems with dislocation emission. Three-dimensional MD simulations of Mode-I crack propagataion was performed in [21], providing visual evidence of dislocation emission from the crack tip. Recent advances in the literature provided ways to understand the dislocation behavior at the atomistic scale and connect it to more complex dislocation interactions. In [22], they studied formation of dislocation junctions and quantified the junction stress to feed a meso-scale dislocation dynamics model. Further works on atomistic and meso-scale coupling were developed to quantify the elasticity parameters, mobility,

and core energy of dislocations [23]. The issue of quantifying the uncertainties from measures at the atomistic scale has recently been getting attention [24–26], but the problem of effectively upscaling material properties and propagating the associated uncertainties remains a challenge.

1.2.3 From Meso to Macro-scale

At the meso-scale, collective dislocation motion gives origin to anomalous features in the form of dislocation avalanches and power-law energy bursts. Discrete Dislocation Dynamics (DDD) [11, 27] models have successfully captured many anomalous characteristics at the meso-scale, while a meaningful connection of dislocation dynamics to macro-scale mechanical failure is still missing. Recent attempts have used machine learning to provide a connection from micro-structure to the continuum through the computation of free-energy [28–30].

At the macro-scale, state-of-the-art phase-field models efficiently simulate qualitative aspects of brittle [31], ductile [32], dynamic [33], and fatigue [34] failure in complex geometries, and under arbitrary loading conditions. However, such models fail to incorporate the anomalies associated with failure processes. On the other hand, the statistics of fracture can be quantitatively recovered in lattice models, such as Random Fuse Models (RFM) [35], percolation theory [36] and stochastic LEFM [3]. Despite the quantitative agreement, those models often are limited to simple geometries and boundary conditions. The pursuit of a unified framework for quantitative and qualitative failure prediction is a remaining challenge.

We now discuss in more detail three components that form a major part of the proposed framework when considering predictive failure modeling and coupling between the meso- and macro-scales. In the context of modeling of collective dislocation dynamics, nonlocality is a dominant effect observed experimentally, yet not studied under formal nonlocal calculus definitions. Therefore, we comment on basic principles of nonlocal calculus and its applications to nonlocal differential equations. Moving to the topic of damage modeling, we discuss some key aspects of phase-field models and why they are attractive. Last, we comment on methodologies to perform forward uncertainty propagation that will later be applied in all major components of the multi-scale

framework.

1.2.4 Nonlocal Models

Nonlocal operators are an elegant alternative for continuum modeling of complex media, such as heterogeneous materials, anomalous diffusion, and fracture, due to their ability to capture long-range interactions. In contrast to their local counterpart, nonlocal models are based on integral operators, leading to challenging numerical implementation. Recent works greatly advanced the field of nonlocal modeling towards a formal and rigorous nonlocal vector calculus [37, 38].

We can formulate a general nonlocal problem as follows. Let $\Omega \subset \mathbb{R}^n$ denote a bounded domain with non-zero volume. The nonlocal diffusion operator $\mathcal{L}$ acting on a function $u(\boldsymbol{x}) : \Omega \to \mathbb{R}$ is defined as [39]:

$$\mathcal{L}u(\boldsymbol{x}) = 2\int_{\mathbb{R}^n} (u(\boldsymbol{y}) - u(\boldsymbol{x}))\gamma(\boldsymbol{x}, \boldsymbol{y})d\boldsymbol{y} \quad \forall \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^n.$$

The kernel $\gamma(\boldsymbol{x}, \boldsymbol{y})$ is a non-negative symmetric mapping. The general nonlocal diffusion equation is then written as

$$\begin{cases} \frac{du}{dt} - \mathcal{L}u = f & \text{on} \quad \Omega \\ u = u_d & \text{on} \quad \Omega_l \\ u(0) = u_0 & \text{on} \quad \Omega, \end{cases}$$

where Ω_l denotes an interaction domain disjoint from Ω, parameterized by a horizon of interactions of length δ.

Under specific choices of kernel and horizon, the general nonlocal operator may take the form of the fractional Laplacian. The fractional Laplacian of order s on a bounded domain Ω is defined through Riesz representation as

$$(-\Delta)^s u = c_{n,s}\int_{\mathbb{R}^n} \frac{u(\boldsymbol{x}) - u(\boldsymbol{y})}{|\boldsymbol{y} - \boldsymbol{x}|^{n+2s}}, \quad 0 < s < 1,$$

where $c_{n,s}$ is a normalizing constant such that

$$c_{n,s} = s2^{2s}\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{1}{2})\Gamma(1-s)}$$

If the kernel $\gamma(\boldsymbol{x}, \boldsymbol{y})$ of the nonlocal diffusion operator is chosen to be [39]

$$\gamma(\boldsymbol{x}, \boldsymbol{y}) = \frac{c_{n,s}}{2|\boldsymbol{y} - \boldsymbol{x}|^{n+2s}}, \tag{1.3}$$

then

$$-\mathcal{L} = (-\Delta)^s, \quad 0 < s < 1,$$

Nonlocal and fractional models have been successfully used in material failure simulation over the last decades. Peridynamic (PD) theory [40] has been introduced as an alternative to classical continuum mechanics modeling of solid structures and failure. Due to the nonlocal character of PD, the problem of discontinuities in crack representation was overcome, and PD has successfully represented many classes of material failure problems, including visco-plasticity [41], fatigue [42], composites [43], polymers [44], multi-physics problems [45], among many others. Fractional models have been used extensively in modeling visco-elastic and visco-elasto-plastic material responses [46, 47].

With the advance of Machine Learning techniques and computational resources, the field of nonlocal modeling has seen increasing efforts in the parameter identification of nonlocal operators, specially in the data-driven construction of the nonlocal kernels which cannot be defined beforehand. In [48], nonlocal Physics Informed Neural Networks, nPINNs (the nonlocal version of PINNs [49]) were used to solve the forward nonlocal Poisson problem, and the inverse parameter identification of the fractional order and horizon of the nonlocal kernel given available data. This approach combines the robustness of Deep Neural Networks with physics-based constraints from the nonlocal problem. Other works employed alternative approaches for learning nonlocal kernels, with notable examples on peridynamics [50], homogenization [51], constitutive laws [52], and molecular dynamics [53].

1.2.5 Phase-field Models

Phase-field models were first developed to solve fluid separation problems [54]. The success of Cahn-Hilliard equation in interfacial dynamics allowed for its expansion into other research areas

that model sharp interface and moving boundaries, including solidification [55], tumor growth [56], two-phase complex fluid flow [57] and fluid-structure interaction [58].

The core idea of phase-field modeling is to define the phase variable (sometimes called order parameter) η to be a smoothly-varying indicator function of any particular field quantity of interest, such as concentration, damage, composition, or a classical phase definition, as observed in phase transformation problems. Evolution equations for the phase-field variable are usually derived following variational principles that promote thermodynamic consistency through the definition of free-energy potentials.

In general form, the free-energy function for a phase-field problem may read:

$$F[\varphi, \Gamma] = \int_{\Omega} K|\nabla\varphi|^2 + f(\varphi, \Gamma), \tag{1.4}$$

where φ denotes the phase-field variable, and Γ represents any other variables that influence the free-energy. We denote K as the diffusion constant, and $f(\varphi, \Gamma)$ is a function that describes the evolution of free-energy in the bulk of each material phase. The interpretation of Eq. (1.4) is a competition between the bulk energy from f and a mixing energy from the gradient term.

The most famous phase-field models, the Cahn-Hilliard [54] and Allen-Cahn [55] equations, follow the variational principles described above to obtain evolution equations for phase variables. Both equations use a double-well potential for f, while major difference between the two is that Cahn-Hilliard equation is conservative and Allen-Cahn is non-conservative. As a consequence, Cahn-Hilliard equation is fourth-order

$$\frac{d\varphi(x,t)}{dt} = \nabla \cdot (M(x,t)\nabla\mu(x,t)), \tag{1.5}$$

where $\mu(x,t)$ denotes the chemical potential

$$\mu(x,t) = f'(\varphi, \Gamma) - \epsilon^2\Delta\varphi(x,t). \tag{1.6}$$

On the other hand, Allen-Cahn equation is second-order

$$\frac{d\varphi(x,t)}{dt} = \nabla \cdot (M(x,t)\nabla\varphi(x,t)) + f'(\varphi, \Gamma). \tag{1.7}$$

The non-conservative characteristics of Allen-Cahn equation are specially interesting in the development of damage models, since damage itself is non-conservative and appears in virgin portions of bulk materials as they undergo mechanical stresses and strains. The development of Allen-Cahn type phase-field equations for evolution of material damage has gained much attention recently, with notable applications in [31], ductile [32], and dynamic [33] fracture. Although they originated from crack regularization techniques, they obey the same variational principles used to derive the original phase-field equations, such as [34]. The Allen-Cahn equation for damage evolution in this case, in one dimension, is originated from the free-energy functional

$$\Psi(\nabla u, \varphi, \nabla\varphi, \mathcal{F}) = d(\varphi)Y\,(\nabla u)^2 + g_c\frac{\gamma}{2}\,(\nabla\varphi)^2 + \mathcal{K}(\varphi, \mathcal{F}), \tag{1.8}$$

where Y denotes the Young modulus, u represents the displacements, g_c is the fracture energy release rate, and $\gamma > 0$ represents the phase-field layer width parameter, $d(\varphi)$ is a degradation function acting on the bulk of the material, and $\mathcal{K}(\varphi, \mathcal{F})$ models the evolution of material damage when fatigue $\mathcal{F}$ evolves from 0 to the Griffith energy g_c. The thermodynamically consistent formulation therefore provides the evolution equation as

$$\lambda\frac{d\varphi}{dt} = \gamma g_c\Delta\varphi + (1-\varphi)(\nabla u)^T Y(\nabla u) - \frac{1}{\gamma}[g_c\mathcal{H}'(\varphi) + \mathcal{F}\mathcal{H}_f'(\varphi)], \tag{1.9}$$

where $\mathcal{H}$ and $\mathcal{H}_f$ are potentials that govern the evolution of damage associated to the evolution of material fatigue, and λ is a positive coefficient. We see the presence of the interfacial term in the form of Laplacian acting as a smoothening agent to the damage profile, and the bulk term responsible for the internal creation of material damage from mechanical stress and fatigue. The main advantage of this formulation in contrast with traditional fracture mechanics modeling is the automatic crack profile characterization by the phase-field variable, without the need to explicitly track the crack path.

1.2.6 Uncertainty Quantification

The goal of Uncertainty Quantification (UQ) in materials science is to study the propagation of uncertainties from inputs to outputs or vice-versa. The former is known as Forward UQ, while

the latter is named Inverse UQ, concerned with quantifying the probability distribution associated to parameter values when one has output data. Forward UQ aims to propagate uncertainty from inputs to outputs, with the objective of computing the moments (mean, variance), of quantities of interest (QoI), their probability distribution, or the reliability of QoI predictions.

There are two basic types of uncertainty: aleatoric and epistemic [59]. Aleatoric uncertainty concern stochastic or probabilistic uncertainties that are typical to the underlying problem and which cannot be further understood by physical or experimental knowledge, yet it could benefit from extra characterization, such as in the case of non-physical parameters. Aleatoric uncertainty can naturally incorporated into a framework. Epistemic uncertainty, on the other hand, originate from simplifying assumptions or missing physics. Modeling and numerical errors are typically understood as being epistemic.

For forward input uncertainty quantification several methods could be applied. The Monte Carlo method consists in computing the moments of the output through random sampling of the stochastic space. It provides an unbiased estimation of the moments of distribution, yet they converge with order $O(N^{-1/2})$, where N is the number of realizations. Perturbation methods are also popular in forward UQ, consisting in writing expressions for the moments of output distribution through a Taylor expansion around the expected parameter values [60, 61]. Alternatively, one could construct surrogate models as an alternative to expensive methods such as Monte Carlo. The method of Polynomial Chaos Expansion (PCE) [62, 63] consists in expressing the output of the stochastic model as a series expansion of input parameters.

More promising methods for higher-dimensional stochastic space have been proposed, such as stochastic Galerkin methods [64, 65]. However, one of the most efficient approaches proposed recently involve non-intrusive collocation methods, such as the Probabilistic Collocation Method (PCM) [66, 67]. PCM consists in approximating the solution in the stochastic space by polynomial expansion, such that the moments of QoI can be efficiently computed by sampling in the collocation/integration points. Therefore, one keeps the independent sampling typical of methods such as MC, yet with much faster convergence rates. In the event of high-dimensional stochastic space, one

could use Sparse Grids [68] to reduce the computational load.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, where Ω_s is the space of outcomes ω, $\mathcal{F}$ is the $\sigma-$algebra and $\mathbb{P}$ is a probability measure, $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$. In a general sense, we can formulate PCM for the QoI Q, and write its mathematical expectation $\mathbb{E}[Q(x, t; \xi)]$ in a one-dimensional stochastic space as

$$\mathbb{E}[Q(x, t; \xi)] = \int_a^b Q(x, t; \xi)\rho(\xi)d\xi, \tag{1.10}$$

where $\rho(\xi)$ is the PDF of random input variable ξ. We evaluate the integration using Gauss quadrature, mapping physical parametric domain to the standard domain $[-1, 1]$. The integral should then be written as

$$\mathbb{E}[Q(x, t; \xi)] = \int_{-1}^1 Q(x, t; \xi(\eta))\rho(\xi(\eta))Jd\xi(\eta), \tag{1.11}$$

where $J = d\xi/d\eta$ represents the Jacobian of the transformation. We approximate the expectation by introducing a polynomial interpolation of the exact solution in the stochastic space, $\hat{Q}(x, t; \xi)$:

$$\mathbb{E}[Q(x, t; \xi)] \approx \int_{-1}^1 \hat{Q}(x, yt; \xi(\eta))\rho(\xi(\eta))Jd\xi(\eta). \tag{1.12}$$

We interpolate the solution in the stochastic space using Lagrange polynomials $L_i(\xi)$:

$$\hat{Q}(x, t; \xi) = \sum_{i=1}^I Q(x, t; \xi_i)L_i(\xi), \tag{1.13}$$

which satisfy the Kronecker delta property at the interpolation points:

$$L_i(\xi_j) = \delta_{ij}. \tag{1.14}$$

Substituting Eq. (1.13) into (1.12), we approximate the integral using the quadrature rule and evaluate the expectation as

$$\mathbb{E}[Q(x, t; \xi)] \approx \sum_{p=1}^P w_p\rho(\xi(\eta))J \sum_{i=1}^I Q(x, t; \xi(\eta))L_i(\xi(\eta)), \tag{1.15}$$

where we compute the coordinates η_p and weights w_p for each integration point $q = 1, 2, \ldots, P$. We choose the collocation points to be the same as the integration points p on the parametric space

by the Kronecker property of the Lagrange polynomials Eq. (1.14), and simplify the approximation from Eq. (1.15) to a single summation:

$$\mathbb{E}\left[Q(x,t;\xi)\right] = \sum_{p=1}^{P} w_p \rho(\xi_p(\eta_p)) JQ(x,t;\xi_p(\eta_p)). \tag{1.16}$$

Through Eq. (1.16), we have an efficient integration scheme through Gauss quadrature, while we retain the property of independent realizations that allows for efficient implementation through parallel computing. The formulation is also open to other forms of polynomial expansion and choice of Gauss quadrature that are most suitable to each specific problem based on the definition of the parametric probability distribution. The PCM formulation presented here can be further applied to the computation of higher moments of the QoI, and in higher-dimensional spaces through the use of tensor product.

1.3 Towards a Predictive Multi-Scale Failure Model

As observed in experimental data, failure is inherently a stochastic and anomalous process. High-fidelity simulations of micro-structural physics come with the burden of high computational cost, which impedes the analysis from a probabilistic point-of-view. Therefore, detailed material models at this scale are often deterministic and do not provide sufficient information to the upper scales. Moreover, despite the existence of extensive literature on avalanche and crackling noise, and the use of mathematical models that accurately describe nonlocality at the micro-scale, there is still a gap in the incorporation of the sub-grid dynamics into continuum, macro-scale failure models. The use of nonlocal/fractional models has certainly advanced our understanding of complex phenomena in a tractable manner, relying on fewer inputs for a generalized representation. However, their numerical challenges, along with the necessity of pre-defining the nature of the interaction kernel still hinder their applicability in most practical problems.

At all scales, accurate and efficient propagation of uncertainty from input parameters to output solutions is another challenge. Intrusive methods require direct coding of stochastic effects into the physics solver. Alternatively, standard non-intrusive methods such as Monte Carlo suffer from slow

convergence and high computational costs. Therefore, efficient uncertainty quantification methods are paramount for failure predictability.

With the goal of providing a robust and predictive failure model at the continuum level, we propose the implementation of a probabilistic data-driven framework for failure analysis, which learns lower-scale parameter distributions to be upscaled to the continuum level. At the macro-scale, robust stochastic models and machine learning methods incorporate the underlying statistics to promote accurate predictions of material and component failure. The framework incorporates the following multi-disciplinary components:

- **Stochastic Modeling:** We incorporate the lower-scale stochastic processes to generate parameter distributions to be upscaled. Furthermore, we construct fast surrogate models to replace expensive high-fidelity solvers by exploiting the statistics of the underlying physical processes, leading to the construction of the corresponding stochastic model based on simulation data.

- **Uncertainty Quantification:** Given the stochastic nature of the coupling between different time and length-scales, we need proper UQ and sensitivity analysis (SA) tools to propagate the uncertainties across the scales. The curse of dimensionality in UQ is an issue that needs to be treated with fast and accurate methods in forward uncertainty propagation, such as Probabilistic Collocation Method (PCM) and Sparse Grids. We explore the capabilities of PCM to develop simple and cost-efficient reliability analysis methods.

- **Machine Learning:** Learning the nature of underlying anomalous physics in the form of nonlocal/fractional kernels and operators requires robust machine learning algorithms, where data mining of large-scale high-fidelity simulations is used to construct such representative stochastic models. Finally, Machine Learning is a robust tool to detect the presence of failure in macro-scale applications, even when the available data contains noise originated from experimental or simulation-based uncertainties.

1.4 Outline of this Work

This research work has the objective of developing a robust framework for multi-scale failure analysis. The ability to consider the evolution of defect networks, and how they affect the ultimate material behavior at the continuum, operational component level, is paramount for a predictive failure analysis. Moreover, we construct a framework capable of obtaining material parameter uncertainty at the micro-scale and to simulating failure at the macro-scale while considering parametric uncertainty. To that end, we show the development of stochastic surrogate models at the nano-scale which efficiently provides measures of parameter uncertainty. We demonstrate how the anomalous effects from collective dislocation dynamics in the micro-structure can be modeled probabilistically with nonlocal models. We propose a framework for incorporating parametric uncertainty in damage models at the continuum level and identifying the sensitive parameters. Furthermore, we develop a Machine Learning framework for failure detection at the continuum level that is robust to noise, either coming from experimental or modeling/parametric uncertainties. We illustrate the framework in Fig. 1.8. This work is divided in five chapters, which are summarized below.

Chapter 2: Understanding, modeling, and real-time simulation of the underlying stochastic micro-structure defect evolution is vital towards multi-scale coupling and propagating numerous sources of uncertainty from atomistic to eventually aging continuum mechanics. In particular, dislocation dynamics is directly connected to macro-scale plasticity, void and crack nucleation, and occurs across different time and length-scales. We aim to study the evolution of dislocation dynamics from the atomistic all the way to continuum level, and incorporate the anomalous dynamics usually disregarded in continuum,macro-scale failure morels. In this Chapter, we start by studying the dislocation mobility, which is a constitutive relation between dislocation velocity and effective shear stress. The Molecular Dynamics (MD) simulations often used to estimate mobility is computationally expensive, so with the goal of accelerating the computation of dislocation mobility, and to provide estimates of mobility uncertainty, we developed a graph-based surrogate model of dislocation glide at the atomistic level. Therefore, the main contributions of this Chapter are:

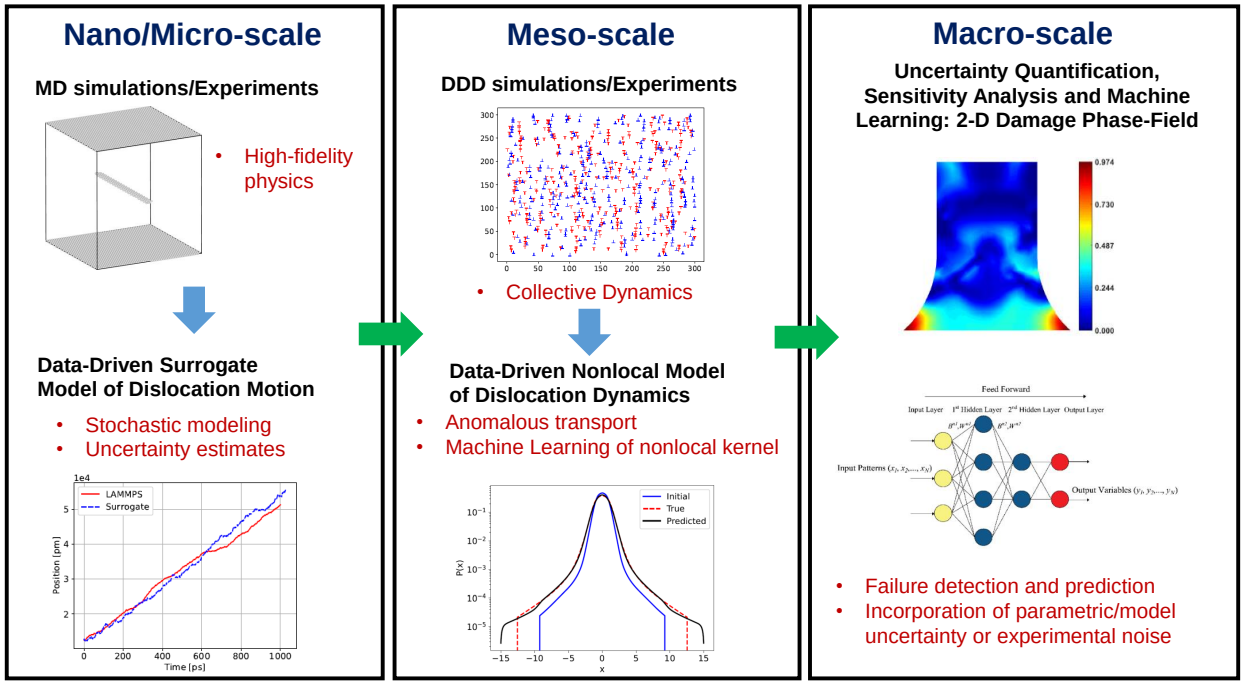


Figure 1.8 Illustration of the multi-scale failure analysis framework proposed. At each scale we learn the physics of their underlying processes, and obtain their corresponding stochastic models. At the macro-scale, robust uncertainty quantification and machine learning algorithms will detect the presence of damage and predict the material failure.

- We develop a graph-based surrogate model of dislocation glide for computation of stochastic dislocation mobility. We model an edge dislocation as a random walker, jumping between neighboring nodes of a graph following a Poisson stochastic process.

- The network representation functions as a coarse-graining of an MD simulation that provides dislocation trajectories for an empirical computation of jump rates, such as forward and backwards jump statistics, and waiting time distribution, which we then use to parameterize the Poisson process on the surrogate model.

- We simulate the dislocation movement on the surrogate model by employing a Kinetic Monte Carlo method, and recover the original atomistic mobility estimates, with remarkable computational speed-up and accuracy.

- Furthermore, the underlying stochastic process provides the statistics of dislocation mobility associated to the original molecular dynamics simulation, allowing an efficient propagation

of material parameters and uncertainties across the scales, establishing a meaningful link for predictive multi-scale failure modeling.

Chapter 3: The collective motion of dislocations at the meso-scale leads to interesting phenomena observed experimentally such as intermittent energy signals, power-law distribution of energy spectrum, crackling noise, all of which are indicators that avalanches and scale-free dynamics occurs during the failure process [11–15]. Experimental evidence suggests that at the meso-scale, dislocation dynamics has nonlocal character due to the collective dynamics.

In this Chapter, we investigate the nonlocal dynamics of collective dislocation motion through a probabilistic perspective, in which the evolution of the probability density function of dislocation position is governed by a nonlocal transport model. Specifically, the main contributions of this Chapter are:

- We simulate a 2-dimensional discrete dislocation dynamics (DDD) problem under periodic boundary conditions, where the dislocations move in a single glide plane.
- The collective dynamics of the dislocation population give origin to Lagrangian trajectories that encode the underlying stochastic process directly. We collect all the trajectories of dislocation motion to generate the PDFs of dislocation position.
- We use a Machine Learning framework to solve the inverse problem of parameterizing the nonlocal operator that solves the PDF evolution. We employ a bi-level learning framework that takes time-series evolution of PDFs and learn the power-law exponent α and horizon size δ.
- We identify the shapes of the nonlocal kernel and observe that the presence of dislocation multiplication induces kernels associated with super-diffusive processes.

Chapter 4: Failure analysis often relies on uncertain data, whether it is coming from experiments, where measurements contain significant levels of noise, or it is coming from their own mathematical

models, in the form of uncertain mathematical operators or parameter definition. Therefore, in order to develop a robust and predictive failure analysis framework, we need a systematic way of analyzing the model uncertainties. In particular, uncertainty quantification also points to the operators which could be improved with the goal of mitigating uncertainty.

In this Chapter, we motivate the necessity of different modling approaches that include the anomalous effects that occur in the material during failure. We consider a stochastic phase-field model at the continuum level, simulating failure through introducing damage and fatigue variables. The damage phase-field is introduced as a continuous dynamical variable representing the volumetric portion of fractured material and fatigue is treated as a continuous internal field variable to model the effects of micro-cracks arising from energy accumulation. We formulate a computational-mathematical framework for quantifying the corresponding model uncertainties and sensitivities in order to unfold and mitigate the salient sources of unpredictability in the model, hence, leading to new possible modeling paradigms. The contributions in this Chapter are:

- We consider 5 parameters related to damage and fatigue to be random variables with a specific range. We solve the equations with Finite Element Method in space, and a semi-implicit method in time. In the stochastic space, we compared the solution statistics of a Monte Carlo Method (MC) and Probabilistic Collocation Method (PCM). We chose the PCM since it discretizes the stochastic space using a Lagrange interpolation of the solution, giving exact integrations in the stochastic space, with just a few realizations, and in our results we found that indeed PCM have a faster convergence than MC.

- We used PCM as a building block for the uncertainty and sensitivity analyses of the damage model. We performed a Local Stochastic Sensitivity Analysis for the 5 parameters by taking the expectation of local sensitivity over the parametric domain range, where the local sensitivity at each point was computed by Complex-Step Differentiation, which is faster and more accurate than Finite-Difference. Then, we performed Global Sensitivity Analysis to quantify the contribution of each parameter into the total variance of the solution. An

Analysis of Variance (ANOVA) based on Sobol index was performed using PCM, where we post-process the realizations in the tensor product of collocation points, which is fast and accurate, contrary to MC.

- In the end, we analyzed two representative problems, a single-edge notched tensile test, and an I-shaped specimen. From the notched case, since we have a pre-existing crack with a known propagation direction due to the loading conditions, we observed that the parameters that contribute most to the solution uncertainty are related to damage evolution rate. For the I-shaped specimen, we found that parameters with more contribution were related to the damage free-energy potential that originated the classical Laplacian operator. The results point to the most sensitive parameters, indicating where we can modify the model to mitigate uncertainty. Our results suggest a revision of the classical free-energy, with the inclusion of neglected nonlocal terms that upscale the anomalous effects disregarded in current models.

Chapter 5: Failure in brittle materials led by the evolution of micro- to macro-cracks under repetitive or increasing loads is often catastrophic with no significant plasticity to advert the onset of fracture. Early failure detection with respective location are utterly important features in any practical application, both of which can be effectively addressed using artificial intelligence.

In this Chapter, we developed a Machine Learning framework for failure prediction of brittle materials. We combine a classification algorithm with a pattern recognition scheme, where we select virtual nodes from Finite Element results of the phase-field damage model to construct a degradation function, and generate patterns of material softening. From each sensing node, we extract a time-series such that we obtain a pattern at each time-step. We investigated the spatial location (by analyzing different patterns) and time occurrence (by defining criteria for onset of failure, and failure state) of the fracture by k-Nearest Neighbors (k-NN), and Artificial Neural Networks (ANN) algorithms to analyze and classify the patterns. The major contributions of the Chapter are:

- Time-series data of the phase-field model is extracted from virtual sensing nodes at different

locations of the geometry.

- A pattern recognition scheme is introduced to represent time-series data/sensor nodes responses as a pattern with a corresponding label, integrated with ML algorithms (k-NN and ANN), used for damage classification with identified patterns.
- We perform an uncertainty analysis by superposing random noise to the time-series data to assess the robustness of the framework with noise-polluted data.
- Results indicate that the proposed framework is capable of predicting failure with acceptable accuracy even in the presence of high noise levels. The findings demonstrate satisfactory performance of the supervised ML framework, and the applicability of artificial intelligence and ML to a practical engineering problem, i.,e, data-driven failure prediction in brittle materials.

Chapter 6: We discuss the main conclusions of this work, and comment on future research steps.

CHAPTER 2

ATOMISTIC-TO-MESO MULTI-SCALE DATA-DRIVEN GRAPH SURROGATE MODELING OF DISLOCATION GLIDE

2.1 Introduction

Multi-scale materials modeling and simulations are a rapidly growing scientific field, where it is critical to propagate uncertainties to accurately and efficiently bridge material properties between adjacent length- and time-scales. Among several types of material imperfections that cause disturbances in crystal structures, dislocations are line defects [1] that are naturally present from manufacturing until failure of crystalline materials. Describing the small-scale buildup and dynamics of dislocations can provide an important insight on early fatigue precursors [69, 70], which are beyond the resolution of existing continuum models of high-cycle fatigue damage. In order to accurately propagate such early statistics of failure to the continuum for large-scale applications, consistent, robust and efficient coupling frameworks between the atomistic and meso-scales are fundamental.

Molecular dynamics (MD) is a first-principle theory that explicitly describes the motion individual atoms at small scales based on Newton's second law. In the context of dislocations, MD has been employed as an effective tool for the atomistic understanding of canonical types of dislocation motion for diverse crystal structures and their corresponding mobility/drag coefficients [71–75], as well as the estimation of core energies, responsible for dislocation self-interactions [23, 76]. In order to describe the complex arrangements and mechanics of dislocation networks at the intermediate scale of scanning electron microscopy [77], discrete dislocation dynamics (DDD) has become a practical computational tool [27] that allowed the discovery of new physics, such as dislocation multi-junctions [78]. Accurate DDD simulations require precise experimental properties from dislocations and the corresponding medium, which can be obtained through MD experiments. However the large number of degrees of freedom required for robust MD simulations may render

such experiments prohibitive, especially when a large number of realizations is needed to propagate the statistical qualities from small- to large-scales.

Aiming to simulate processes at longer time-scales, while still respecting the intrinsic physics of lower-scale dynamics, different approaches have emerged. Kinetic Monte Carlo (KMC) methods became popular in the last decades in a myriad of materials science applications. KMC is a type of continuous-time Markov process [79, 80], where the process rates should be known in advance. This method appeared originally for simulation of vacancies [81] and Ising spin systems [82], gaining popularity among a variety of applications, including crystal growth [83], visco-elasticity [84], and surface kinetics [85]. Researchers have also used KMC methods to construct low-fidelity models for dislocation motion in materials ranging from bcc metals [86] to Silicon [87–89], where temperature, size, and stress effects are investigated. More recently, [90, 91] used KMC to study the interaction between solute atoms and screw dislocation bcc metals. This approach has the advantage to capture rare thermally-activated motions, which is not possible in MD simulations [92]. However, such models are limited due to uncertainties in atomistic estimation of parameters used in the computation of rate constants, commonly obtained from activation energies derived from transition State theory [93]. Phase Field Crystal (PFC) is another fast growing method for simulation of crystalline structures with atomistic detail, while reaching diffusive time-scales, and has been used to model dislocation dynamics [94–97].

More recently, graph theory [98] has also presented itself as a robust approach in the field of materials science, with applications in coarse-graining [99], and chemical kinetics, in combination with KMC method [100]. Graph theory has a leading potential to provide efficient coarse-graining of micro-scale dynamics, furnishing suitable ground for stochastic simulations of underlying dislocation dynamics through a random walk over a network. For an extensive review of random walks on networks, we refer the reader to [101] and references therein.

In this work, we propose a data-driven framework for the construction of a surrogate model of edge dislocation glide at the atomistic level, where dislocation position as a time-series data is collected from high-fidelity MD simulations to train the model. We first perform a coarse-

graining of the atomistic domain through a graph-theoretical formulation. In the case of dislocation glide in a periodic domain, a ring graph provides an accurate representation. However, the general construction of the network and associated operators allows further enhancements for more complex dynamics in a direct way. We model dislocation motion as a random walker, jumping between neighboring nodes on the network, following a continuous-time, Markovian stochastic process. The waiting times for forward or backward jumps between neighboring nodes is exponentially distributed with rate parameter directly computed from the MD time-series data. We supply a KMC algorithm with the estimated rate constants to simulate the dislocation motion under different applied shear stresses, providing fast and accurate calculations of dislocation velocity and mobility.

Ultimately, beyond the efficient estimates of material properties at the atomistic-level, the proposed framework allows the propagation of uncertainties across the scales. With the stochastic description of dislocation motion through a random walk over a network, governed by a Markov jump process, we can compute statistics associated to the dislocation motion that are intrinsically attached to the original atomistic setup. In that sense, we provide a mobility experiment of similar nature to the associated MD simulation, with the advantage of quantifying parametric uncertainty, which would be prohibitively costly through high-fidelity MD simulations. This approach differs from existing KMC dislocation models that simulate dislocations at the meso-scale [87], and expands over current atomistic coarse-graining methods for dislocations [102], which do not estimate uncertainties associated with the high-fidelity simulations. To the best of our knowledge, this is the first computational effort in providing uncertainty estimates of dislocation mobility properties from the atomistic level. Mobility estimates and associated uncertainties provided by the surrogate model can later be upscaled to meso-scale dislocation simulations, such as DDD. At that stage, the collective behavior of dislocations would intrinsically incorporate stochastic effects of lower scales that would be propagated to the continuum (*i.e.*, through dislocation density and plastic strains), therefore providing efficient multi-scale coupling starting in the MD domain. This feature is essential to the development of predictive models at the component level, whether the interest is on visco-elasto-plasticity [46, 47, 103], fracture [104, 105] or fatigue [34].

2.2 Data-Driven Framework

We develop a surrogate model for dislocation glide parameterized by MD data to quickly obtain estimates of dislocation mobility in a short time-frame. The numerical framework for model construction and simulation is illustrated in Fig. 2.1. To construct the surrogate, the atomistic domain is coarse-grained and idealized as a periodic line graph (a ring graph), where nodes correspond to the sub-domains inside the crystal.

From the coarse-grained description, we represent the dislocation as a random walker that jumps between neighboring nodes following a Poisson stochastic process. The rate constants that parameterize the process are obtained directly from MD simulation data of an edge dislocation gliding under shear stress, allowing the reconstruction and simulation of the stochastic dislocation motion through KMC method. KMC and MD are independent techniques for dislocation motion, yet here we combine both, leading to a fast computation of dislocation mobility using KMC, in which the parameters come from high-fidelity, costly, MD simulations.

We start by discussing the methodology of dislocation simulation through MD. Then, we describe the coarse-graining of the physical domain as a ring graph, and construct the dislocation random walker based on Poisson processes. Computing the rate constants from MD simulations, we ensure that sequences of States coming from KMC converge in distribution with MD trajectories [79], yet using far less computation time, allowing for longer simulation times that are not achievable in MD.

2.2.1 Molecular Dynamics Simulation of Edge Dislocation Glide

Following body-centered-cubic Fe-C simulations from [23], we generate synthetic dislocation motion data in a pure Fe system and estimate the edge mobility property through MD simulations utilizing the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [106]. All the MD simulations in this work are run in 80 Intel Xeon Gold 6148 CPUs with 2.40GHz.

The MD system under consideration is illustrated in Fig. 2.2, consisting of a simulation box of

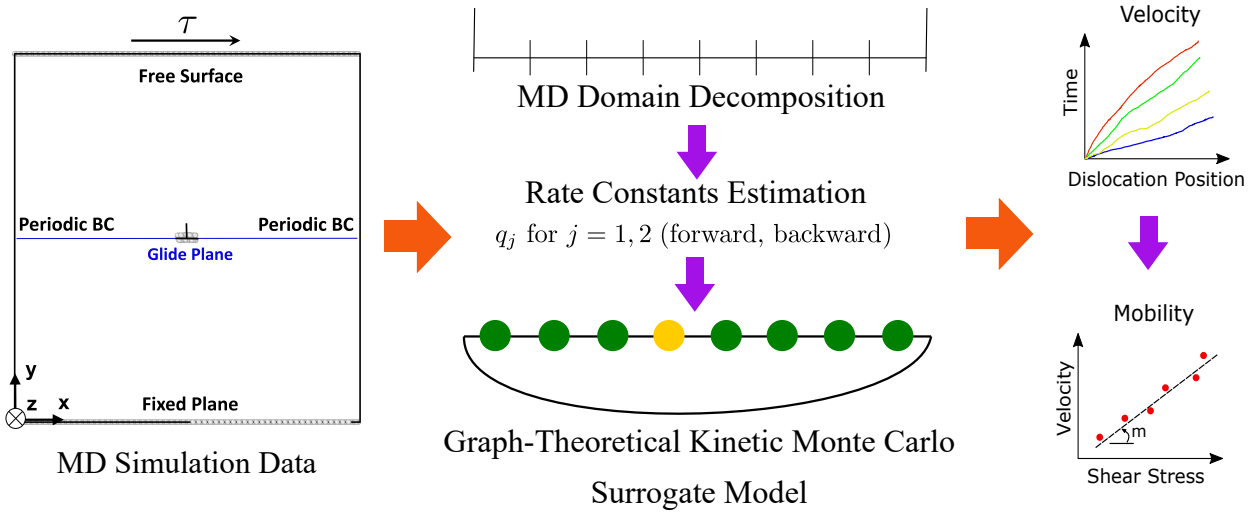
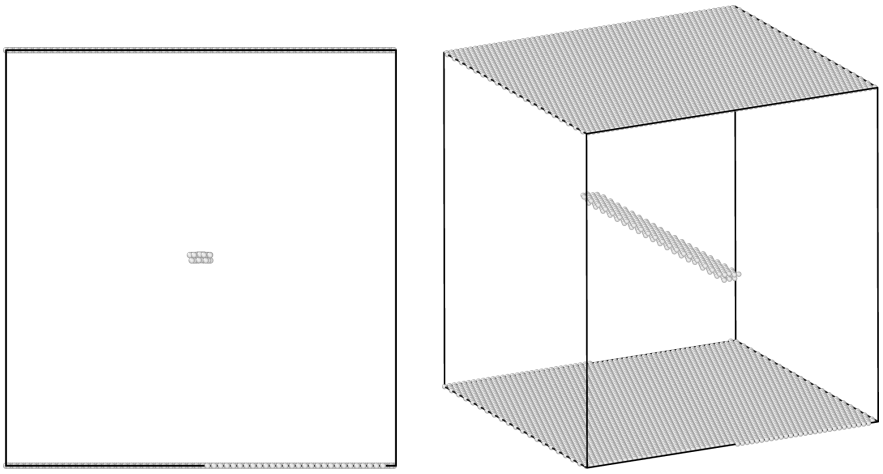


Figure 2.1 Framework for constructing a network-based KMC surrogate model for dislocation glide. The surrogate is then employed for fast and accurate simulations of dislocation motion, obtaining velocity data at different stress levels, leading to the estimation of the dislocation mobility.

$61 \times 40 \times 20$ α-Fe unit cells with dimensions $25.14 \times 26.96 \times 24.06$ $[nm]$ in the x, y, z directions. A straight edge dislocation with Burgers' vector $\mathbf{b} = \frac{1}{2}[1, 1, 1]$ is generated by removing a $(1, 1, 1)$ half-plane of atoms from the center of the box. The MD domain consists of 1 353 132 atoms with periodic boundary conditions applied in the x and z directions, and shrink-wrapped boundary conditions applied to the unit cells in the top- and bottom-planes along the y-direction. We perform an NVE time-integration, where the system's temperature is relaxed to $T = 750\,[K]$ through velocity-rescaling for $100\,[ps]$ *(see Fig.2.3a)*. We utilize a combined Tersoff bond-order and repulsive Ziegler-Biersack-Littmark (ZBL) interatomic potential, with corresponding parameters from [107].

We apply shear stress values in the range $\tau \in [15, 100]\,MPa$ to the top layer in Fig. 2.2a, parallel to $\mathbf{b}$, which induces a glide motion in the x-direction on the $(1, 1, 0)$ plane. No temperature control is enforced in this stage and we run the simulation over $1\,[ns]$ with time-step size $\Delta t^{MD} = 2\,[fs]$. The MD time-series data is saved every 100 time-steps and the atom positions are post-processed utilizing the Polyhedral Template Matching (PTM) method [108] implemented in OVITO (`https://www.ovito.org/`) [109], which allows us to detect and track the lattice disturbance. We define the dislocation position as the average of all x-coordinates of atoms belonging to the



(a) (b)

Figure 2.2 MD domain of the dislocation mobility test. (a) $x - y$ plane, illustrating the edge dislocation core as the lattice perturbation at the center. (b) 3D view of the MD domain with the BCC lattice removed, showing the dislocation line along the z-axis.

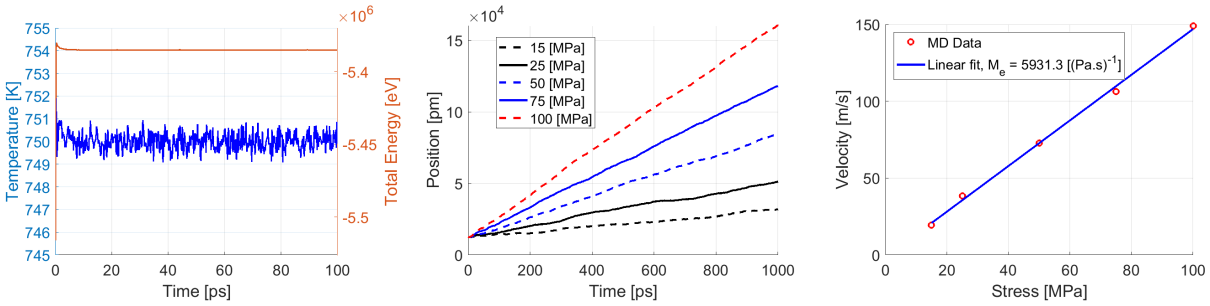
disturbed region (dislocation core) in Fig.2.2a. Therefore, for every applied shear stress τ, we obtain a position vector $\mathbf{x}^{MD}(t)$ with 5000 data-points (*see Fig. 2.3b*) of size $\Delta t^{MD} = 2\,[fs]$, from which we compute the corresponding velocity v^{MD} through a linear fit. The obtained velocity from the post-processed MD simulation can be related to the one-dimensional solution from dislocation dynamics denoted by v_x, and given by the following relationship:

$$v_x = M \cdot b \cdot \tau. \tag{2.1}$$

where M denotes the edge dislocation mobility, and $b = \sqrt{3}a/2$ represents the magnitude of $\mathbf{b}$. Equation (2.1) is obtained from the balance between the applied Peach-Kohler force induced by the shear stress τ and the dislocation drag force. Therefore, setting $v^{MD} = v_x$ and from the slope $m = M \cdot b$ of the velocity *versus* stress curve in Fig. 2.3c, we estimate the edge dislocation glide mobility as $M = m/b \approx 5931.3\,[(Pa.s)^{-1}]$, which is in good quantitative agreement (1.73% difference) compared to the results obtained by Lehtinen et al.[23].

2.2.2 Graph-theoretical Coarse-graining

We begin the surrogate framework by idealizing a coarse-grained version of the atomistic domain as a graph $G(V, E)$, with a set vertices (or nodes) V connected by edges E. In this representation,



(a) (b) (c)

Figure 2.3 (a) Temperature and total energy for the equilibration step, (b) Edge dislocation position $\mathbf{x}^{MD}(t)$ and (c) mobility through MD simulations for distinct values of applied shear stresses τ under $T = 750\,[K]$. We observe an overdamped motion for the applied shear stress range and a linear mobility relationship.

each node on the network represents a sub-domain from the original MD system. In the case of a dislocation glide along a single slip plane, a one-dimensional ring graph is an adequate simplification of the dislocation movement, also assuring the periodicity presented in the MD domain.

The coarse-graining is achieved by dividing the domain into n sub-domains, or bins, such that

$$n = \left\lfloor \frac{L}{\max(\boldsymbol{d})} \right\rfloor \tag{2.2}$$

where L is the size of the domain (in the $x - y$ plane), and $\boldsymbol{d}$ is the vector containing the distance traveled by the dislocation between each MD time-step, with entries $d_i = x_{i+1}^{MD} - x_i^{MD}$. We choose this upper bound to ensure that the dislocation only travels to neighboring nodes. In this sense, we identify the dislocation as corresponding to node i of the graph if the dislocation position in the MD simulation lies between the bounds of bin i of width $\Delta x = L/n$.

We now define the standard operators for a continuous-time random walk on a network. The adjacency matrix $\boldsymbol{A}$ has elements $A_{ij} = 1$ if there is a link between nodes i and j, and $A_{ij} = 0$ otherwise, for $i, j = 1, 2, \ldots, N$. The degree matrix $\boldsymbol{K}$ represents the number of edges attached to the node, computed as $K_{ii} = \sum_{j=1}^{N} A_{ij}$, and $K_{ij} = 0$ for $i \neq j$. From $\boldsymbol{A}$ and $\boldsymbol{K}$ we define the transition matrix $\boldsymbol{W}$, with elements $w_{i\to j} = \frac{A_{ij}}{K_i}$, representing the probability of the random walker to transition from node i to node j.

Specifically for the ring graph considered for the surrogate model, every node is attached to two other nodes, which makes entries $A_{ii+1} = A_{ii-1} = 1$, except when $i = 1$ or $i = N$. In those cases, A_{1n} and A_{n1} are set to one to ensure periodicity. As a consequence, degree matrix $\boldsymbol{K}$ has all entries $K_{ii} = 2$.

The transition matrix is finally computed with elements $w_{i\to i} = w_{i\to i+1} = w_{i\to i-1} = \frac{1}{2}$. Again, the exception is for nodes $i = 1$ and $i = N$, where we obtain $w_{1\to N} = \frac{1}{2}$ and $w_{N\to 1} = \frac{1}{2}$, respectively, due to periodicity. At this point we use the transition matrix $\boldsymbol{W}$ as a building block for introducing the dynamics of dislocation motion. Its purpose is to initially restrict the movement of the random walker to the neighboring nodes with equal probability, later modulated by empirical rates computed from MD simulations.

2.2.3 Construction of the Random Walk

The construction of the random walk representative of dislocation glide is dependent on two main aspects: first, on the simplification of dislocation motion and its coarse-graining through a graph-theoretical framework, as discussed before; second, on the statistical representation of dislocation mobility through a Poisson process that naturally leads to the use of KMC method. We now discuss the formulation of the random walk, where we will follow closely the ideas in [79].

The main attractiveness of KMC is the simplification of complex dynamics into a counting process, where the entire system moves from State to State. For each possible escape path from the current State, there is an associated rate constant q_{ij}, which is the probability per unit time to transition to State j from State i.

For modeling the dislocation motion through a random walk, we first assume that the transition probabilities for dislocation motion are independent of history, therefore, characteristic of a Markov processes. Second, for systems such as the pure Fe-Fe studied in this work, there is no evident acceleration of dislocation in the long range. Therefore, we assume that the underlying process is stationary with independent increments.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, where Ω is the space of outcomes ω, $\mathcal{F}$ is the

σ-algebra and $\mathbb{P}$ is a probability measure, $\mathbb{P} : \mathcal{F} \rightarrow [0; 1]$. From the assumptions, we model the total number of jumps $N_t(t)$ between States over time $t \in [0, \infty)$ as a Poisson process with total rate Q, such that for any t, $N_t(t) \sim \text{Poisson}(Qt)$.

For an arbitrary process with several possible States j from current State i, each with rate q_{ij}, the total rate Q is $Q = \sum_j q_{ij}$, following the assumption that the different processes are independent and non-overlapping. In the dislocation motion studied here, there are only two possible escape paths from any current State, a forward or backward jump, with respective rates q_f and q_f. Therefore, we have

$$Q = q_f + q_b. \tag{2.3}$$

Furthermore, let $X : \Omega \rightarrow \mathbb{R}$ be a random variable that represents the waiting times between jumps over the graph G. Then, $X \sim \text{Exponential}(Q)$, meaning that the process is first-order with exponential decay statistics, i.e., memoryless. The probability of the random walker not performing any jump, therefore staying on the current node, is given by

$$p_{\text{stay}}(t) = e^{-Qt}, \tag{2.4}$$

leading to the standard computation of time increments Δt in KMC algorithms,

$$\Delta t = -\frac{\ln(r)}{Q}, \tag{2.5}$$

where r is a random number sampled from the uniform distribution $\mathcal{U}(0, 1)$.

After each time-step with size given by Eq. (2.5), the system will evolve to a new State with probability proportional to q_f and q_b. In general, this is accomplished by recomputing the elements of $\boldsymbol{W}$ as p_{ij}, representing the probability of a jump per unit of time, in units of s^{-1}. Probabilities are obtained through

$$p_{ij} = \frac{w_{i\rightarrow j} q_j}{\sum_j w_{i\rightarrow j} q_j}, \tag{2.6}$$

where p_{ij} is now the walker's probability to go from node i to node j, per unit time . The result is normalized to make $\sum_j p_{ij} = 1$. Equivalently, we may simply take

$$p_{ij} = \frac{q_j}{\sum_j q_j} = \frac{q_j}{Q} \tag{2.7}$$

for $q_j \in \{q_f, q_b\}$.

Remark 2.2.1. *Note that the increment in time and the selection of the next State are independent of each other. First the system waits for any jump with probability related to the total jump rate Q. Then, in a separate drawing, the next State is chosen with probabilities proportional to q_f and q_b.*

Remark 2.2.2. *The general graph-theoretical description of the physical system allows flexibility and future incorporation of more complex cases, beyond the ring graph currently adopted for the case of dislocation glide. The inclusion of inhomogeneous Poisson processes (either in time or space), dislocation climb, or even non-Markovian network dynamics as in the case of Lévy flights [110] can be built on top of this fundamental framework in a straight-forward fashion.*

Since the graph nodes are positioned in the center of each bin, as illustrated in Fig. 2.1, we have an approximation for distance traveled by the dislocation from the internodal distance Δx. Then, at each time-step, the dislocation spatial position is updated by

$$\begin{cases} x^{n+1} = x^n + \Delta x, & \text{if dislocation jumps forward} \\ x^{n+1} = x^n - \Delta x, & \text{if dislocation jumps backwards} \end{cases} \tag{2.8}$$

where x^{n+1} is the new dislocation position ate time-step t_{n+1}. In that sense, this model is still a discrete-space random walk, which calls for extra care when computing the dislocation velocity.

One possibility is to mimic the procedure from MD simulations, and plot the dislocation distance as a function of time, performing a linear regression to obtain the dislocation velocity v. We run the simulation for each stress level, and plot the dislocation velocity as a function of stress. Again, a linear regression is used to obtain the slope of the curve m for a linear mobility rule as in MD, and the dislocation mobility from the network dynamics is estimated through Eq. (2.1).

Algorithm 2.1 summarizes the procedure of running a KMC simulation of dislocation glide through a random walk on a network for a total of M time-steps when we know the rates of forward q_f and backward q_b jumps. We implemented Algorithm 2.1 in Python 3.7, in addition to a routine for the computation of empirical jump rates from MD. The ring graph, corresponding matrices, and operators are constructed using the NetworkX Python package [111].

Algorithm 2.1 Kinetic Monte Carlo method for Dislocation Glide as a Random Walk on a Graph

Given: rates for jump forward q_f and jump backward q_b, compute total rate through Eq. (2.3).
Given: number of nodes n through Eq. (2.2), and the distance between nodes Δx, compute transition matrix W.
for Time-steps $m = 0 \rightarrow M - 1$ **do**
    Given the current node position i, get the corresponding $i-$th line of $\boldsymbol{W}$.
    Update line $\boldsymbol{W}_i$ as in Eq. (2.7).
    Choose next position j based on the pdf given by W_i.
    Generate a random number $r \sim \mathcal{U}(0, 1)$.
    Advance time by a time-step Δt from Eq. (2.5).
    Update the dislocation's spatial position by Δx using Eq. (2.8).
end for

2.2.4 Empirical Computation of Rate Constants

One of the major drawbacks of KMC methods is the required knowledge of process rates as inputs to the method, which may not always be a trivial task, where traditional approaches involve the computation of rates through physical principles [79, 93]. In this work, we propose a data-driven approach for the computation of jump rates from dislocation position data obtained in MD simulations. In this way, the atomistic, high-fidelity simulation with observable dislocation motion parameterizes the surrogate model through the rate constants.

From the coarse-graining procedure, at each time-step we can identify and track the node associated with the dislocation position in MD. With this information, we are able to compute the waiting times between two consecutive jumps, classified in three main groups of events: forward, backward, or any jump. We also compute the total number of jump events in any of the three groups, respectively N_f, N_b, and $N_t = N_f + N_b$. Both groups of data can be used to estimate the rate constants.

We model $N_t(t)$ following a Poisson distribution, and given that the expectation of a Poisson random variable with parameter $\lambda = Qt$ [112] is

$$\mathbb{E}[N_t(t)] = Qt, \tag{2.9}$$

we may infer the rate parameter Q from empirical data by taking

$$Q = \frac{\mathbb{E}[N_t(t)]}{t}. \tag{2.10}$$

The expected number of jumps $\mathbb{E}[N_t(t)]$ is taken here to be the number of jumps that occurred in the MD simulation during simulation time t. Equivalently, we can replace $N_t(t)$ by N_f and N_b, to respectively compute q_f and q_b.

Alternatively, we can look at the probability that a jump happened by time t', which is the integral of the probability of the first jump $p(t)$, and it is given by

$$\int_0^{t'} p(t)dt = 1 - p_{\text{stay}}(t'). \tag{2.11}$$

It follows that $p(t)$ can be obtained by taking $p(t) = -\frac{\partial p_{\text{stay}}(t)}{\partial t}$, so that

$$p(t) = Qe^{-Qt}, \tag{2.12}$$

which is an exponential distribution of waiting times. Taking the first moment of Eq. (2.12) gives the average waiting time for a jump μ as

$$\mu = \int_0^{\infty} tp(t)dt = \frac{1}{Q}. \tag{2.13}$$

Note that again we may generalize the result from Eq. (2.13) to average waiting time between two consecutive forward and backward jumps exclusively, μ_f and μ_b, just by isolating those events from the complete time-series of waiting times. In that case, we can also obtain q_f and q_b from waiting time distributions.

The last method we may use to compute the rate constants is also through distributions of waiting times. Yet, this time we fit an exponential function to the histogram of waiting times using Maximum Likelihood Estimation (MLE). The MLE estimator for an exponential fit is equivalent to the reciprocal of sample mean, i.e. $1/\mu$, therefore we can expect identical results when using both methods [113]. We compare the accuracy of all three methods in the following section by using user-defined true rates as reference solution.

2.3 Results and Discussion

We now present the numerical results from the surrogate model simulations. We start by investigating the accuracy of the rate estimation algorithm, and the convergence as a function of the number

of time-steps from the original data-set using manufactured known process rates. Then, we apply the framework to real MD simulation data of dislocation glide and compute the mobility using the surrogate, comparing the results with MD mobility computations.

2.3.1 Convergence of Rate Constant Estimation

We investigate the accuracy and convergence of the rate estimation algorithm through KCM simulation of a single random walker in a ring graph, with manufactured true rates q_{true} for forward and backward jumps. We test different rate combinations for the jumps, and apply Eqs.(2.10) and (2.13), and MLE to estimate the original rates in one realization of the stochastic process. We check the convergence of the rate estimate with different number of time-steps, which in this case is the exact number of total jumps $N_t(t)$. We consider a graph with $n = 20$ nodes.

We show results in Tables 2.1 and 2.2, for the estimation through Eq. (2.10). The other two methods yield identical results for the manufactured solution, and are omitted. We present the estimated rates q_{est}, and the relative error to the true rates, computed as

$$\text{error} = \frac{|q_{\text{true}} - q_{\text{est}}|}{|q_{\text{true}}|}. \tag{2.14}$$

We observe that accuracy is dependent on the number of time-steps, which is natural, since more time-steps provide more data for a reliable statistical representation of the true rates. Second, the estimate is more accurate for higher rates, relative to lower ones, as in Table 2.1, where the ratio between the rates is large. For rates of similar magnitude, error levels are comparable, since there is sufficient data for both estimates.

2.3.1.1 Uncertainty quantification of rate estimation

Due to the probabilistic nature of this framework, results from Tables 2.1 and 2.2 show oscillations in error measures, which only represent the accuracy of a single realization of the problem in the stochastic space. This motivates an Uncertainty Quantification (UQ) analysis, where we using

Table 2.1 True rates: 200 (forward) and 1 (backward), in units of s^{-1}.

Number of time-steps	Forward Rate	Error	Backward Rate	Error
10^1	191.9520	4.02%	0.0000	100.00%
10^2	202.8340	1.42%	0.0000	100.00%
10^3	214.0373	7.02%	1.9438	94.38%
10^4	197.1265	1.44%	0.9309	6.91%
10^5	199.2291	0.39%	0.9066	9.34%
10^6	200.0162	0.01%	0.9272	7.28%

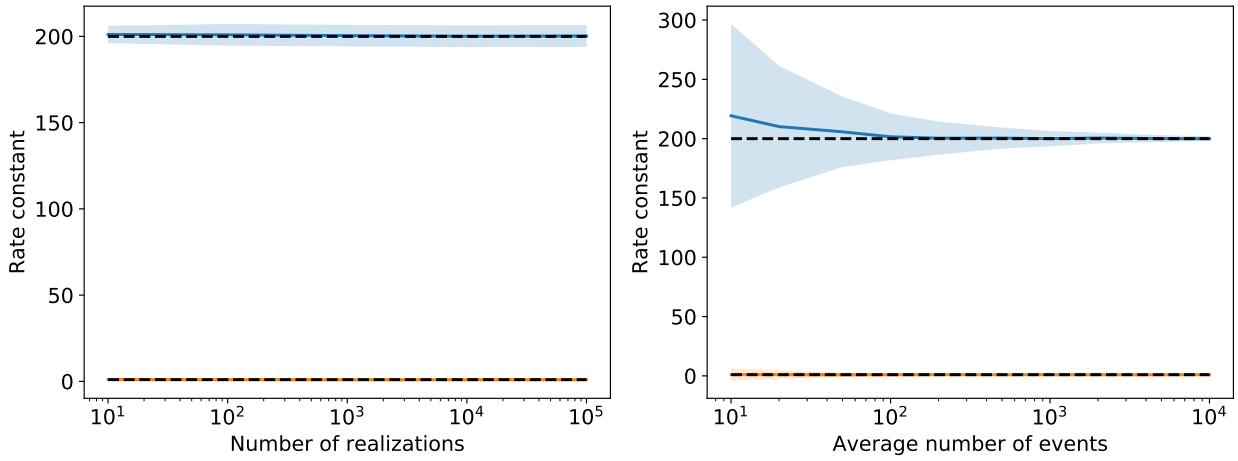
Table 2.2 True rates: 100 (forward) and 100 (backward), in units of s^{-1}.

Number of time-steps	Forward Rate	Error	Backward Rate	Error
10^1	51.5011	48.50%	51.5011	48.50%
10^2	124.4437	1.82%	101.8176	1.82%
10^3	108.6405	3.66%	96.3416	3.66%
10^4	98.6444	0.40%	100.3960	0.40%
10^5	100.3814	0.32%	99.6812	0.32%
10^6	99.8083	0.04%	100.0381	0.04%

Monte Carlo method to quantify the level of uncertainty in the rate estimation for data with different number of time-steps.

Two types of analysis were performed. First, for a fixed number of 1000 time-steps, expectation and standard deviation were obtained for different number of MC realizations. Last, for a fixed number of 1000 realizations, we obtained expectation and standard deviation for different number of time-steps, i.e., by considering different final simulation times from the time-series data. To show this result, we average the number of statistical events (total jumps) from each realization to construct the x-axis. For the same network of $n = 20$ nodes as before, and true rates of $q_{f,\text{true}} = 200\ s^{-1}$ and $q_{b,\text{true}} = 1\ s^{-1}$ as in Table 2.1, we plot UQ results in Fig. 2.4.

From Fig. 2.4 we see that the precise computation of rates from data is almost exclusively dependent on the number of statistical events, therefore on the length of the simulation. Increasing the number of realizations does not increase the accuracy of the recovered rates, and the uncertainty region is kept constant. However, increasing the number of time-steps through considering longer simulation times leads to the expected value to converge to the true rate, and shrinks the uncertainty



(a) Number of realizations.

(b) Number of time-steps.

Figure 2.4 Convergence to true rates (y-axis) as a function of number of realizations with fixed time-steps (a), or number of time-steps with fixed realizations (b). Dashed lines are the true rates (200 and 1), solid lines are the expected rates, and the shaded areas are the regions of uncertainty based on standard deviation.

region.

2.3.2 Dislocation Mobility

Here we present numerical results for one complete cycle of the framework, from MD simulation of dislocation glide, to rate estimation and final surrogate simulation through a random walk in the constructed network.

2.3.2.1 Rate estimation

From the raw data of dislocation position and time obtained from LAMMPS and Ovito, we apply the domain decomposition into bins equivalently to graph nodes, and track the current node over time. We count the number of jumps forward and backward between two nodes, as well as the waiting times between events. This also allow us to compute the waiting times between two forward or backward jumps.

Now we show the rate estimation procedure. First, we compile the waiting time statistics in histograms, and plot the normalized histograms with a corresponding exponential fit in Fig. 2.5 for

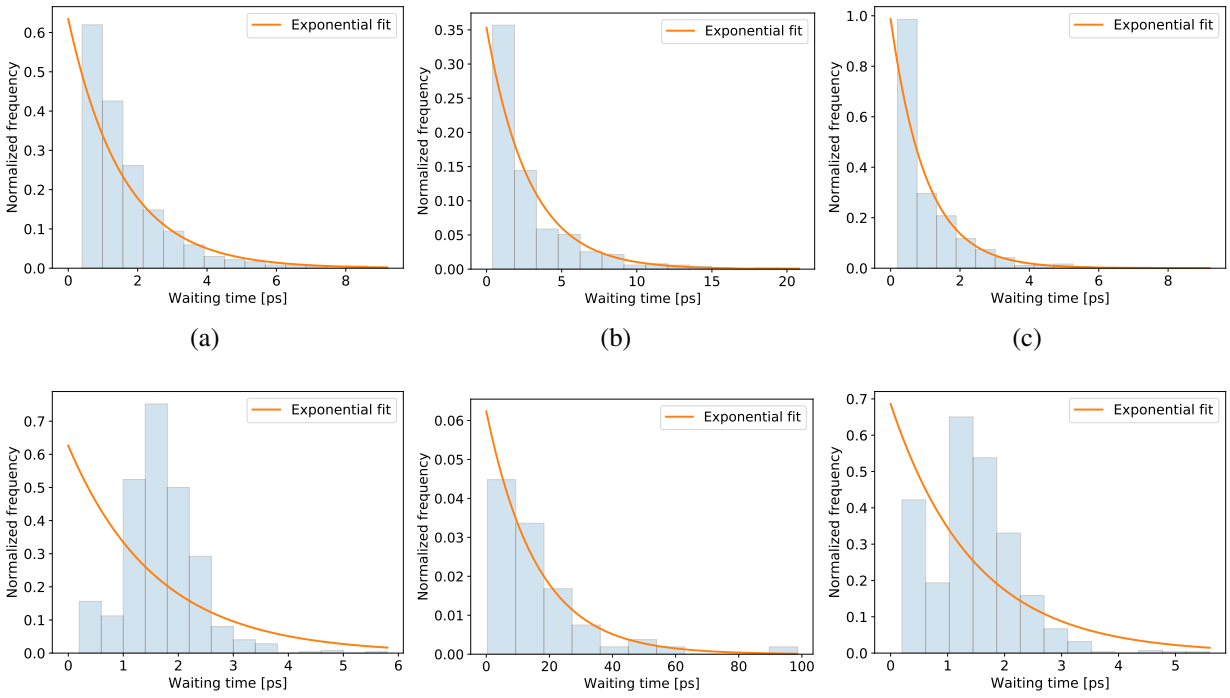


Figure 2.5 Normalized histograms of waiting times between forward (a) and (d), backward (b) and (e), and any jump (c) and (f), along an exponential fitted curve resulted from MLE parameter estimation for $\tau = 25\ MPa$ (top row) and $\tau = 100\ MPa$ (bottom row).

two values of shear stress, $\tau = 25\ MPa$ and $\tau = 100\ MPa$. Observe that distributions of waiting times can be approximated by an exponential decay through its mean value, given the assumption made in the random walk construction.

We also point that for the lower stress (top row), the distribution of backward waiting times, Fig. 2.5 (c), is closer to the forward case, when compared to a higher stress level (bottom row), Fig. 2.5 (f), which is a direct translation of physical effects that occur at the atomic level into a statistical description of dislocation motion. Furthermore, waiting times for backward jumps at $\tau = 100\ MPa$ are longer than at $\tau = 25\ MPa$, since higher stresses hinder the backward dislocation motion.

From the statistical description of waiting times, we compute the rate constants for forward, backward, and total jumps using the expectation of number of events, Eq. (2.10), average waiting times, Eq. (2.13), and the parameter of the exponential fit in Fig. 2.5, obtained by MLE. Again, we

Table 2.3 Rate estimates from MD data for different values of shear stress, using Eq. (2.10), Eq. (2.13) and MLE fit.

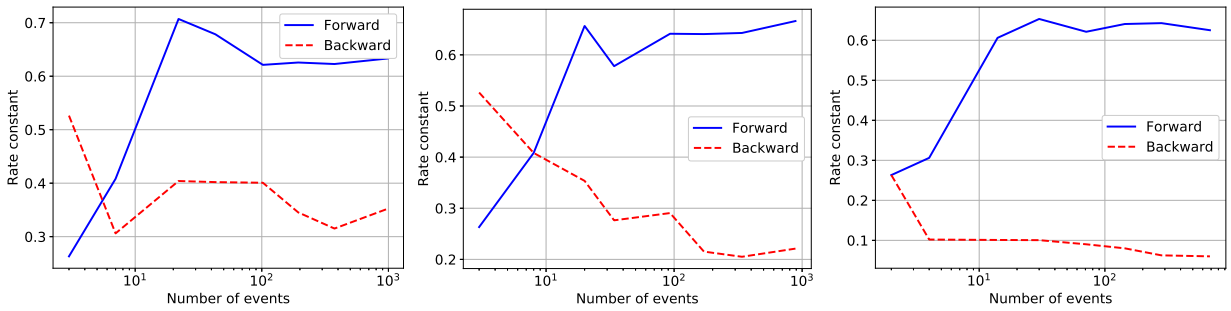
τ	25 MPa			100 MPa		
Method	$\mathbb{E}[N(t)]/t$	$1/\mu$	MLE	$\mathbb{E}[N(t)]/t$	$1/\mu$	MLE
q_f	0.633	0.634	0.634	0.625	0.626	0.626
q_b	0.352	0.353	0.353	0.060	0.062	0.062
Q	0.985	0.987	0.987	0.685	0.686	0.686
$q_f + q_b$	0.985	0.987	0.987	0.685	0.688	0.688
Error (%)	0.00	0.00	0.00	0.00	0.29	0.29

compare results for $\tau = 25\ MPa$ and $\tau = 100\ MPa$, and construct Table 2.3.

Table 2.3 shows the estimates of q_f, q_b, and Q directly. We also compute the quantity $q_f + q_b$ and compare it with total rate Q through a relative error measure. We assume that Q is the reference value since it comes directly from data. We observe that all methods yield nearly identical results, specially for q_f, which has more available data points. For q_b, difference is greater in the $\tau = 100\ MPa$ case due to lower number of data points. We also observe greater error between $q_f + q_b$ and Q for $\tau = 100\ MPs$, for the same reason.

Nevertheless, the three methods are equivalent, and the differences between their results are negligible, so the choice of any particular method yields nearly identical results in the stochastic simulation. The MLE estimate and the $1/\mu$ result are identical, as expected for the exponential fit. The sample mean estimation from $1/\mu$ should converge to the first case, $\mathbb{E}[N(t)]/t$ as $t \to \infty$ or as $N \to \infty$, since the computation of μ involves the summation of waiting times, which will approach the total simulation time when the t or N are large. For simplification purposes, for the remaining simulations we will use the expectation estimate, Eq. (2.10) only due to the agreement between $q_f + q_b$, and total rate Q obtained directly from data points.

We also check the convergence of estimated rates as in the example with manufactured true rates. Here, we do not know the exact rates, therefore we observe the trend of forward and backward rates as we increase the number of observations. Similarly to the manufactured case, each data point in the plot is generated by considering a truncated time-series, until the final data point which includes the whole time series. Fig. 2.6 shows the results of rate estimation, where the x-axis



(a) $\tau = 25\ MPa$. (b) $\tau = 50\ MPa$. (c) $\tau = 100\ MPa$.

Figure 2.6 Convergence in the jump rates from MD time-series data for different stress levels. We observe a more steady and monotonic trend with higher stress levels.

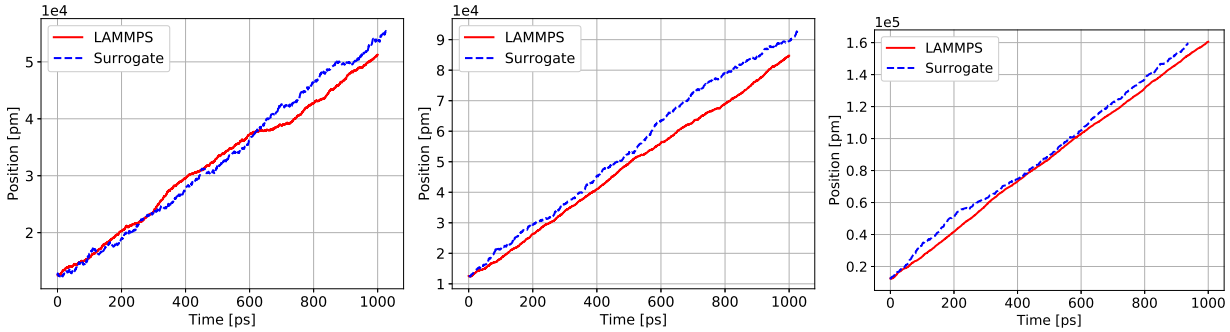
again shows the number of statistical events (total number of jumps $N_t(t)$). We observe that the higher the stress level, the smoother is the curve, which is physically consistent. Higher stresses make the forward rates much larger than the backward rates, and the dislocation movement in the MD simulation flows with less noise, so the rate estimates will tend towards a final value with less oscillations.

2.3.2.2 Surrogate results

For each value of shear stress in the surrogate simulation, we obtain the corresponding rate constants through Eq. (2.10) and simulate the random walk on a ring graph through the KMC framework, Algorithm 2.1. In the end, we are able to plot the distance traveled by the dislocation as a function of time, similar to what is done in MD, by updating the spatial position using Eq. (2.8). We plot the position-time evolution of one realization of the random walk under three different shear stresses, in comparison with the MD results in Fig. 2.7.

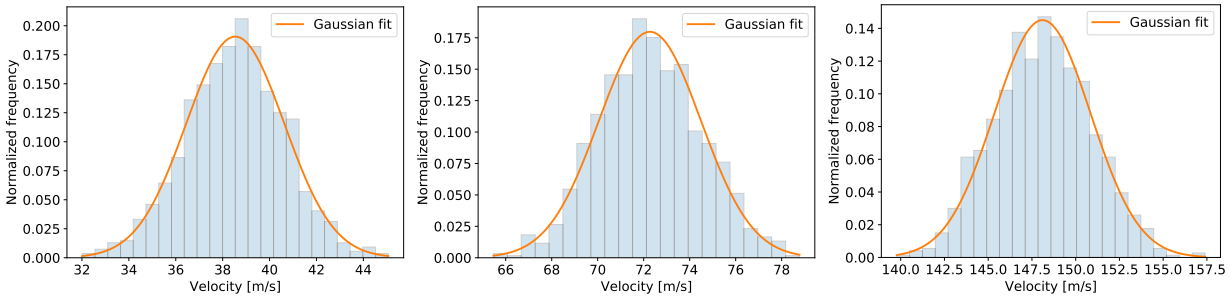
From Fig. 2.7 we make some observations. First, under lower stress, MD results are intrinsically noisy, with the dislocation moving more easily under higher stresses, where the MD plot becomes smoother. Those characteristics are manifested in the rate constants as discussed in Table 2.3 and Fig. 2.6, and in the position *versus* time plots generated from the stochastic process in Fig. 2.7.

We also verify form Fig. 2.7 that the position evolution of the random walk closely follows the same trend as in the original data set. We then compute the dislocation velocity by applying a linear



(a) $\tau = 25\ MPa$. (b) $\tau = 50\ MPa$. (c) $\tau = 100\ MPa$.

Figure 2.7 Position versus time of edge dislocation, comparison between MD results from LAMMPS and one realization of surrogate model through the random walk on a network.



(a) $\tau = 25\ MPa$. (b) $\tau = 50\ MPa$. (c) $\tau = 100\ MPa$.

Figure 2.8 Normalized histograms of velocity estimates from different applied shear stresses. Gaussian fit is plotted after computation of expectation $\mathbb{E}[v]$ and standard deviation $\sigma^2[v]$ from 1000 MC realizations.

regression model to the plots and computing the slope of the linear fit. We repeat this procedure for a large number of realizations, and run a UQ analysis to obtain the statistics of dislocation mobility.

We use a simple MC framework to run several realizations of the surrogate simulation, and we obtain the expectation $\mathbb{E}[v]$, and standard deviation $\sigma^2[v]$ of dislocation velocity under each value of stress. We collect velocity results under $\tau = 25\ MPa$, $\tau = 50\ MPa$, and $\tau = 100\ MPa$, and plot the histograms in Fig. 2.8. Using the estimated values of $\mathbb{E}[v]$ and $\sigma^2[v]$ we approximate a Gaussian to the velocity distributions, closely following the histogram. The agreement between the curve and the histogram comes from the Central Limit Theorem [114], given that the total simulation time of the surrogate is a summation of exponentially distributed random variables X.

We plot the results of velocity as a function of applied stress in Fig. 2.9, where we show the

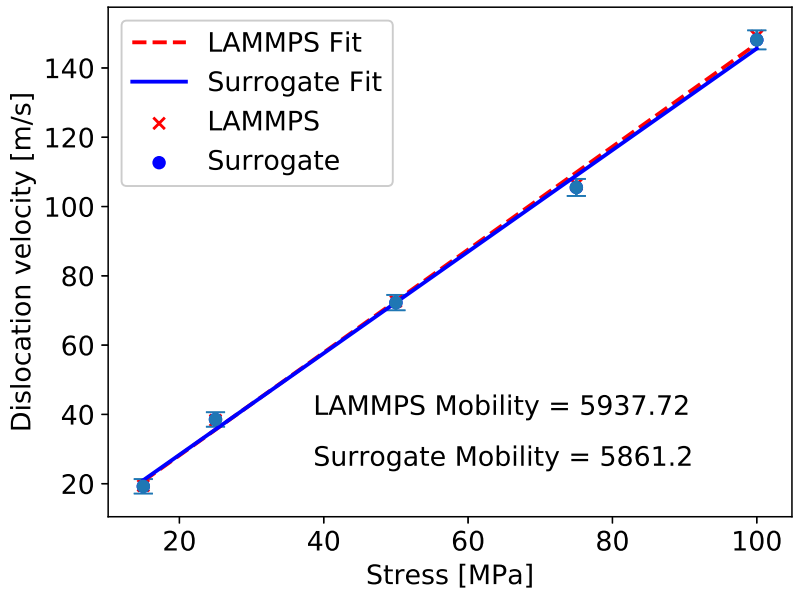


Figure 2.9 Velocity versus stress plot, comparison between MD results of dislocation glide from LAMMPS, and surrogate model simulations using a random walk in a network under two different system temperatures. The surrogate model accurately estimates the mobility with 1.29% relative error.

expected velocity value, and its corresponding uncertainty represented as error bars, for 1000 MC realizations of the surrogate model. We apply a linear regression model to the velocity-stress plot and obtain the mobility M using the linear fit slope m, as in Eq. (2.1).

By introducing the expected velocity with corresponding uncertainty, as in Fig. 2.9, we can propagate the uncertainty to the computation of mobility itself. For the set of 1000 realizations shown in Fig. 2.9, we obtain the corresponding standard deviation for mobility $\sigma_M = 137.27$ $[1/(Pa.s)]$. This is an important contribution of this framework, as it allows a multi-scale propagation of uncertainties related to material properties, starting with the mobility estimate through its modeling as a Poisson process.

2.3.3 Discussion

Through the definition of a KMC algorithm for a random walk defined on a ring graph topology, where the jump rates are computed directly from time-series data of dislocation motion from an MD simulation, we successfully reproduced the stochastic motion of a dislocation glide in a bcc

crystal. The computational advantage of this procedure is two-fold. First, the coarse-graining lumps all the atomic domain information into the network topology, with the dislocation represented as a random walker. The atomistic degrees-of-freedom are condensed into the n nodes that define the graph. Second, we are able to reach the same simulation time faster, which allows for longer time integration, due to the computation of waiting time statistics that feed the KMC algorithm. In the end, 94 hours of one MD simulation with postprocessing at a single stress level turns into an average of 0.45 second surrogate simulation. If we consider the MC estimation of the mobility with 1000 runs at each stress level, the surrogate takes around 50 minutes.

One important aspect is that the physics of dislocation motion is embedded in the time-series data originated from the MD simulation. Therefore, the computation of process rates of forward and backward jumps already takes that into account from the data itself. This is evident, for example, in Fig. 2.6, where the effect of higher stress rates applied to the atomistic structure translates into higher forward jump rates and lower backward rates. Much of the physics of dislocation motion is embedded in the jump rates, and it would be natural to expand this reasoning to other physical features beyond stress. The characterization of process rates in this broader parametric space can then be achieved with the use of State-of-the-art machine learning (ML) algorithms, with MD simulations used as training data, for a more effective and robust upscaling of dislocation properties.

Furthermore, the mobility uncertainty can be propagated to higher scales to be used as an input with associated error, *e.g.*, in DDD simulations. Later, outputs from stochastic DDD may be used to inform lumped-element models of elasto-visco-plasticity, or even phase-field models of failure. Through the use of this surrogate model, we provide a quick and efficient method for propagation of uncertainties across scales, starting form the uncertainty estimation at the atomistic level.

CHAPTER 3

DATA-DRIVEN LEARNING OF CONTINUUM NONLOCAL EQUATIONS FOR DISLOCATION DYNAMICS

3.1 Introduction

Dislocation dynamics is intrinsically connected to plasticity [22] and material failure, emitted from crack tips [115], and piling-up leading to fatigue crack initiation [116]. The long-range interaction of dislocation stress-fields leads to collective motion characterized by avalanches, intermittency, and power-law scaling in energy and velocity distributions [11, 117, 118]. Numerical simulations have successfully reproduced those features from discrete dislocation dynamics (DDD) models [27]. From a continuum perspective, early attempts of proposing evolution laws led to overly phenomenological models [119, 120]. Stochastic approaches have been proposed to account for uncertainties during dislocation motion [121, 122]. Lately, continuum dislocation dynamics (CDD) emerged as another alternative for the continuous modeling of dislocation lines [123, 124], yet still focused on explicit modeling of dislocation-dislocation interactions. A meaningful representation of the collective dynamics of dislocations that highlights the nonlocal, stochastic, and anomalous behavior of dislocation ensembles in a fluid-limit continuous model is still missing. The use of nonlocal vector calculus for continuous modeling of dislocation dynamics is a natural, yet novel alternative.

Nonlocal models present an alternative to classic differential models where discontinuities are allowed, and long-range interactions are naturally present in an integral formulation. These features are attractive in the solution of problems involving convection-diffusion [125, 126], heterogeneous media [127], turbulent flows [128–132], anomalous materials [47], and subsurface dispersion [133, 134]. For more applications, please refer to [135] and references therein. The peridynamic theory [40] was proposed as a nonlocal alternative to classical continuum mechanics of solids, with applicability in fracture problems with discontinuities [136, 137]. Over the last decade, a

formalization of nonlocal models into a nonlocal vector calculus has been extensively discussed [37, 38], along with advances towards the unification of nonlocal/fractional models [39, 138].

With the popularity of Machine Learning (ML) methods, several disciplines have seen increasing applicability of learning algorithms to enhance the understanding of the physics, to learn parameters of a model, or to construct robust surrogates based on high-fidelity data. Data-driven approaches for dislocation dynamics have lately acquired more interest. In [139], authors used two-dimensional DDD simulations to train an algorithm for prediction of stress-strain curves. A ML approach for prediction of material properties from dislocation pile-ups was presented in [140]. Classification algorithms have also been used in the context of dislocation micro-structures [141, 142]. Data-driven surrogate modeling of dislocation glide for computation of mobility estimates with uncertainty was proposed in [143]. Other ML approaches have also grown in the context of learning the physics in the form of PDEs. We note the contributions of Physics-Informed Neural Networks (PINNs) [49] which enhances deep neural networks with physics-based constraints, and PDE discovery approaches through the use of candidate terms and operators [144–147].

The problem of learning kernels in integral operators has gained attention over the last years, with major contributions in the context of homogenization via nonlocal modeling and, more generally, in nonlocal and fractional diffusion diffusion. In one front, nPINNs [48] was introduced as the nonlocal counterpart of the PINNs framework. Here, nonlocal equations are incorporated as constraints while training a deep neural network, for both forward and inverse problems with power-law kernel and finite horizon. The extraction of more complex kernels was investigated, via an operator regression approach, in [51], allowing the possibility of sign-changing kernels by representing the kernel function through a polynomial expansion. This approach was further used in diverse applications such as peridynamics [50], constitutive laws [52], coarse-graining of molecular dynamics simulations [53], and homogenization of subsurface transport through heterogeneous media [134].

In the present work, we use two-dimensional DDD simulations to generate probability distribution functions from shifted dislocation positions obtained from numerous realizations of the DDD

problem. This approach gives us directly the Lagrangian dynamics of dislocation position. We transform the particle dynamics into a continuum Probability Density Function (PDF) evolving over time through an Adaptive Kernel Density Estimation method, generating a time-series of dislocation position PDFs. We propose a nonlocal model defined through a kernel-based integral operator for the evolution of the PDFs as the fluid-limit of the underlying stochastic process, and develop a ML framework to parameterize the nonlocal kernel, learning from the PDF snapshots generated from DDD data.

We summarize our main contributions below.

- We obtain the probabilistic particle dynamics directly from DDD simulations. We highlight the effect of external loading and multiplication in the final probability distribution of dislocation position. Such differences are not evident in velocity distributions.

- We propose a general nonlocal equation to model the evolution of dislocation probability distributions in space, establishing the link between the discrete nature of dislocation dynamics at the mesoscale and the origin of its corresponding nonlocal operator at the continuum scale.

- We develop a ML framework to solve the inverse problem of recovering the parameters of the nonlocal equation from high-fidelity data. Specifically, we feed PDFs obtained from DDD simulations into the ML algorithm and obtain the parameters of the nonlocal power-law kernel, in terms of the fractional order α, horizon δ, and a linear coefficient.

This work establishes, for the first time, a systematic, direct connection between the discrete anomalous dynamics of dislocations at the mesoscale to their ultimate effect in a continuum sense. With this mindset, we obtain a fast alternative to simulate dislocation dynamics through the nonlocal surrogate model, while still maintaining the underlying physics of micro-structural processes. We focus on the collective motion of dislocations as an indication of microstructural evolution and adopt the position PDFs as its measure. When we discover the governing equations for the PDFs, we have a more efficient and fast evaluation of microstructure evolution compared compared to

high-fodelity DDD. This leads to a more efficient connection to macroscale problems such as visco-elasticity [47] and fracture [104, 105].

We also establish an efficient framework for the parameterization of nonlocal kernels, overcoming limitations of existing methods. Existing learning approaches have the disadvantage of minimization in a high-dimensional space [51] with gradient-based optimization. The present work overcome these challenges by defining a bi-level learning algorithm. Given the linearity of our operator, we separate the learning of coefficients from the nonlocal kernel parameterization. In the first level, trial pairs of kernel parameters are used to obtain the best coefficients through a Least-Squares algorithm. The minimization occurs at the second level, where we define a minimization problem for the kernel parameters only, restricting the optimization to two dimensions and using a gradient-free search method.

3.2 Two-Dimensional Discrete Dislocation Dynamics

The simplified setup of a two-dimensional simulation, although lacking the curvature and natural multiplication mechanisms that a three-dimensional simulation provides, is still a robust and efficient way of observing the collective interactions of dislocation populations in a controlled manner. It allows the extraction of important quantities of interest, such as velocity distributions, stress and plastic strain evolution, and has been adopted in the literature to understand dislocation avalanches and power spectrum time-signals [11, 139]. Therefore, here we adopt a two-dimensional discrete dislocation approach with the goal of learning the main characteristics of collective dislocation dynamics as a first step to translate such effects into a nonlocal continuum model. Particular implementation details will be explained in each section where necessary.

We consider a two-dimensional square domain of size L, populated with straight edge dislocations with directions along the z direction, each of them with a Burgers direction $\boldsymbol{b} = \pm b$ along x, assuming a single-slip system. We assume there is no climb mechanism, so dislocations may only move in the x direction.

Immersed in an elastic continuum medium, dislocations create a long-range stress field, such

that each dislocation is affected by the presence of all other dislocations in the crystal through an interaction stress, as well as any external stress σ_{ext}. Given the distance between dislocations $\boldsymbol{r}$, the dislocation-dislocation interaction stress, $\sigma_i(\boldsymbol{r})$, is given by [148]

$$\sigma_i(\boldsymbol{r}) = \frac{\mu b}{2\pi(1-\mu)} \frac{x(x^2-y^2)}{(x^2+y^2)^2}, \tag{3.1}$$

where x and y represent the distances between the edge dislocations in the x and y directions, respectively, μ is the shear modulus, and ν is the Poisson ratio.

We simulate a domain in the bulk of the material, and assume it to be sufficiently far from any free surface, therefore Periodic Boundary Conditions (PBC) are needed. In order to apply the PBC and take into account all the long-range interactions, we include the forces due to infinite images of the simulation box. The exact form of the interaction stress is [149]

$$\sigma_i(\boldsymbol{r}) = \frac{\mu b}{2(1-\mu)} \frac{1}{L} \frac{\sin(X)\,[\cosh(Y) - \cos(X) - Y\sinh(Y)]}{[\cosh(Y) - \cos(X)]^2}, \tag{3.2}$$

where $X = \frac{2\pi x}{L}$ and $Y = \frac{2\pi y}{L}$.

Then, under the assumption that dislocation motion is overdamped under the *viscous drag* regime, the equation of motion for the $i-$th dislocation along the x direction, from the single-slip and no climb assumption, is

$$\frac{1}{M}\frac{dx_i}{dt} = b_i \left(\sum_{m \neq i}^{N} \sigma_i(\boldsymbol{r}_m - \boldsymbol{r}_i) + \sigma_{ext} \right), \tag{3.3}$$

where M is the dislocation mobility. All stress definitions refer to the shear stress component τ_{xy}, such that the combination of σ_{ext} and σ_i result in the resolved shear stress acting on the dislocation. The resolved shear stress is the effective driver of motion in the edge dislocation.

We can solve this equation at first using a Forward-Euler scheme. For simplicity, we rescale the units and solve the problem with length in units of b, stress in units of $\sigma_0 = \frac{\mu}{2\pi(1-\nu)}$, and time in units of $t_0 = \frac{1}{M\sigma_0}$.

The plastic strain resulting from dislocation motion can be computed following Orowan's relation

$$\gamma = \frac{1}{L^2} \sum_{i=1}^{N} b_i \Delta x_i. \tag{3.4}$$

Beyond the constitutive relations that govern the dislocation glide velocity due to interactions and external stress, Eq.(3.3), two-dimensional discrete dislocation dynamics simulations also need to consider other phenomenological aspects such as annihilation and multiplication.

Results from linear elasticity become invalid near the dislocation core due to nonlinearity in the stress field. Therefore, when two dislocations of opposite Burgers vector are within a distance d_a, they annihilated each other and are removed from the simulation.

Dislocation multiplication does not occur naturally in 2D simulations, as compared to 3D-DDD. In order to mimic the bowing of dislocation curves due to Frank-Read sources, we need to consider a phenomenological model. Here we follow the procedure in [149], where we distribute N_s dislocation sources randomly in the domain. At each time-step, we check at the sources the resulting shear stress, and compare it to a critical stress τ_c. If the stress is above τ_c for more than t_{nuc} time-steps, we generate a pair of dislocations with distance L_{nuc}, such that, in scaled units,

$$L_{nuc} = \frac{1}{\tau_c}. \tag{3.5}$$

3.2.1 Representative Example: Single Crystal Under Creep

We simulate three examples of single crystals under creep loading, following the setup from [11]. We consider a domain of size $L = 300\ b$, initial number of dislocations of $N_0 = 1500$, and annihilation with a critical distance of $d_a = 2\ b$. We distribute 20 dislocation sources randomly throughout the domain, with mean critical nucleation time $\bar{t}_{nuc} = 10$, and critical distance with mean $\bar{L}_{nuc} = 50$, both parameters with variance of 10% $\bar{L}_{nuc}$.

We test three representative examples to understand the effect of dislocation sources and external load in the form and parameterization of the final nonlocal kernel, where the external shear load is expressed using the definition of rescaled stress units, $\sigma_0 = \frac{\mu}{2\pi(1-\nu)}$.

- **Case 1:** load of $\sigma_{ext} = 0.0125\ \sigma_0$ without dislocation multiplication.
- **Case 2:** load of $\sigma_{ext} = 0.0125\ \sigma_0$ with dislocation multiplication.

- **Case 3:** load of $\sigma_{ext} = 0.0250\ \sigma_0$ with dislocation multiplication.

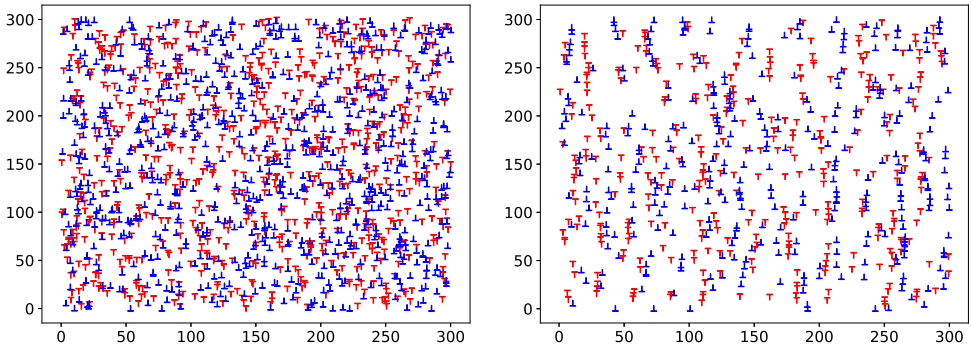
Case 1 represents locations inside the mechanical part without the presence of imperfections, impurities, or microcracks such that dislocations that are present inside the material do not multiply, and hence just glide until occasional annihilation. Therefore, Case 1 is representative of regions with lower internal stresses, less intense dislocation activity and plastic flow, and no evident rapid failure processes.

Conversely, Cases 2 and 3 contain dislocation multiplication sources in a phenomenological way, representing regions in a component where we would normally observe higher degradation, under the presence of microcracks, voids, impurities and rough surfaces. Those characteristics are natural dislocation generators and are typically associated with failure regions. Therefore, in Cases 2 and 3 we are observing what happens near failure-inducing locations.

The DDD simulations are executed with an in-house Python code running on Intel Xeon Gold 6148 CPUs with 2.40GHz. In all cases, we first let the system relax for 10000 time-steps of size $\Delta t = 1\ t_0$ with no external stress. This procedure leads to intense activity and annihilations until the dislocations reach a meta-stable configuration with about half the number of original dislocations. Then, we apply σ_{ext} with a time-step of $\Delta t = 0.01\ t_0$ until final time of $T = 30000$ for Case 1, and $T = 25000$ for Cases 2 and 3.

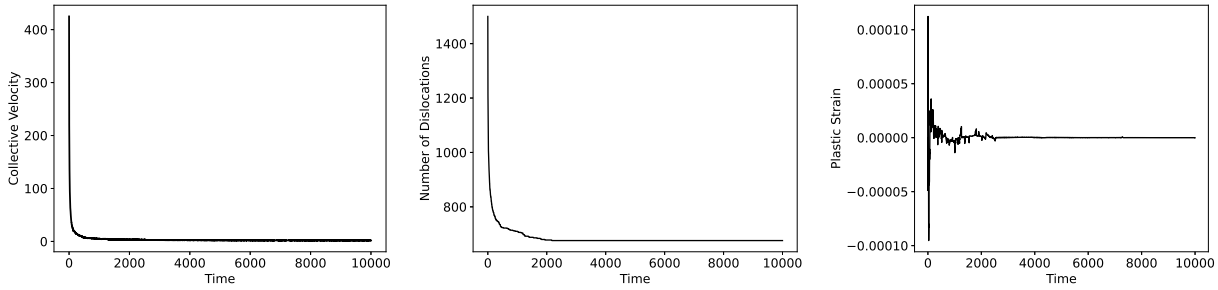
Fig. 3.1 shows one realization of an initial dislocation configuration and the relaxed metastable configuration for Case 1. The time-series plots of the collective velocity $V = \frac{\sum_i v_i}{N}$, number of dislocations, and plastic strain during the relaxation steps are shown in Fig. 3.2.

The collective statistics for the representative realizations discussed in this Section can be seen in Fig. 3.3. We plot the collective velocity, number of dislocations, and accumulated plastic strain during the creep load. The collective velocity signal is intermittent for Cases 2 and 3 with multiplication, where the spikes indicate bursts of activities during an avalanche. The higher load of Case 3 makes the baseline collective velocity be higher than Case 2, yet the spikes of Case 3 are not as large, since dislocations will tend to move faster, therefore having less time to interact close to the critical regime, as we see in Case 2. The higher baseline velocity also affects the accumulated



(a) Initial configuration. (b) Relaxed configuration.

Figure 3.1 Dislocation distribution at the beginning of the simulation (a), and after the relaxation (b) in a metastable structure for Case 1. Red and blue markers correspond to dislocations with positive and negative Burgers, respectively.



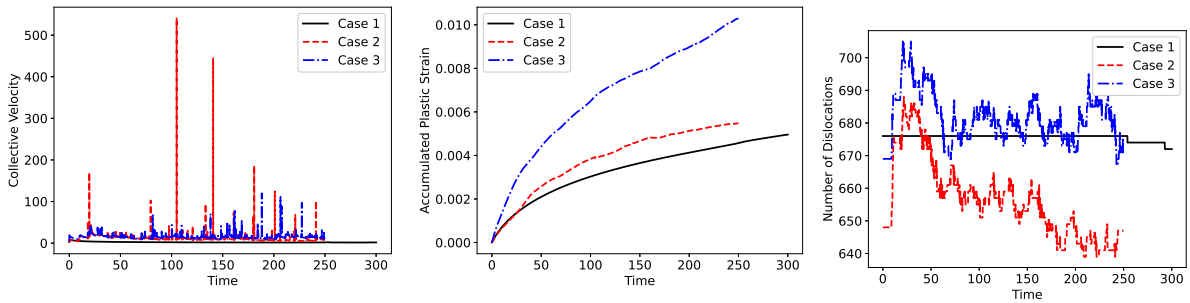
(a) Collective velocity. (b) Number of dislocations. (c) Plastic strain $\gamma(t)$.

Figure 3.2 Time-series plots of collective velocity, number of dislocations in the system, and plastic strain during the relaxation steps to show the system's stabilization.

plastic strain that is larger for Case 3. The number of dislocations is almost stable for Case 1, with Cases 2 and 3 showing more oscillations. This is due to the rearrangements that occur after a new dislocation pair is introduced, which eventually leads to more annihilations than Case 1.

Last, we investigate the velocity statistics from the DDD simulations. Fig. 3.4 shows the PDF of individual dislocation velocity statistics collected throughout the whole simulation time for the single DDD realization of each case discussed in this section. We find that, in accordance with [11], the velocity PDFs show a power-law decay in the form $v \propto \sigma^{-\beta}$ with exponent around $\beta = 2.4$ for Cases 2 and 3 with multiplication. For Case 1, we see that the decay is slightly sharper.

The velocity PDF has been extensively studied both experimentally and numerically over the past



(a) Collective velocity. (b) Accumulated plastic strain. (c) Number of dislocations.

Figure 3.3 Time-series plots of collective velocity, accumulated plastic strain, and number of dislocations in the system for a single realization of creep test for all three cases.

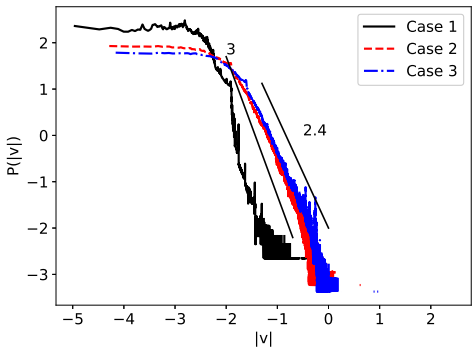


Figure 3.4 Probability Density Function of dislocation velocity for Cases 1, 2, and 3. We observe a power-law scaling of order $\beta = 2.4$ for Cases 2 and 3 with multiplication, and a sharper decay for Case 1.

years, and strongly suggests that the nature of dislocation dynamics has anomalous characteristics. The power-law exponent of $\beta = 2.4$ is directly associated to intermittent velocity signals typical of avalanches and super-diffusive behavior. However, dislocation dynamics simulations, whether discrete or continuous are still too expensive to be run in the long time with the goal of understanding how the dislocation-network anomalous behavior influences the failure processes at the macroscale.

One could look at dislocation dynamics from the perspective of particle dynamics, where the dislocations would move under a underlying stochastic process. Ideally, we would analyze the statistics and construct a stochastic process that governs such particle dynamics, as a way of generating infinitely many particle trajectories that lead to the fluid-limit dynamics of the process. However, in the case of dislocation dynamics, even though the velocity distributions give us an idea

of the type of stochastic behavior due to the heavy tails with power-law decay, the process cannot be simply described. Some dislocations are stuck, others jiggle around an equilibrium state, and a few others move intermittently with large velocity in a highly correlated motion.

Therefore, in order to obtain the fluid-limit dynamics, we still need to obtain statistics from a sufficient number of dislocation particles. The steps we take to this end will be discussed in the next section.

3.3 Data Generation

In this section we describe the methodology to obtain empirical PDFs directly from DDD simulations, without the construction of a stochastic process, as discussed before. On the one hand, this makes the data generation process expensive, as it relies on simulation of multiple DDD simulations, instead of cheaper stochastic process trajectories. On other hand, this approach benefits from directly using high-fidelity dislocation data and is the most accurate representation of the dynamics we could generate.

3.3.1 Obtaining Data of Shifted Positions

We start by defining the shifted position $X_i(t)$ of a dislocation i in a single realization of the DDD simulation. Given the initial position in the absolute frame of reference of the DDD box, $x_i(0)$, and the current absolute position in the simulation frame of reference, $x_i(t)$, the shifted position is a measure of the relative displacement of dislocation i with respect to its initial position:

$$X_i(t) = x_i(t) - x_i(0), \quad \text{for} \quad t \in (0, T]. \tag{3.6}$$

We obtain a statistically significant collection of data-points $X_i(t)$ by considering the DDD simulation as stochastic. We run $n_r = 2000$ DDD simulations, each with random initial positions of dislocations and multiplication sources (for Cases 2 and 3). We distribute the execution of realizations among several HPC cores to take advantage of embarrassingly parallel stochastic simulations. With the number of dislocations after the relaxation between 600 and 700, the

compilation of all dislocation shifted positions across the n_r realizations gives the trajectories of 10^6 Lagrangian particles that move following an underlying stochastic process starting at $X_i(0) = 0$.

We take the Lagrangian particle trajectories obtained directly from DDD and translate them into an evolving PDF $\hat{p}(x,t)$, defined as the probability to find a dislocation at a distance x from its initial position from the start of the DDD simulation. In the end, we want to construct a model for the time-evolution of $\hat{p}(x,t)$, and we propose that its evolution is governed by an integral operator that we model as a nonlocal Laplacian. In the following, we discuss how to transform the DDD position data into density estimates that will be used as training data for the ML algorithm.

3.3.2 Density Estimation

Under the proposed nonlocal model for evolution of dislocation position PDF, we are most interested in the nature of the dynamics of dislocation particles, i.e., how they react when put under creep stress along with an ever-changing stress landscape due to addition and removal of other dislocations. Given the focus on the dynamics due to load, multiplication, and annihilation mechanisms in a broader sense, in the limit of infinitely many particles, and not attempting to model the creation and destruction themselves, we do not include the birth/death in the nonlocal formulation. Instead, we only describe the nature of the dislocation motion from the continuum perspective, as a consequence of those mechanisms from the discrete representation at the DDD level.

In this sense, the annihilation and creation of dislocations in DDD induce a level of noise when considering a continuous PDF representation. In the creep regime, we can minimize the interference of such noise by selecting a time domain over which the initial burst of dislocation multiplications and motion due to the rapid increase in stress up to the creep level has reached a steady-state regime. For training and testing of the ML framework, we select the last 10000 time-steps from the DDD simulations to generate the PDFs, as to minimize the effect of applying the creep load, and to obtain a data-set with the least changes in the number of particles. In the time-series selected, we observe only 0.1%, 0.69%, and 0.92% in relative difference between final and initial number of dislocations in the selected data for Cases 1, 2, and 3, respectively. We can then assume a conservative nonlocal

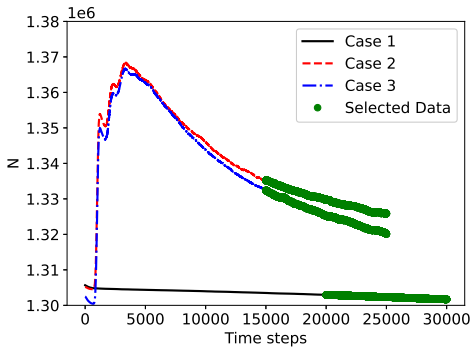


Figure 3.5 Time-series of the total number of dislocations across the $n_r = 2000$ realizations of DDD for Cases 1, 2, and 3. We highlight the selected data for training and testing the ML algorithm.

transport model in the continuum, yet annihilation, multiplication, and external load will directly influence the shape and parameterization of the nonlocal operator. We highlight the specific range of selected data over the whole time-series generated from DDD in Fig 3.5.

Before applying the estimator, we select the central 99.99% of the probability mass, therefore discarding the outermost particles at each time-step. This procedure clearly defines a compact support for the PDF.

We quickly summarize the classical and Adaptive Kernel Density Estimation formulations below.

3.3.2.1 Kernel Density Estimation (KDE)

KDE are non-parametric estimators that do not require assumptions on the form of the sampling distribution. Yet, to use KDE, we need to apply a specific kernel $k(x - x_i; w_0)$ parameterized by a bandwidth w_0. At this stage, we obtain an initial density estimate $\hat{p}_0(x)$ from

$$\hat{p}_0(x) = \frac{1}{n} \sum_{i=1}^{n} k(x - x_i; w_0), \tag{3.7}$$

where x is the coordinate over which we wish to evaluate the PDF, and x_i are the positions of data points $i = 1, 2, \ldots, n$.

Kernels are normalized to unity, i.e.,

$$\int_{-\infty}^{\infty} k(x; w_0) dx = 1, \tag{3.8}$$

and take the form

$$k(x - x_i; w_0) = \frac{1}{w_0} K\left(\frac{x - x_i}{w_0}\right). \tag{3.9}$$

Then, the final density estimator is

$$\hat{p}_0(x) = \frac{1}{n w_0} \sum_{i=1}^{n} K\left(\frac{x - x_i}{w_0}\right). \tag{3.10}$$

Let s be the sample standard deviation. In the limit of number of data-points $n \to \infty$ of normally distributed data, the Mean Integrated Square Error of $\hat{p}_0$ is minimized when [150]

$$w_0 = 1.06 s n^{-1/5}. \tag{3.11}$$

3.3.2.2 Adaptive Kernel Density Estimation (AKDE)

Given the large jumps observed in DDD, we can expect the PDFs to have heavy tails, yet with limited data, while the majority of the mass would fall into the central part. A uniform binning method such as KDE would lead to the occurrence of noise in the tails, which we need to avoid as those curves feed the ML algorithm. Since higher density regions need narrower bins than low density tails, we use AKDE to obtain a continuous, smooth function.

We run the AKDE starting from an initial classical KDE estimate $\hat{p}_0(x)$ with a fixed bandwidth w_0 from Eq. (3.11). Then, the adaptivity takes place in the variable bandwidth for each data point [150]:

$$w_i = w_0 \lambda_i = w_0 \left(\frac{\hat{p}_0(X_i)}{G}\right)^{-\xi}, \tag{3.12}$$

where G is defined as

$$G = \exp\left(\frac{1}{n} \sum_{i=1}^{n} \ln \hat{p}_0(X_i)\right). \tag{3.13}$$

Furthermore, $0 \leq \xi \leq 1$ is a sensitivity parameter that controls how important the shape of the initial guess is with respect to the second estimation [151]. A theoretical optimal value was found to be $\xi = 0.5$ [152]. We finally obtain the density from AKDE as

$$\hat{p}_1(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{w_i} K\left(\frac{x - x_i}{w_i}\right). \tag{3.14}$$

We define a domain Ω for the density function by adding a band of zeros beyond the right and left-most points in the x direction, providing some space between the compact support of the PDF and the nonlocal simulation domain. We compute the estimates $\hat{p}_1$ at equally spaced points inside Ω defining a fixed grid size of $h = 0.05$. In Case 1, we have $\Omega_1 = [-15, 15]$, and $m = 601$ points. Cases 2 and 3 are defined with $\Omega_2 = \Omega_3 = [-40, 40]$ leading to $m = 1601$ points. Computation of $\hat{p}_1(x)$ for each time-step is also executed in parallel, each core processing a distinct time-step, for a total of 1000 HPC cores used for each case.

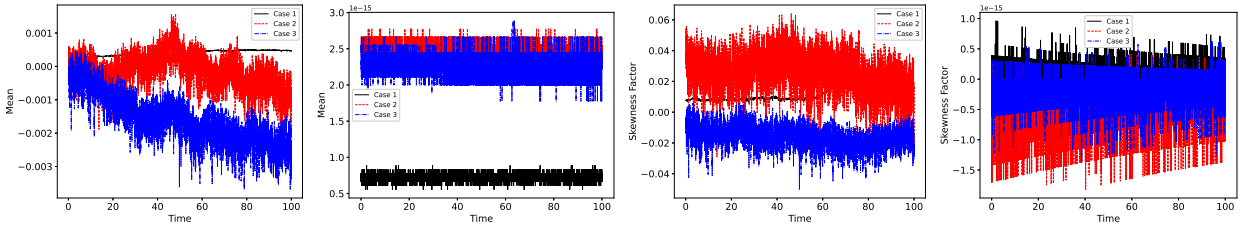
Finally, once the final estimates $\hat{p}_1(x)$ are obtained for all time-steps, in the last operation we enforce symmetry, as to avoid any inconsistencies in the ML algorithm, as we adopt a symmetric, radial-basis nonlocal kernel in the definition of our operator. We apply the symmetrization by checking the evolution of mean μ and skewness factor $\tilde{\mu}_3$ from $\hat{p}_1(x)$, defined as

$$\mu = \mathbb{E}[\hat{p}_1(x)] = \int \hat{p}_1(x) x dx, \tag{3.15}$$

$$\tilde{\mu}_3 = \mathbb{E}\left[\left(\frac{\hat{p}_1(x) - \mu}{\sigma}\right)^3\right] = \frac{\int (x-\mu)^3 \hat{p}_1(x) dx}{\left(\sqrt{\int (x-\mu)^2 \hat{p}_1(x) dx}\right)^3}. \tag{3.16}$$

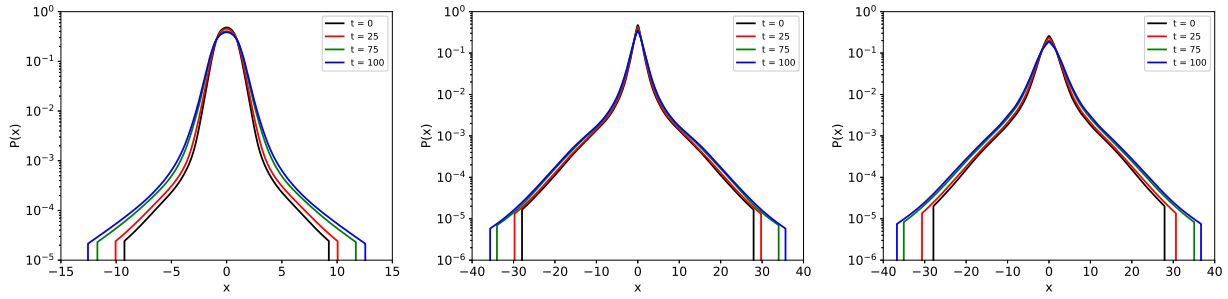
We initially verify that they are sufficiently close to zero, assuming any non-zero measure of those parameters are due to lack of sufficient data. Then, we take the left side of $\hat{p}_1(x)$, mirror it to the other side, and re-normalize the PDF. We verify that after this procedure we guarantee a symmetric PDF that will better fit the radial-basis nonlocal kernel. Fig. 3.6 shows the values of $\mu(t)$ and $\tilde{\mu}_3(t)$ before and after the symmetrization.

Fig. 3.7 shows snapshots of the final PDF estimate for all three cases at selected time-steps. We plot the PDFs in logarithmic scale in the y direction: we can readily see that the PDFs do not decay fast as one would expect from a Gaussian process. Instead, we have power-law decaying tails, with seemingly heavier ends on Cases 2 and 3 where multiplication mechanisms activate the collective dynamics more intensively. The heaviness on tails is accompanied by a larger support of the PDF. As we will see in the following sections, when we feed the PDFs into the learning algorithm, the kernel of the nonlocal operator will reflect those differences, establishing a meaningful link between the discrete and continuous dynamics.



(a) (b) (c) (d)

Figure 3.6 Evolution of mean and skewness factor of $\hat{p}_1(x)$. The evolution of mean (a) before, and (b) after the symmetrization and re-normalization. The evolution of skewness factor (c) before, and (d) after symmetrization and re-normalization.



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 3.7 Final shape of dislocation shifted position PDF from AKDE with symmetrization and re-normalization at selected time-steps.

3.4 Nonlocal Transport Models

For the evolution of the dislocation position PDFs, we propose a parabolic nonlocal transport model defined through a nonlocal operator characterized by a kernel function, that we aim to determine. We use the data from DDD simulations to train a machine-learned surrogate model, for which we identify the model parameters.

We let $p(x,t)$ represent the empirical PDF estimate at time $t \in [0,T]$, and x denote the position in over the domain $\Omega = [-L,L]$, where L is defined by the taking the maximum support at the last time-step and including extra zeros, as seen in Fig. 3.7 . We model the evolution of p following the nonlocal parabolic equation

$$\begin{cases} \dot{p}(x,t) = \mathcal{L}p, & x \in \Omega \\ \mathcal{B}_{\mathcal{I}}p(x) = g(x), & x \in \Omega_{\mathcal{I}}, \end{cases} \tag{3.17}$$

where $\mathcal{L}$ denotes the nonlocal (linear) Laplacian operator defined as

$$\mathcal{L}p = \int_{B_\delta(x)} K(|y-x|)(p(y)-p(x))dy. \tag{3.18}$$

$B_\delta(x)$ represents the ball centered at x of radius δ, also called the *horizon*, defining the compact support of $\mathcal{L}$. It is relevant to note that for specific choices of kernel functions, $\mathcal{L}$ corresponds to well-known operators such as the fractional Laplacian [39, 138]. In fact, when $K(|y-x|) \propto |y-x|^{-\alpha}$, with $\alpha = 1+2s$, $s \in (0,1)$, the operator $\mathcal{L}$ corresponds to the one-dimensional fractional Laplacian. Furthermore, when the same kernel is restricted to the compact support $B_\delta(x)$, $\mathcal{L}$ corresponds to the so-called truncated fractional Laplacian. The latter turns out to be the operator of choice in our framework.

The interaction domain where nonlocal boundary conditions (or volume constraints) are prescribed is defined as:

$$\Omega_I = \{y \in \mathbb{R} \setminus \Omega \text{ such that } |y-x| < \delta \text{ for some } x \in \Omega\}. \tag{3.19}$$

We prescribe nonlocal homogeneous Dirichlet volume constraints, given (in one dimension) by the nonlocal interaction operator $B_I : [-L-\delta, -L) \bigcup (L, L+\delta] \rightarrow \mathbb{R}$, such that $g(x) = 0$ at $x \in \Omega_I$.

The objective of the proposed ML algorithm is to train the nonlocal model on the basis of the series of PDFs; specifically, we find the best form and parameters of the kernel $K(|y-x|)$ such that Eqs. (3.17,3.18) are satisfied and the predicted distributions are as close as possible to the high-fidelity dataset.

3.5 Machine Learning of Nonlocal Kernels for Dislocation Dynamics

3.5.1 A Bi-level Machine Learning Framework

We start the approximation by assuming that the kernel $K(|y-x|)$ is a radial function compactly supported on $B_\delta(x)$, decaying with $\alpha-$th order power-law, multiplied by a $P(|y-x|)$ function defined over $[0, \delta]$

$$K(|y-x|) = D\,\frac{P(|y-x|)}{|y-x|^\alpha}, \tag{3.20}$$

where we assume the coefficient $D \in \mathbb{R}$, $D > 0$. The form of the function P is part of the learning problem and its form strongly depends on the underlying physical system we want to reproduce. In the literature [51–53], the choice of a linear combination of Bernstein polynomials has been particularly successful. However, for the application considered in this work, the employment of Bernstein polynomials does not increase the surrogate's prediction power. For these reasons, we consider the simplified case of $P(|y - x|) = 1$, for which the resulting operator corresponds to a truncated fractional Laplacian. Thus, the learning problem consists of finding the parameters α, δ, and the coefficient D that parameterize the kernel.

We adopt a bi-level learning approach to reduce the dimensions of the minimization problem by exploiting the linearity of the nonlocal operator. Level 1 consists in obtaining the best coefficient D for a given pair of parameters α and δ, while at Level 2 the algorithm iterates over different values of α and δ and minimizes a cost function, each iteration using the optimal D found in Level 1.

For the numerical solution of the bi-level optimization problem, we rewrite the nonlocal transport model, Eq. (3.17), in a semi-discrete manner using a meshless approach, i.e.

$$\dot{p}(x_i, t) = \sum_{j \in \mathcal{H}} K(|x_j - x_i|)(p(x_j) - p(x_i))h, \tag{3.21}$$

where H is the family of points x_j in the neighborhood of point x_i, and h is the distance between the points.

By using the power-law definition of the kernel in (3.20), with $P = 1$, we can write the equation as

$$\dot{p}(x_i, t) = \sum_{j \in \mathcal{H}} \frac{D}{|x_j - x_i|^{\alpha}}(p(x_j) - p(x_i))h. \tag{3.22}$$

3.5.1.1 Level 1

We adapt the ideas presented in [146] for discovery of PDEs, yet, instead of identification of different PDE terms, our goal is to use the linear structure of Eq. (3.22) to obtain the coefficient D given a specific pair of values (α, δ).

For given values of δ and α, we construct vectors U, and U_t. U contains the RHS of Eq. (3.22), where the spatio-temporal data are reshaped into a single stacked column array. U_t is the LHS of Eq. (3.22) with the time-derivative computed through a forward Euler method at all space and time points, also transformed into a single column array. Given n time-steps and m grid points, both U and U_t have size nm.

Our problem $U_t = UD$ reads:

$$\begin{bmatrix} \dot{p}(x_0, t_0) \\ \dot{p}(x_1, t_0) \\ \dot{p}(x_2, t_0) \\ \vdots \\ \dot{p}(x_{m-1}, t_n) \\ \dot{p}(x_m, t_n) \end{bmatrix} = D \begin{bmatrix} C(x_0, t_0) \\ C(x_1, t_0) \\ C(x_2, t_0) \\ \vdots \\ C(x_{m-1}, t_n) \\ C(x_m, t_n) \end{bmatrix} \tag{3.23}$$

with

$$C(x_i, t_k) = \sum_{j \in \mathcal{H}} \frac{1}{|x_j(t_k) - x_i(t_k)|^\alpha} (p(x_j(t_k)) - p(x_i(t_k))h. \tag{3.24}$$

Then, for every pair of α, δ being considered in the minimization, we use a Least Squares solver to obtain the best D.

3.5.1.2 Level 2

We use a minimization algorithm to find α and δ that minimize the Mean Logarithmic Absolute Error (MLAE):

$$\epsilon = \frac{1}{nm} \sum_{l=1}^{nm} |\log(p_l) - \log(\tilde{p}_l)|, \tag{3.25}$$

where p_l represents the true value of the function at a particular x ant t, and $\tilde{p}_l$ is the solution of the nonlocal model at x and t starting from the initial conditions at $t = 0$, for the current trial values α_{trial} and δ_{trial}, and its corresponding D. We adopt the MLAE with the goal of giving as much significance to the information on the tails as we give to the central part of the PDF.

Algorithm 3.1 Bi-Level Machine Learning Algorithm with Nelder-Mead Minimization.

1: Choose the initial guess (α_0, σ_0).
2: **for** Each iteration i of NM (Level 2) **do**
3: Construct the matrix equation Eq. (3.23) and obtain the coefficient D with trial parameters (α_i, σ_i) (Level 1).
4: Solve the nonlocal model and obtain the trial solutions $\tilde{p}$ using Eq. (3.26).
5: Using the true p and trial solutions $\tilde{p}$, Compute the error using Eq. (3.25).
6: **end for**
7: The algorithm gives the optimal parameters $(\alpha_{\text{opt}}, \sigma_{\text{opt}})$ and associated D_{opt}.

For all time-steps, we obtain $\tilde{p}$ at time-step $k+1$ and grid-point i, $\tilde{p}_i^{k+1}$, from $\tilde{p}_i^k$, using the forward Euler scheme. Thus, Eq. (3.22) becomes

$$\tilde{p}_i^{k+1} = \tilde{p}_i^k + \Delta t \sum_{j \in \mathcal{H}} \frac{D}{|x_j - x_i|^\alpha} (\tilde{p}_j^k - \tilde{p}_i^k) h. \tag{3.26}$$

For the solution of the minimization problem we adopt, among several possible choices, the Nelder-Mead Method (NM), which is a gradient-free, downhill simplex approach that uses a direct search method (based on function comparison). The overall bi-level algorithm for the identification of kernel parameters based on minimization by NM is presented in Algorithm 3.1.

In the solution of the inverse problem, the advantages of high-performance computing become more evident, since we solve a regression problem and simulate 10000 time-steps of a nonlocal diffusion equation at each iteration. Therefore, it is paramount that we exploit parallelism in the solution of Algorithm 3.1. We implement the learning algorithm in Python; we make use of the NumPy library for the Least Squares regression, and SciPy for the minimization, using the built-in Nelder-Mead method. We parallelize both Level 1 and 2 using the MPI4Py library. At Level 1, each processor computes a section of the RHS, as they are independent computations from the already available training dataset. The LHS is computed once at the beginning of the algorithm, as it is constant for all iterations. Then, in Level 2, at each time-step of Eq. (3.26), we parallelize the computation of $\tilde{p}_i^{k+1}$. In the end, the parallel implementation speeds up the costly computation of nonlocal operators, and allows the algorithm to converge in less than two hours for Case 1, with $m = 601$ grid points running 200 cores, and in slightly more than two hours for Cases 2 and 3, with $m = 1601$ grid points using 400 cores.

3.6 Results and Discussion

3.6.1 Method of Manufactured Solution

We assess the learning algorithm and nonlocal modeling proposed via the Method of Manufactured Solutions, where we produce training data from a known kernel and recover it through the ML algorithm as a necessary consistency check of the proposed ML framework. Starting from an initial condition, we solve the nonlocal diffusion equation, Eq. (3.17), with the kernel parameterized by known α_{true}, δ_{true}, and D_{true}, generating the snapshots of $p(x,t)$ to be provided to the ML algorithm.

We simulate the nonlocal diffusion problem, Eq. (3.22), in a domain $\mathcal{D} = [-1,1]$ with $L = 2$, and select $\alpha_{\text{true}} = 1.5$, $\delta_{\text{true}} = L/2$, and $D_{\text{true}} = 0.1$ as our parameters. For comparison, we simulate the nonlocal diffusion problem with spatial discretization using $m = 101$ points in space, solving the equation over $n = 200$ time-steps in size with $\Delta t = 0.01$. The initial condition for the nonlocal diffusion problem is a Dirac delta function at $x = 0$ with area equal to 1. Similarly to the DDD dataset, we let the system evolve and only use the last 200 time-steps of the simulation to collect the training and testing sets, using 80% of time-steps for training, and the rest for testing.

We compare the relative errors of α, δ, and D by the following expression

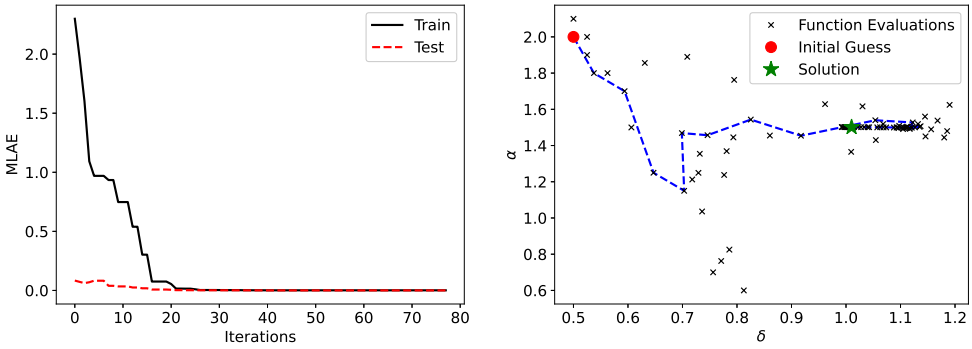
$$\varepsilon_i = \frac{|\xi_{i,\text{opt}} - \xi_{i,\text{true}}|}{|\xi_{i,\text{true}}|}, \tag{3.27}$$

where ξ_i represents the parameter in consideration, opt corresponds to the optimal value found by the algorithm, and true denotes the true parameter value.

We adopt a parametric space with bounds of $\alpha = [0,4]$, and $\delta = [\Delta x, L]$. Given the true values of $\alpha_{\text{true}} = 1.5$ and $\delta_{\text{true}} = L/2$, we take the initial guess to be $\alpha_0 = 2$, $\delta_0 = L/4$. We present the parameter, training and testing error results in Table 3.1. We verify that the algorithm successfully identifies the parameters within a maximum of 1% error for the horizon δ, while α and D are within 0.01% error. This example showcases that the decoupling of the learning in 2 levels leads to the correct kernel. Given the higher number of training points, and the time-dependent dynamics, it is expected to have a lower testing error, compared to training.

Table 3.1 Parameter and algorithm errors for the manufactured solution

α	δ	D	Training	Testing
9.81e-4	1.02e-2	6.63e-4	9.50e-4	1.06e-4



(a) Iteration errors.

(b) Solution path.

Figure 3.8 Iteration errors (training and testing), and the solution path of the ML algorithm when solving the inverse problem of a manufactured solution with a known kernel.

We plot the training and test errors over the number of iterations in Fig 3.8. We also illustrate the solution path from the initial guess to the final parameter estimates of α and δ, explicitly showing the function evaluations driving the iterations of Nelder-Mead optimization. We further explore the robustness of the algorithm with the DDD-based dataset.

3.6.2 DDD-Driven Results

We now employ the ML framework on the dataset generated by DDD simulations, represented by the PDFs of shifted dislocation positions obtained at the last 10000 time-steps, as highlighted in Fig. 3.5. We expand the robustness assessment of the framework, and we test other critical aspects such as sensitivity to the initial guess and train/test ratios.

The AKDE algorithm removes the noise from lack of data-points, especially at the tails, and produces a smooth curve throughout the domain. The intrinsic noise related to the variable number of particles is embedded in the PDF estimation, leading to smooth curves throughout the time-range of our data. For these reasons, we expect the algorithm to perform well with the DDD data.

We start by investigating the solution with different train-test splits. We compare the results of a 80/20 split as in the manufactured solution with a 60/40 split. We adopt an initial guess of $\alpha_0 = 2$, $\delta_0 = L/2$, which resides at the center of the same parametric range used before, $\alpha = [0, 4]$, and $\delta = [\Delta x, L]$. We run Algorithm 3.1 for Cases 1, 2, and 3, and collect the optimal values of α, δ, D, the computational cost in terms of Nelder-Mead iterations, the cost function evaluations, and the training and validation cost. We present those results in Table 3.2.

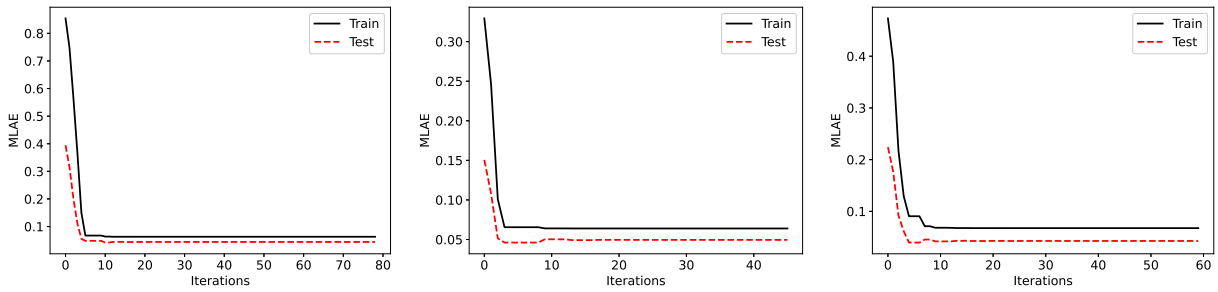
We highlight the results of α in Table 3.2 in comparison to the exponent of power-lay scaling in velocity distributions from Fig. (3.4). We note that the faster velocity decay in Case 1 with $\beta = 3$ is consistent with the kernel exponent $\alpha = 2.99$. Similarly, for Cases 2 and 3, the velocity distribution decay was found to be around $\beta = 2.4$, while the kernel exponent from the ML was found to be $\alpha = 2.40$ for Case 2 and $\alpha = 2.54$ for Case 3 under the 80/20 split. We will further comment this connection on the Discussion section.

The results obtained with the two train-test splits are equivalent in terms of the kernel parameters and the overall cost. Indeed, there is no evident difference in choosing one ratio over the other. The main difference comes in the overall training and testing cost. We observe that in all cases the training cost is larger than the test cost, similarly to the results obtained with the manufactured solution. This is due to the time-dependent dynamics of the PDF evolution and the higher availability of training points, as in the manufactured case. However, here we have another contributing factor. As discussed in Section 3.3, earlier data-points will be heavily influenced by the initial load application, while late points will be closer to a steady-state. The test set in the 60/40 split includes more earlier points, and therefore sees their influence reflected in higher training costs, besides allowing for less training points to make the model more general. For the remaining results, we choose the 80/20 split as our representative case.

We plot the evolution of training and test MLAE values over the Nelder-Mead iterations in Fig 3.9. We see that the algorithm quickly gets near the solution as the errors drop sharply near the initial iterations. Then, the errors remain nearly constant as the minimizer further explores the research space in the proximity of the minimum.

Table 3.2 Machine Learning results for two train-test split solutions.

	Case 1		Case 2		Case 3	
Train/Test Split	80/20	60/40	80/20	60/40	80/20	60/40
α	2.99	3.00	2.40	2.37	2.54	2.51
δ	20.62	18.79	33.7	33.07	34.03	33.72
D	3.64e-4	3.55e-4	7.40e-4	7.66e-4	1.78e-3	1.83e-3
# Iterations	79	75	46	79	60	85
# Evaluations	154	149	89	157	108	159
Training Cost	6.30e-02	5.73e-02	6.39e-02	5.55e-02	6.75e-02	5.57e-02
Testing Cost	4.42e-02	6.33e-02	4.95e-02	8.70e-02	4.28e-02	6.21e-02



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 3.9 Evolution of training and testing MLAE values computed over the NM iterations.

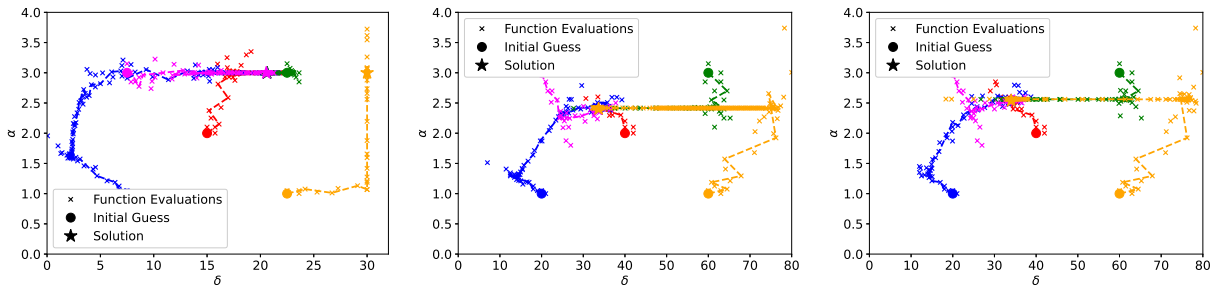
We further explore the capabilities of the proposed ML algorithm and test the performance of the parameter learning under different initial conditions beyond the central point, and choose four extra points near the corners of our parametric search space:

1. Guess 1 (original guess at the center): $\alpha_0 = 2$, $\delta_0 = L/2$.
2. Guess 2: $\alpha - 0 = 1$, $\delta_0 = L/4$.
3. Guess 3: $\alpha_0 = 3$, $\delta_0 = 3L/4$.
4. Guess 4: $\alpha_0 = 3$, $\delta_0 = L/4$.
5. Guess 5: $\alpha_0 = 1$, $\delta_0 = 3L/4$.

We present the final results in Table 3.3. In general, an initial guess close to the center of the parametric space leads to less iterations for Cases 2 and 3. The only different result we obtain is

Table 3.3 Machine Learning results for different initial guess combinations of α and δ.

Case		Guess 1	Guess 2	Guess 3	Guess 4	Guess 5
1	α	2.99	2.99	2.99	2.99	2.99
	δ	20.62	20.63	20.63	20.56	30.00
	D	3.63e-4	3.63e-4	3.63e-4	3.63e-4	3.63e-4
	# Iterations	79	122	57	76	30
	# Evaluations	154	219	114	152	59
2	α	2.40	2.40	2.40	2.40	2.40
	δ	33.70	33.66	33.66	33.68	33.68
	D	7.40e-4	7.40e-4	7.40e-4	7.40e-4	7.40e-4
	# Iterations	46	73	86	67	129
	# Evaluations	89	142	163	132	253
3	α	2.54	2.54	2.54	2.54	2.54
	δ	34.03	34.02	34.03	34.03	34.01
	D	1.78e-3	1.78e-3	1.78e-3	1.78e-3	1.78e-3
	# Iterations	60	87	120	68	211
	# Evaluations	108	169	233	126	408



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 3.10 Solution path from different combinations of initial guess.

for Case 1, Guess 5, where the horizon δ is computed as the upper-bound L, yet with α, D, and MLAE values sufficiently close to the results of other initial guess combinations. Based on this observation, the horizon δ seems to have a lower bound, above which the results are less sensitive to increasing horizon. We show the different paths the algorithm takes under the proposed initial guess combinations in Fig. 3.10, where we can see the function evaluations made by the algorithm and the final solutions, illustrating their proximity. We can also distinguish the upper-bound solution of Case 1, Guess 5 in the same figure.

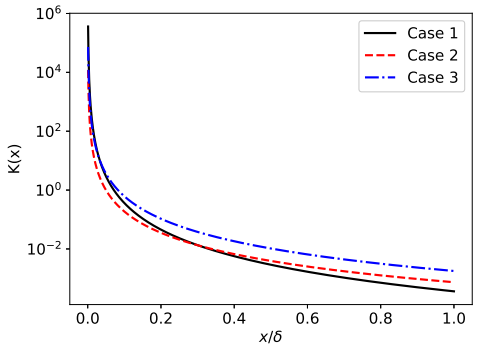


Figure 3.11 Final kernel shapes from optimized parameters obtained trough the ML algorithm, scaled by δ.

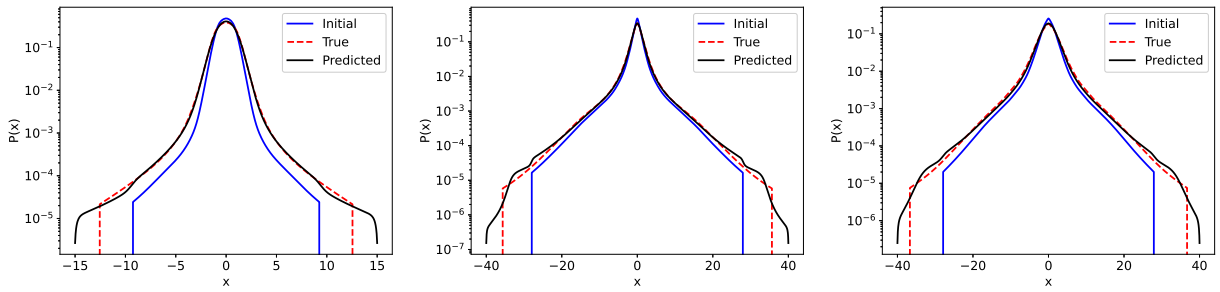
We choose the results from Guess 1, at the center of the parametric space, to be the representative parameters for kernel reconstruction and visualization. From the optimal values of the power-law decay exponent α, horizon δ, and coefficient D, we compute the nonlocal kernel following the definition from Eq. (3.20). We plot the kernel shapes for Case 1, 2, and 3 in Fig 3.11.

Finally, to illustrate the nonlocal model's potential to simulate the evolution of position PDFs from DDD simulations, we run the model using Eq. (3.26) with the optimal parameter values from Guess 1 combination, starting from the initial time-step of training, until the last time-step of testing, covering the whole range of available data. We measure the accuracy of the model using the l_2 relative error at the last time-step, defined as

$$\epsilon = \frac{\|\tilde{p} - p\|_2}{\|p\|_2}, \tag{3.28}$$

where $\tilde{p}$ represents the model solution at the specified time-step, and p is the true PDF obtained from DDD at the same time measure.

We compute the relative l_2 error and obtain ϵ_1 = 4.75e-2, ϵ_2 = 4.22e-2, and ϵ_3 = 6.30e-2 for Case 1, Case 2, and Case 3, respectively. Considering that this simulation takes over the 10000 time-steps of available data, the maximum relative error of 6.3% in the l_2 sense for Case 3 shows that the model can successfully reproduce the overall dynamics of the fluid-limit motion of dislocation particles in one dimension. We further illustrate the final shape of the PDF from the model, and compare it with the true shape at the final time-step in Fig 3.12.



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 3.12 Simulation of the nonlocal model for the whole time-interval of available data, highlighting the initial PDF, and the final distributions of the true data and the nonlocal prediction.

We verify that the ML algorithm successfully captured the parameters that best describe the evolution of dislocation position PDFs according to the proposed nonlocal diffusion model.

3.6.3 Discussion

Given the broad scope of this work, we divide the main discussion among three dominant facets. We discuss the overall capabilities of the proposed bi-level ML algorithm, followed by a discussion on the nonlocal model itself, and how the nonlocal kernel is connected to the particle dynamics observed in DDD.

We start by examining the ML aspects of the data-driven approach. It is evident that the proposed framework for solving the inverse problem of finding the parameters of a nonlocal Laplace operator is successful in the present scenario. Starting with the manufactured solution, we see the training error quickly approaching the plateau in Fig. 3.8. The decoupling of the nonlocal diffusion coefficient D from other kernel parameters α and δ (which is possible due to the linearity of the operator) facilitates the dimensionality reduction and the implementation of the bi-level algorithm. This shows one advantage of the proposed algorithm when compared to existing optimization-based kernel learning methods [51].

The robustness is guaranteed by the data-driven learning of a DDD-based kernel. The bi-dimensional optimization algorithm converged in as few as $O(10^2)$ function evaluations and $O(10)$ Nelder-Mead iterations in most cases. We also observed the convergence to the same parameters in

all but one different combinations of initial guess, which is an advantage over existing algorithms based on deep-neural networks [48]. Additionally, the proposed framework easily generalizes to more complex kernel shapes other than a pure power-law decaying shape, by simply adding the multiplicative factor P and learning its parameters. In fact, when defining the kernel as, e.g., a combination of basis functions, the algorithm seamlessly accommodates this approximation through inserting additional columns on the RHS of Eq. (3.23), and solving for more coefficients at Level 1. In other words, the minimization will always be two-dimensional, and the inclusion of polynomial basis functions would only affect the Least-Squares problem. In the end, this is more efficient than minimizing over all parameters, since the bottleneck of the learning problem is in the computation of the forward nonlocal solution during the optimization. The use of high-performance computing and parallelization further enhanced the performance of the learning framework, making it scalable and reducing the computational burden of the forward problem.

The efficiency of the algorithm is reflected in the consistency of results obtained in Cases 1, 2, and 3. It is clear that the nonlocal model is the appropriate choice for this particular problem, evidenced by the large value of horizon δ, to the order of 400-600 times the grid size. The long-range interactions from DDD are therefore represented as a nonlocal kernel with large horizon. Moreover, multiplication mechanisms correspond to larger values of δ, since the avalanches and the associated collective dynamics, represented by intermittency in velocity signals, lead to heavy-tail velocity distributions, which translates into heavy tails in the corresponding PDFs. This is in contrast with Case 1 without multiplication, where we have the PDFs closer to normal distributions than the ones from Cases 2 and 3. Therefore, the anomalous behavior in the discrete case leads to nonlocality also in the continuum case.

The other immediate observation related to the nonlocal model is the meaning of α, which clearly distinguishes the dynamics of dislocations with and without multiplication. For Case 1 without multiplication, we obtain a value of α closer of 2.99, while the multiplication mechanisms of Cases 2 and 3 are translated into $\alpha = 2.4$ and $\alpha = 2.54$, respectively. As anticipated in Section 3.4, the operator $\mathcal{L}$ obtained with the power-law nonlocal kernel with finite horizon is equivalent

to the truncated fractional Laplacian of fractional order s via the relationship $\alpha = 1 + 2s$. Under such view, Case 1 would correspond to a fractional Laplacian with $s = 0.99$, while Cases 2 and 3 would take $s = 0.7$ and $s = 0.77$, respectively. In this perspective, it is straightforward to see the evolution of dislocation PDFs as being super-diffusive, with Case 1 being the closest to a classical diffusion process, yet with a pronounced nonlocality due to rearrangements in the dislocation structure due to annihilations. However, the multiplication mechanism is the main factor that turns a rather diffusive process into super-diffusive. Moreover, the super-diffusion is intensified under a lower load, Case 2, where the external stress state allows faster relaxation and stronger subsequent avalanches compared to Case 3, where the overall higher stress state makes all dislocations move faster, yielding less relaxation time to a critical, metastable configuration.

The most striking observation related to the kernel discovery is the correspondence between the kernel fractional parameter α and the scalings of velocity distribution tail from Fig. 3.4. The empirical scaling observed here and among other works in the literature matches the values of α found by the data-driven kernel learning algorithm. This is not surprising, as we can take the velocity distribution to be jump size distributions that define a particular Lévy measure, essential when transforming a stochastic process into differential equations through the Semigroup theory [114]. The formal and complete definition of the stochastic process that governs the dislocation trajectories is out of the scope of this paper, yet we see jump size distributions following Fig. 3.4 lead to operators defined by power-law kernels taking the form of Eq. (3.20).

From a wider perspective, the procedure and reasoning developed in this paper do not need to be restricted to dislocation dynamics. The problem of learning kernels and dynamics from high-fidelity simulations and real-life data is a relevant research topic of increasing popularity. While the methodology presented in this work highlights the inference of a nonlocal operator from physical mechanisms, more specifically dislocation dynamics, the same procedure may be applied to tie the use of other non-standard nonlocal operators to the anomalous physical processes that they describe.

CHAPTER 4

AN INTEGRATED SENSITIVITY-UNCERTAINTY QUANTIFICATION FRAMEWORK FOR STOCHASTIC PHASE-FIELD MODELING OF MATERIAL DAMAGE

4.1 Introduction

Companies want to deliver safe products to clients with a solid knowledge of the component's life cycle within admissible ranges of operation. With that objective, industry spends millions of dollars every year in manufacturing, instrumenting and conducting validation tests. In order to reduce costs, manufacturers invest heavily in early stages of product design to reduce final project cost and development time. Decisions made in design phase have an impact of 70% of final cost. In that sense, numerical simulations have contributed to reduce the cost and time spent in product development through CAE applications in solid and fluid mechanics, and failure analysis play an important role in saving costs and ensuring safety.

The importance of reliable failure analysis motivates the development of trustworthy mathematical models and robust numerical methods. Several researchers have developed models of damage initiation, propagation and fatigue life of materials, using continuum damage models and fracture mechanics [153–155]. The *ad hoc* characteristics of those classical models prevent them to apply to a wider range of problems, affecting their predictability. Some of the limitations appear when dealing with crack initiation or branching, for example.

Recently, phase-fields have become a solid alternative to treat those difficulties in damage and fatigue modeling. Phase-field models were first developed to solve fluid separation problems [54]. Its capability of modeling sharp interfaces through a smooth continuous field extended its application to different multiphase problems with moving boundaries, including solidification [55], tumor growth [56], two-phase complex fluid flow [57] and fluid-structure interaction [58]. Over the last decade, phase-field models were used in simulations of brittle [31, 156, 157] and ductile fracture [32, 158]. Phase-field damage models are able to capture many effects such as

crack initiation, propagation, branching and coalescence. Crack branching is typically observed in dynamic fracture [33, 159]. Fatigue effects were modeled using thermodynamically consistent approaches [160] and fractional derivatives [161]. Boldrini et al (2016) [34] developed a non-isothermal and thermodynamically consistent framework for damage and fatigue using phase-fields. Spatial convergence and 2D results were presented by Chiarelli et al (2017) [162]. A comparison between semi and fully implicit time integration schemes was analyzed by Haveroth et al (2018) [163].

Despite the ability to describe crack geometry and incorporate naturally fatiguing mechanisms and different constitutive laws, most examples of phase-field solutions studied so far include geometric characteristics that drive the crack path to determined places. The presence of notches, indentations, regions of stress concentration, is recurrent and leads to controlled experiments with a predictable crack path (always assuming a perfect material). In those models, there is no consideration of stochastic effects to account for material and manufacturing imperfection, surface roughness, or even misalignment of loading conditions that, in practice can drive failure to other locations.

The development of solution methods for stochastic differential equations have been under great focus over the last decades. Traditional sampling methods such as the Monte Carlo (MC) method [59, 164] are widely used to compute solution statistics. However, MC has slow convergence rate, to the order of $N^{-1/2}$ (N is the number of realizations), requiring an abundant number of simulations, which can be prohibitive for a computationally expensive forward solver. A remedy is to build surrogate models through established methods such as Polynomial Chaos [63, 165, 166], where the stochastic model is approximated by a set of polynomial basis, or its generalization via Galerkin projections [64, 65, 167].

Such surrogate models have the disadvantage of being intrusive, in which they require modifications on the governing equations and become impractical in complex systems. Conversely, non-intrusive methods use the forward solver as a black-box, with general application. Probabilistic Collocation Methods (PCM), or Stochastic Collocation[66, 67], sample the random space

efficiently, leading to better convergence than MC, while still preserving a simple solution structure, where realizations can be independently sampled, with easy parallelization. The curse of dimensionality carried by the tensorial products involved in the computation of solution statistics can be overcome with the use of Sparse Grids [68], or dimensionality reduction techniques, such as active subspace methods [168–170].

Yet, only a few works focused on studying the stochastic nature of phase-field models. Uncertainty quantification and sensitivity analysis in the context of phase-field models for polymeric composites was addressed by Hamdia et al (2015) [171]. Authors used different methods to address the parametric sensitivity from a toughness test geometry with two notches. In that scenario, matrix elasticity modulus, volume fraction of clay platelets and the fracture energy of the matrix were the most influential parameters. In another application, sensitivity analysis was performed in a tumor growth phase-field model [172]. Using stochastic collocation, authors identified two most sensitive parameters: the rate of cellular mitosis and nutrient mobility. In that case, nutrient transport was governed by a traditional diffusion equation.

Besides the deterministic treatment of failure process, historically, phase-field models neglected nonlocal terms in the free energy potentials. The question of how those assumptions affect the model predictability of failure motivates a systematic way to evaluate model form uncertainty, which has become an active research topic recently [173, 174].Through determining the salient sources of uncertainty, educated modifications in the modeling process can be proposed, informed by the model itself. This self-assessment procedure could be further extended to incorporate stochasticity in space, and stochastic processes.

In this chapter we develop a framework to assess model form uncertainty of a damage and fatigue phase-field model through uncertainty propagation and sensitivity. We define parameters related to damage and fatigue, as well as viscous damping effects, as random variables and solve a stochastic system of equations that model material damage using phase-fields. We compute the expected solution and standard deviation fields in univariate and multivariate setups. We define local sensitivity expectation, where we use complex-step differentiation [175] to compute local

sensitivity at each collocation point in the random space. A variance-based method [176] is used to compute each parameter's Sobol indices in the global sensitivity analysis. These methods establish a framework to systematically investigate model form uncertainty. An incorrect operator can be detected by looking at the parameters that multiply them, which will be more sensitive and influential in the total output uncertainty. Assuming the model form is correct transfers the salient uncertainty associated with the operator to its parameters. We identify the parameters that most contribute to total output variance so that new models can be derived, mitigating model form uncertainty.

The chapter is organized as follows. In Section 4.2 we present a stochastic version of the damage and fatigue phase-field model derived in Boldrini et al (2016) [34]. We discuss the system of PDEs, the finite element spatial discretization, the semi-implicit scheme for time integration and the stochastic discretization methods, namely Monte Carlo sampling and Probabilistic Collocation. Then, we present methods used for local and global sensitivity analysis in Section 4.3. In Section 4.4 we show the results of uncertainty and sensitivity for two representative numerical examples.

4.2 A Stochastic Damage and Fatigue Phase-Field Framework

We show a stochastic version of the phase-field framework for structural damage and fatigue presented in Boldrini et al (2016) [34], which consists in a deterministic system of coupled differential equations for the evolution of displacement $\mathbf{u}$, velocity $\mathbf{v} = \dot{\mathbf{u}}$, damage φ and fatigue $\mathcal{F}$. Damage φ is a phase-field variable describing the volumetric fraction of degraded material, taking values $\varphi = 0$ for virgin material, $\varphi = 1$ for fractured material, and can change between those states, $0 \leq \varphi \leq 1$, as a damaged material. The evolution equation for the damage field is an Allen-Cahn type, and is obtained along with the equations of motion for $\mathbf{u}$ and $\mathbf{v}$ through the principle of virtual power and entropy inequalities. The fatigue field $\mathcal{F}$ is treated as an internal variable, whose evolution equation is obtained through constitutive relations that must satisfy the entropy inequality for all admissible processes. The geometry is defined over a spatial domain $\Omega_d \subset \mathbb{R}^d$, $d = 1, 2, 3$ at time $t \in (0, T]$.

The overall construction of the model gives origin to a general set of equations that can take different forms based on suitable choices of free-energy potentials, boundary and initial conditions. The choice of free-energy potentials affects directly the final model form, the specific coupling between the fields of interest and material behavior. The free-energy potential $\mathcal{J}(\varphi, \mathcal{F})$, related to damage φ and fatigue $\mathcal{F}$ has the traditional form [34]

$$\mathcal{J}(\varphi, \mathcal{F}) = g_c \frac{\gamma}{2} |\nabla \varphi|^2 + \mathcal{K}(\varphi, \mathcal{F}), \tag{4.1}$$

where g_c is the Griffith energy, $\gamma > 0$ is the phase-field layer width parameter and $\mathcal{K}(\varphi, \mathcal{F})$ is a function that relates the change of damage to material fatigue. This general form for free-energy has been used since the first phase-field models, where the first term corresponds to *interfacial energy* and originates the Laplacian operator in Allen-Cahn type equations. The second term is called *mixing energy* in other disciplines, and may take different forms based on the application, even within damage models.

From the governing equations of the deterministic phase-field model we can identify many material parameters that are easily obtained from experimental procedures, namely elasticity constants (E and ν), and density ρ_0. For that reason, and to reduce the complexity of the analysis, we consider them to be deterministic. The remaining parameters, on the other hand, are either not physically measurable, or their value is uncertain, and are considered to be stochastic. First, we have those parameters which are proportionality constants due to mathematical modeling, such as rate of change of damage and fatigue, c and a, respectively. Similarly, phase-field layer width γ is a parameter that controls the diffusivity of damage field, therefore is a mathematical artifact that should be as close to zero as possible to recover the sharp interface. Furthermore, the viscous damping coefficient b is not promptly identifiable, and it would require a correlation with damping ratio to be obtained experimentally through modal testing. Finally, the Griffith energy g_c, although related to stress intensity factor, would require further experiments for each specific material, since its range varies broadly for materials with the same elastic properties.

We define the parameters mentioned as random variables and derive the stochastic version of the damage and fagitue phase-field model. Let $(\Omega_s, \mathcal{G}, \mathbb{P})$ be a complete probability space, where Ω_s is

the space of outcomes ω, $\mathcal{G}$ is the $\sigma-$algebra and $\mathbb{P}$ is a probability measure, $\mathbb{P} : \mathcal{G} \rightarrow [0, 1]$. We define the five-dimensional set of random parameters $\xi(\omega) = \{\gamma(\omega), g_c(\omega), a(\omega), b(\omega), c(\omega)\}$. The random nature of ξ makes the operators and output fields to also be random. However, we simplify the notation and only explicitly represent the random parameters as $\xi = \xi(\omega)$, supressing the random variable indication elsewhere. Choosing appropriate free-energy potentials, we obtain the stochastic equations for a linear elastic and isotropic material without temperature evolution defined over $\Omega_d \times (0, T] \times \Omega_s$:

$$\begin{cases} \dot{\mathbf{u}} = \mathbf{v}, \\ \dot{\mathbf{v}} = \text{ div } \left((1-\varphi)^2 \dfrac{C}{\rho_0} \mathbf{E} \right) + \dfrac{b(\omega)}{\rho_0} \text{ div } (\mathbf{D}) - \dfrac{\gamma(\omega) g_c(\omega)}{\rho_0} \text{ div } (\nabla\varphi \otimes \nabla\varphi) + \mathbf{f}, \\ \dot{\varphi} = \dfrac{\gamma(\omega) g_c(\omega)}{\lambda} \Delta\varphi + \dfrac{1}{\lambda}(1-\varphi)\mathbf{E}^T C \mathbf{E} - \dfrac{1}{\lambda\gamma(\omega)} [g_c(\omega)\mathcal{H}'(\varphi) + \mathcal{F}\mathcal{H}'_f(\varphi)], \\ \dot{\mathcal{F}} = -\dfrac{\hat{F}}{\gamma(\omega)} \mathcal{H}_f(\varphi), \end{cases} \tag{4.2}$$

subjected to appropriate initial and boundary contitions, which depend on the physical problem. For the equation of motion, usually displacement or stress are known at the boundaries. For damage evolution, $\nabla\varphi \cdot \mathbf{n} = 0$ at the boundary, $\partial\Omega_d$.

The infinitesimal strain and the rate of strain tensors are represented by $\mathbf{E} = \nabla^S \mathbf{u}$ and $\mathbf{D} = \nabla^S \mathbf{v}$, respectively. C represents the elasticity tensor, as a function of Young's modulus E and Poisson coefficient ν. Parameter b is the viscous damping of the material, ρ_0 corresponds to the material's density. Following an argument by Lemaitre and Desmorat (2005) [153], λ is constructed such that the rate of change of damage should increase as damage increases, following the relation:

$$\frac{1}{\lambda} = \frac{c(\omega)}{(1 + \delta - \varphi)^{\varsigma}}, \tag{4.3}$$

where c and ς are positive parameters that should depend on the material. The relation includes δ, a small positive constant, in order to avoid numerical singularity.

The terms $\mathcal{H}'(\varphi)$ and $\mathcal{H}'_f(\varphi)$ are the derivatives of $\mathcal{H}(\varphi)$ and $\mathcal{H}_f(\varphi)$ with respect to φ and play an important role in the evolution of damage. Different choices for those potentials change the form of the transition of damage phase-field as fatigue changes from zero to g_c. Further details

about the behavior of fatigue potentials can be found in Boldrini et al (2016) [34]. If we choose the transition to be continuous and monotonically increasing, suitable choices for the potentials are:

$$\mathcal{H}(\varphi) = \begin{cases} 0.5\varphi^2 & \text{for } 0 \leq \varphi \leq 1, \\ 0.5 + \delta(\varphi - 1) & \text{for } \varphi > 1, \\ -\delta\varphi & \text{for } \varphi < 0. \end{cases} \tag{4.4}$$

$$\mathcal{H}_f(\varphi) = \begin{cases} -\varphi & \text{for } 0 \leq \varphi \leq 1, \\ -1 & \text{for } \varphi > 1, \\ 0 & \text{for } \varphi < 0. \end{cases} \tag{4.5}$$

The growth of the fatigue field $\mathcal{F}$ is controlled by the $\hat{F}$ parameter, which is related to the formation of micro-cracks that occur in cyclic loadings. Even in monotonic loadings there is growth of the fatigue variable, because a monotonic loading can be considered as one portion of a complete cyclic load. The form of $\hat{F}$ depends on the absolute value of the power related to stress in the virgin material:

$$\hat{F} = a(\omega)(1 - \varphi) \left| (C\mathbf{E} + b\mathbf{D}) : \mathbf{D} \right|, \tag{4.6}$$

where the parameter a in this case is chosen to give a linear dependence of the power of stress.

We construct a general framework to compute the stochastic solutions using three levels of discretization: in space, using finite element method; in time, using a semi-implicit integration scheme; and in the random space, where we choose our realizations randomly through Monte Carlo sampling, or by Probabilistic Collocation. Fundamentally, the finite element solver acts as a black box for any nonintrusive stochastic method. Details on the spatio-temporal discretization of the phase-field model can be found in the Appendix A.

4.2.1 Stochastic Discretization

Displacement, velocity, damage and fatigue from the phase-field framework solution are functions of random parameters; therefore, they are random fields. In order to obtain the statistical moments

of those outputs we will solve the deterministic system of equations over an ensemble of different realizations, each of them with distinct parameter values. The inputs for each realization depends on the sampling method. In this work we employ Monte Carlo (MC) and Probabilistic Collocation Method (PCM). Since those methods are non-intrusive, the finite element solver acts as a black box, where the choice of parameter values and computation of statistical quantities are simply pre- and post-processing tasks, respectively.

4.2.1.1 Probabilistic collocation method

The Probabilistic Collocation Method (PCM) poses a great advantage over the MC method, since PCM uses polynomial interpolation to approximate the solution in the random space. The mapping between the random and physical space is made through the probability density function of the uncertain parameters. Using orthogonal polynomials, such as Lagrange, the computation of expectation and variance reduces to running the simulation at the collocation points, reducing computational cost significantly, while improving convergence rates.

The mathematical expectation of the solution, $\mathbb{E}\left[U(x, y, t; \xi)\right]$, in a one-dimensional random space can be written as

$$\mathbb{E}\left[U(x, y, t; \xi)\right] = \int_a^b U(x, y, t; \xi)\rho(\xi)d\xi, \tag{4.7}$$

where $\rho(\xi)$ is the probability distribution function of ξ. In order to use Gauss quadrature, we must map the domains of the distribution to the interval $[-1, 1]$ in the standard domain of a variable η. The integral should then be written as

$$\mathbb{E}\left[U(x, y, t; \xi)\right] = \int_{-1}^1 U(x, y, t; \xi(\eta))\rho(\xi(\eta))Jd\xi(\eta), \tag{4.8}$$

where $J = d\xi/d\eta$ represents the Jacobian of the transformation. We approximate the expectation by introducing a polynomial interpolation of the exact solution in the random space, $\hat{U}(x, y, t; \xi)$:

$$\mathbb{E}\left[U(x, y, t; \xi)\right] \approx \int_{-1}^1 \hat{U}(x, y, t; \xi(\eta))\rho(\xi(\eta))Jd\xi(\eta). \tag{4.9}$$

We interpolate the solution in the random space using Lagrange polynomials $L_i(\xi)$:

$$\hat{U}(x, y, t; \xi) = \sum_{i=1}^{I} U(x, y, t; \xi_i) L_i(\xi), \tag{4.10}$$

which satisfy the Kronecker delta property at the interpolation points:

$$L_i(\xi_j) = \delta_{ij}. \tag{4.11}$$

We substitute Equation (4.10) into (4.9), and approximate the integral using a quadrature rule. Then, the expectation is now written as

$$\mathbb{E}\left[U(x, y, t; \xi)\right] \approx \sum_{q=1}^{Q} w_q \rho(\xi(\eta)) J \sum_{i=1}^{I} U(x, y, t; \xi(\eta)) L_i(\xi(\eta)), \tag{4.12}$$

where we compute the coordinates η_q and weights w_q for each integration point $q = 1, 2, \ldots, Q$. This computation can be evaluated efficiently by choosing the collocation points to be the same as the integration points q on the parametric space. Through the Kronecker property of the Lagrange polynomials (4.11), the approximation (4.12) is simplified as a single summation:

$$\mathbb{E}\left[U(x, y, t; \xi)\right] = \sum_{q=1}^{Q} w_q \rho(\xi_q(\eta_q)) J U(x, y, t; \xi_q(\eta_q)). \tag{4.13}$$

We use a linear affine mapping from the standard to the real domain: $\xi_q(\eta_q) = a + \frac{(b-a)}{2}(\eta_q + 1)$. This mapping gives us the Jacobian (for a one-dimensional integration) simply as $J = (b - a)/2$. In practice, after we find the quadrature points in the standard domain, we use the mapping to find the respective values of the random variable in our interval.

We can now approximate the integral and rewrite it as a summation over the collocation points, again assuming a *uniform distribution* for the parameters over the interval $[a, b]$, which gives us a constant value of $\rho(\xi) = 1/(b - a)$. The expectation becomes $\mathbb{E}\left[U(x, y, t; \xi)\right] = \frac{1}{2}\sum_{q=1}^{Q} w_q U(x, y, t; \xi_q)$. Similarly to the Monte Carlo method, the standard deviation is $\sigma\left[U(x, y, t; \xi)\right] = \sqrt{\frac{1}{2}\sum_{q=1}^{Q} w_q \left(U(x, y, t; \xi_q) - \mathbb{E}\left[U(x, y, t; \xi)\right]\right)^2}$.

If we want to generalize PCM for higher dimensions, it is just a matter of having additional integrals to Eq. (4.7). In discrete form, this reduces to

$$\mathbb{E}\left[U(x,y,t;\xi^1,\ldots,\xi^k)\right] = \mathbb{E}_{PCM}\left[U(x,y,t;\xi^1,\ldots,\xi^k)\right] \approx \sum_{q=1}^{Q}\cdots\sum_{l=1}^{L} w_q\ldots w_l\,\rho(\xi_q)\ldots\rho(\xi_l)\,J_q\ldots J_l\,U(x,y,t;\xi_q^1,\ldots,\xi_l^k) \tag{4.14}$$

where we have k summations, one for each dimension in the random space. In ξ_l^k the superscript indicates the dimension in the random space, and the subscript specifies the collocation point in that dimension. Simplifying the notation using $\mathbb{E}\left[U(x,y,t;\xi^1,\ldots,\xi^k)\right] = \mathbb{E}\left[U\right]$, the standard deviation becomes

$$\sigma\left[U(x,y,t;\xi^1,\ldots,\xi^k)\right] = \sigma_{PCM}\left[U(x,y,t;\xi^1,\ldots,\xi^k)\right] \approx \sqrt{\sum_{q=1}^{Q}\cdots\sum_{l=1}^{L} w_q\ldots w_l\,\rho(\xi_q)\ldots\rho(\xi_l)\,J_q\ldots J_l\,\left(U(x,y,t;\xi_q^1,\ldots,\xi_l^k)-\mathbb{E}\left[U\right]\right)^2}. \tag{4.15}$$

Here, we are required to perform a tensor product to obtain the set of parameters for each realization. We have the flexibility to introduce anisotropy on the sample space by using different number of integration points in each dimension, depending on the parameter. Moreover, once we have simulated all multidimensional tensor product simulations, obtaining first and total-order sensitivity indices is just a matter of post-processing. In all examples showed in this work we assume the random variables to be mutually independent. That assumption may not always hold true for a specific scenario, in which case a proper treatment should be given due to correlation between parameters.

Remark 4.2.1. *For the model studied in this work, a fully tensorial product as performed in PCM is sufficient. However, in more general systems with higher parametric dimensions (more than the 5 dimensions considered here), such approach is not efficient, since the tensor product increases the number of simulations exponentially. In medium-to-high dimensions, one alternative is to use Smolyak sparse grids[68] to reduce the number of realizations, while still achieving*

comparable accuracy. However, even such methods become expensive in high dimensions. Then, UQ is usually combined with dimensionality reduction techniques. Traditional methods include Principal Component Analysis (PCA) [177, 178] and low-rank approximations [179]. Recently, active subpace methods have been developed to find and exploit low-dimensional structures in the random space to obtain the solution statistics, and perform sensitivity analysis [168–170].

4.3 Sensitivity Analysis

The stochastic methods presented give us the ability to propagate parametric uncertainty through the model and obtain statistical information from the solution. We can take a step further and use the stochastic information to systematically study the importance of each parameter to solution uncertainty. Using a univariate PCM framework we can compute local sensitivity in a probabilistic way. Through multivariate PCM uncertainty propagation we can analyze the influence of each parameter in the total variance, by calculating sensitivity indices. In this Section, we derive the local and global sensitivity frameworks from the phase-field model stochastic discretization.

4.3.1 Stochastic Sensitivity Analysis

In local sensitivity analysis, we study the effect of only one parameter while keeping all others at their expected values. The sensitivity of the output $U(\xi^j)$ with respect to input parameter ξ^j, $j = 1, 2, 3, 4, 5$ being one of the parameters in $\xi(\omega) = \{\gamma(\omega), g_c(\omega), a(\omega), b(\omega), c(\omega)\}$, is also a random variable, so we compute its expected value as

$$\overline{S}_{U,\xi^j} = N_f \mathbb{E}(S_{U,\xi^j}) \tag{4.16}$$

where N_f is a normalization factor defined as

$$N_f = \frac{\bar{\xi}^j}{\bar{U}} \tag{4.17}$$

In Eq. (4.17), $\bar{\xi}^j$ is the expected value of the input parameter under consideration and $\bar{U}$ is the volume average of the expected solution U. In Eq. (4.16) we consider the expectation of sensitivities

over all the parameter intervals, computed using PCM as

$$\mathbb{E}\left[(S_{U,\xi^j})\right] \approx \frac{1}{2}\sum_{q=1}^{Q} w_q S_{U_q,\xi_q^j} \tag{4.18}$$

with S_{U_q,ξ_q^j} being the sensitivity at collocation point q for parameter j. The sensitivity around each integration point is computed using complex-step differentiation [175]. Unlike traditional finite-difference approximations, in complex-step differentiation we perturb the imaginary part of the parameter. Let $\xi^j = p + ih$ and $h \in \mathbb{R}$. Expanding $U(\xi)$ in Taylor series about the real part p we obtain

$$U(\xi^j) = U(p + ih) = U(p) + ihU'(p) - h^2U''(p) + O(h^2). \tag{4.19}$$

We then take the imaginary part on both sides to obtain the derivative of U around the real part of ξ:

$$U'(p) = S_{U,\xi^j} \approx \frac{Im(U(p + ih))}{h} \tag{4.20}$$

We can also take the real part of $U(p + ih)$ to recover the unperturbed solution and compute the expectation used in the normalization factor, Eq. (4.17).

The complex-step differentiation is second-order accurate and allows for small perturbations without incurring round-off and cancellation errors. Using complex-step differentiation has another enormous advantage over finite-difference schemes. We only need to evaluate one solution at each point, instead of two. We take the imaginary part of the solution to compute the derivative, and use the real part to calculate the volume average of the expected solution in Eq. (4.17).

Remark 4.3.1. *Using MATLAB, this method can be applied almost immediatly by running the simulation with an imaginary perturbation. However, in the derivation of complex-step differentiation, we have to assume that $U(\xi)$ is analytic, implying the Cauchy-Riemman equations must hold. We must also assure all operators in the implementations to be analytic [175] . Most of operators in MATLAB are compatible with complex numbers. Operations like maximum, minimum, greater or less than, should always compare only the real part. Moreover, when transposing vectors and matrices, we should use the dot-apostrophe transpose, that do not take conjugate of the elements.*

The only operation that we must redefine to be analytic is the absolute value, implemented as

$$abs(x+iy) = \begin{cases} -x - iy, \; if \, x < 0 \\ +x + iy, \; if \, x \geq 0 \end{cases} \tag{4.21}$$

4.3.2 Global Sensitivity Analysis

In a global analysis we are interested in how sensitive is the output when we perturb all input parameters. We use a variance-based method based on Sobol indices [180], where we compute the relative importance of input parameters to total output variance. To derive the algorithms for variance-based sensitivity analysis we refer to Saltelli et al (2010) [176]. For simplicity, we now denote our solution vector only as $U(\xi) = f(\xi^1, \xi^2, \ldots, \xi^k)$, a random variable function of random parameters ξ^j, $j = 1, 2, \ldots, k$, k being the total dimension of the random space (the number of random parameters). The effect of parameter ξ^j on variance V is

$$V_{\xi^j}\left(\mathbb{E}_{\xi^{\sim j}}(U|\xi^j)\right) \tag{4.22}$$

where $\xi^{\sim j}$ denotes the combination of all possible values for random parameters with the exception of ξ^j, which is fixed at some value. We interpret Eq. (4.22) as taking the expected value of U having fixed a value for ξ^j, and then taking the variance over all possible values of ξ^j. From the Law of Total Variance, we have

$$V_{\xi^j}\left(\mathbb{E}_{\xi^{\sim j}}(U|\xi^j)\right) + \mathbb{E}_{\xi^j}\left(V_{\xi^{\sim j}}(U|\xi^j)\right) = V(U) \tag{4.23}$$

The second term on the left-hand side is called the residual and $V(U)$ is the total variance. Normalizing Eq. (4.23) we obtain the sensitivity index that measures the effect on total variance by random variable ξ^j as:

$$S^j = \frac{V_{\xi^j}\left(\mathbb{E}_{\xi^{\sim j}}(U|\xi^j)\right)}{V(U)} \tag{4.24}$$

The sensitivity indices S^j only measure the first-order effect on variance from ξ^j, and disregards the interactions between ξ^j and other parameters. The total sum of S^j should be less than 1, the remainder being the high-order interactions between the parameters.

If we would like to compute any second, third, or higher-order indices, we would have to repeat this procedure for all combinations of interactions. An alternative is to compute the sensitivity index related to a given parameter and all its possible interactions with all parameters. In order to obtain the total effect index S_T^j we start from the total normalized variance and subtract the contribution of all first and high-order effects that do not include ξ^j. We write

$$S_T^j = 1 - \frac{V_{\xi_{\sim j}}\left(\mathbb{E}_{\xi^j}(U|\xi^{\sim j})\right)}{V(U)} \tag{4.25}$$

In practical terms, for every combination of parameters that do not include ξ^j we compute the expectation with respect to ξ^j. That means performing different combinations of univariate expected solutions with respect to ξ^j. Then, all these different expectations compose a $k-1$ dimensional random space, from which we compute the variance. From 5-D PCM full tensor product we already possess all possible combinations between the parameters, so the computation of S^j and S_T^j are just a matter of postprocessing the realizations.

4.3.3 Integrated Sensitivity-Uncertainty Quantification Framework

Given the stochastic discretization methods from Section 4.2.1 as integrating building blocks, we formulate a framework to quantify model-form uncertainty through a parametric uncertainty/sensitivity propagation. The stochastic and global sensitivity analyses are naturally integrated to the framework using PCM. The stochastic sensitivity procedure is summarized in Algorithm 4.1, where the solution sensitivity with respect to a given parameter is obtained through the complex-step differentiation. The participation of each parameter in total uncertainty given by sensitivity indices is outlined in Algorithm 4.2. We further elaborate on the integrated framework's development and capabilities via two canonical examples in the next Section.

4.4 Numerical Results

We now present two representative numerical examples to show the capabilities of the proposed methodology to assess uncertainty and sensitivity of damage phase-field models. The first example

Algorithm 4.1 Stochastic Sensitivity Analysis

for Random parameters $\xi^j = 1 \rightarrow k$ **do**
    Choose a perturbation h, and define the random parameter as $U(\xi^j) = U(p + ih)$.
    Solve the stochastic problem in one dimension with perturbed inputs $\xi^j = p + ih$.
    Compute the sensitivity of the solution at every collocation point following Eq. (4.20).
    Compute the expectation of sensitivities $\mathbb{E}\left[(S_{U,\xi^j})\right]$, following Eq. (4.18).
    Compute the normalization factor N_f using the volume average of expected solution, Eq. (4.17).
    Given $\mathbb{E}\left[(S_{U,\xi^j})\right]$ and N_f compute the local sensitivity $\overline{S}_{U,\xi^j}$ using Eq. (4.16).
end for

Algorithm 4.2 Global Sensitivity Analysis

For all k random parameters, solve the $k-$dimensional stochastic problem employing PCM.
for Random parameters $\xi^j = 1 \rightarrow k$ **do**
    Given the tensor product results in all collocation points, compute first-order sensitivity index S^j using Eq. (4.24).
    Given the tensor product results in all collocation points, compute total-order sensitivity index S_T^j using Eq. (4.25).
end for

is the single-edge notched tensile test case, a traditional benchmark test with mode I crack propagation. With the notched geometry, we first investigate the convergence of MC and PCM methods in the univariate and multivariate uncertainty propagation. Then, we show the expectation and standard deviation of damage evolution for each parameter and compare that with 5-D parametric uncertainty propagation. Next, we show local sensitivity expectation results for each parameter in the univariate framework. Last, we compute Sobol indices in the global sensitivity analysis and comment on the different influence of each parameter.

The second example is a standard tensile test specimen, symmetric and with no notches or existing cracks. We run the stochastic framework and show the expected crack path and its uncertainty. We also run local and global sensitivity analyzes to understand how the lack of pre-existing crack or notch affects the output uncertainty.

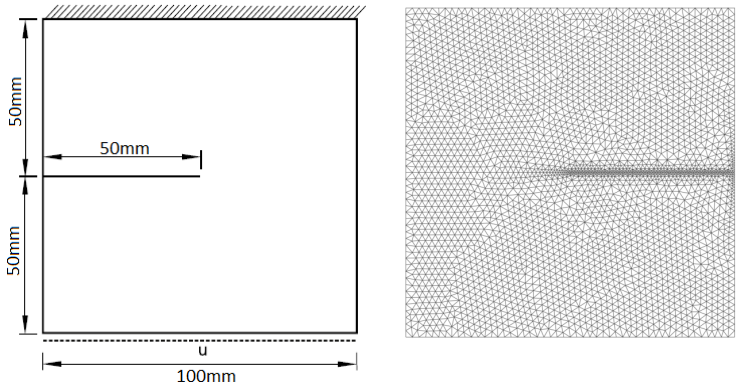


Figure 4.1 Left: Geometry and boundary conditions for single-edge notched tensile test. Right: Finite element mesh.

Table 4.1 Expected value of stochastic parameters for single-edge notched tensile test.

Parameter	Value
a (rate of change of fatigue)	$5 \times 10^{-7}\ m^2$
b (viscous damping)	$1 \times 10^{8}\ Ns/m^2$
c (rate of change of damage)	$1 \times 10^{-5}\ m/Ns$
g_c (Griffith energy)	$2700\ N/m$
γ (phase-field layer width)	$1 \times 10^{-3}\ m$

4.4.1 Single-Edge Notched Tensile Test

The notched geometry is a benchmark test, consisting in a square of material with a pre-existing crack in the middle of the specimen, see Fig. 4.1. We constrain the body at the top and apply a prescribed displacement at the bottom, at a rate of $3 \times 10^{-4}\ m/s$. The finite element mesh has 3395 nodes, composing 6498 linear triangle elements with smallest element size being 0.404 mm. The final time is $T = 0.5\ s$ and we integrate the solution over time-steps of $\Delta t = 1 \times 10^{-3}\ s$. We consider a material with $E = 160\ GPa$, $\nu = 0.3$ and $\rho_0 = 7800\ kg/m^3$ under plane stress conditions with thickness of $h = 5\ mm$. As stated in Section 4.2.1, we assume a uniform distribution for the random variables, with a range of ±10% from their expected values, which are given in Tables 4.1.

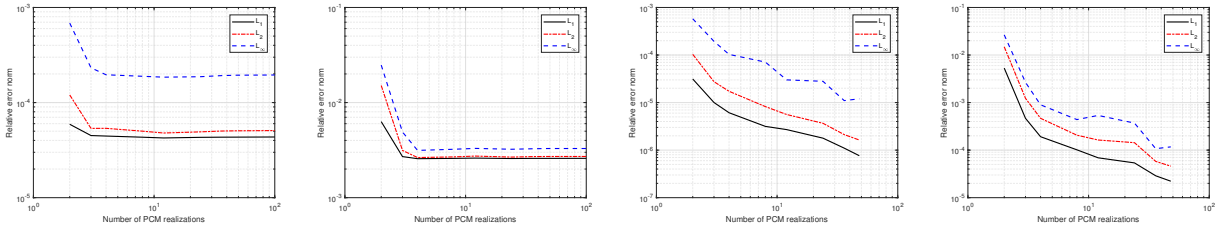


Figure 4.2 Univariate simulations: expectation (a) and standard deviation (b) error between MC results with 10^4 samples and solutions from PCM with different number of collocation points. We observe that with 4 points we obtain comparable accuracy in PCM. Expectation (c) and standard deviation (d) PCM convergence, where the reference for error computation is a solution with 100 collocation points in the univariate case. The convergence rate is close to linear.

4.4.1.1 Convergence

We study the convergence of PCM estimates (expectation and standard deviation) of the damage field $\hat{\varphi}(x, y, T)$, at final simulation time $T = 0.5\ s$. The relative error, e_{PCM}, between the PCM estimate $\hat{\varphi}(x, y, T)$ and a reference solution $\varphi_{ref}(x, y, T)$, over the spatial domain at time T, is defined as

$$e_{PCM} = \frac{\|\hat{\varphi} - \varphi_{ref}\|_p}{\|\varphi_{ref}\|_p}, \tag{4.26}$$

where the $p-$norm for a vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ is $\|\mathbf{w}\|_p := \left(\sum_{i=1}^{n} |w_i|^p\right)^{1/p}$, for $p = 1, 2$, and $\|\mathbf{w}\|_\infty := \max_i |w_i|$.

In Fig. 4.2 we plot the univariate convergence analysis for random parameter γ. We first choose the reference solution to be a MC estimate with 10^4 realizations ((a) and (b)), showing that, with only a few realizations, PCM has equivalent accuracy as MC. Since our method is shown to be convergent, although biased, we establish a PCM benchmark solution with 100 collocation points. The goal is to estimate the practical convergence rate of PCM, which sheds light on the smoothness of the solution in the parametric space. We see that the convergence rate of PCM with respect to the PCM benchmark solution is close to linear ((c) and (d)), instead of e.g. exponential, revealing that the true solution to our problem is not smooth, and it does not belong to higher-class Sobolev spaces in the parametric space.

Fig. 4.3 (a) and (b) show the relative error of PCM, taking a 10^4 sample MC estimate as

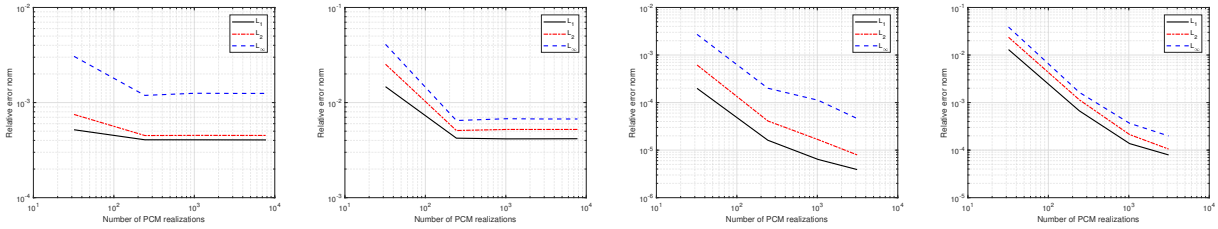
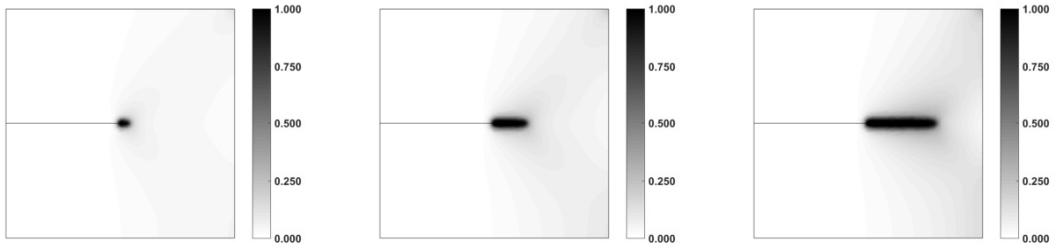


Figure 4.3 (a) and (b): Comparison between MC results with 10^4 samples, and solutions from PCM with different number of realizations in the multivariate case. With higher dimensions the advantage of PCM over MC becomes more evident, with only 3 points needed in each dimension to stabilize the error in both expectation (a) and standard deviation (b). (c) and (d): Convergence of damage field on multivariate PCM simulations of notched geometry, where the reference for error computation is a solution with 6 collocation points in each dimension. We obtain linear convergence for both expectation (c) and standard deviation (d).

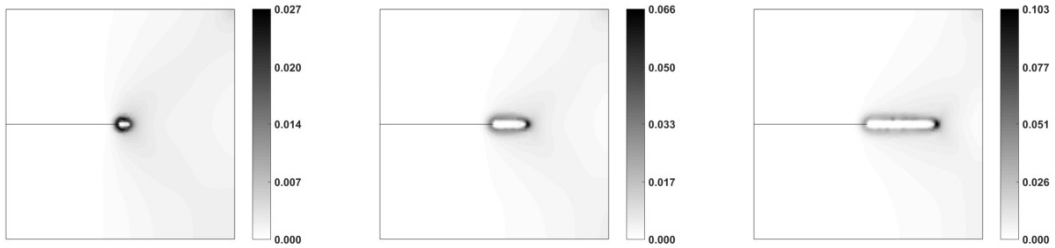
reference, in the multivariate case (considering the parameters from Table 4.1, a 5-D random space). Again, we observe that PCM quickly achieve comparable accuracy in expectation and standard deviation to 10^4 MC realizations. On Fig. 4.3 (c) and (d), we see the convergence of PCM in 5 dimensions, where the PCM benchmark is a solution with 6 points in each dimension (total of $6^5 = 7776$ realizations). The convergence rates within the PCM algorithm are again close to linear. Nonetheless, we consistently show that PCM exhibits a faster convergence than MC (a half-order increase), while requiring far less realizations for a desired accuracy.

4.4.1.2 Uncertainty and sensitivity analyses

We first investigate the univariate uncertainty propagation, where we assume that each random parameter has a uniform distribution centered at the values on Table 4.1 with 10% variation to left and right. In this 1-D parametric setting, we assume that the other parameters are deterministically known at their expected values. Fig. 4.4 shows the damage field expectation and standard deviation when we consider the phase-field layer width γ as uncertain, while keeping the other parameters fixed in PCM simulations with 4 integration points. The expected crack propagates to the right, as we could expect from a tensile load, and the uncertainty follows the crack tip. In Fig. 4.5 we plot the damage field from random γ over the crack propagation line (at $y = 50\ mm$, from



(a) Time $t = 0.3\ s$. **(b)** Time $t = 0.4\ s$. **(c)** Time $t = 0.5\ s$.



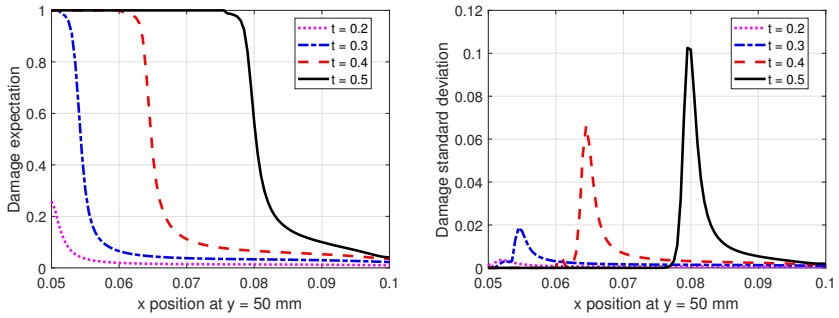
(d) Time $t = 0.3\ s$. **(e)** Time $t = 0.4\ s$. **(f)** Time $t = 0.5\ s$.

Figure 4.4 Damage phase-field expectation (top) and standard deviation (bottom) after crack propagation taking γ as random input. From tensile load the crack propagates in Mode-I as expected. γ has influence around the crack path, because it controls the diffusion of damage. Once the crack propagates and the expected value is 1 in the crack path, the uncertainty vanishes in the cracked region. However, the deviation around the crack tip grows with time.

$50\ mm \leq x \leq 100\ mm$). Looking at the damage field profiles from the expected solution we see the crack tip as a moving interface. The standard deviation's peak follows the interface and grows in time. Since γ controls the diffusivity we see that with increasing uncertainty the interface profile of the expected solution becomes less sharp with time.

For the other parameters, the damage field expectation is similar to the ones shown in Fig. 4.4. The major difference lies in the deviation field, as shown in Fig. 4.6, where we see that for the other variables the uncertainty is more concentrated around the crack tip. This example has a known crack path, so most of uncertainty is related to the speed of crack growth through the Griffith fracture energy g_c that controls how fast damage grows with intermediate values of fatigue, and the rate of change of damage c, directly affecting damage time evolution.

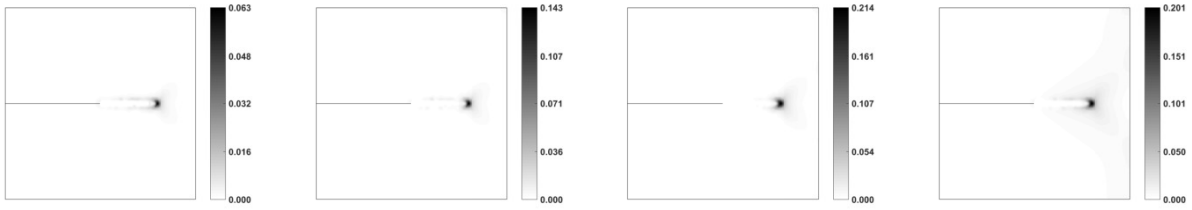
Still in the univariate framework, we use complex-step differentiation to evaluate the stochastic sensitivity of every random parameter, where the perturbation is chosen to be $h = 0.001(\xi^j_{max} - \xi^j_{min})$.



(a) Damage expectation profile.

(b) Damage standard deviation profile.

Figure 4.5 Time evolution of damage phase-field expectation and standard deviation profiles at the crack path line taking γ as random input. From damage expectation we observe the crack tip as a moving interface. The standard deviation peak follows the advecting boundary and grows in time, which makes the expected interface less sharp.



(a) Rate of change of fatigue a.

(b) Viscous damping b.

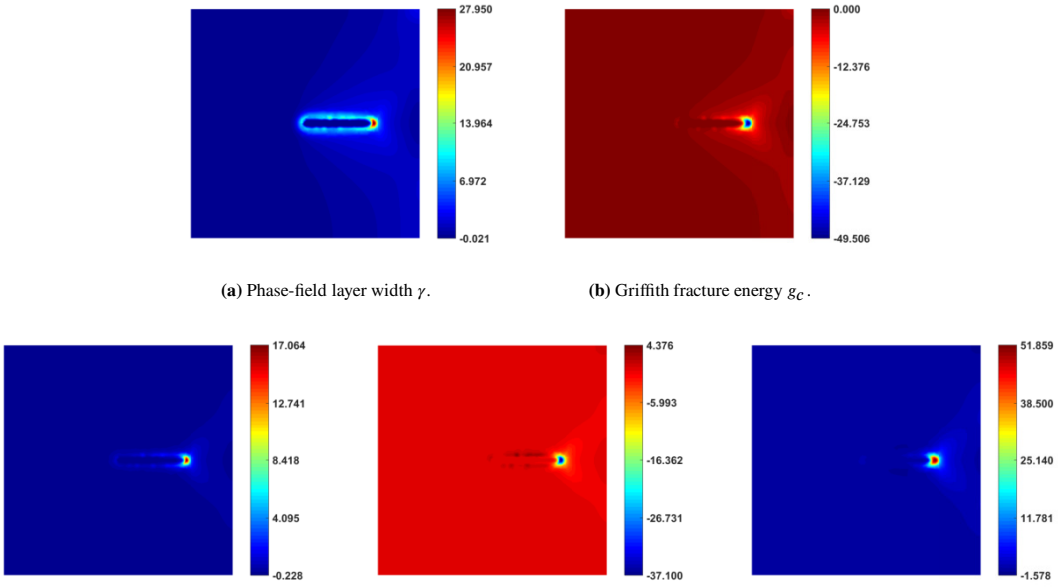
(c) Rate of change of damage c.

(d) Griffith fracture energy g_c.

Figure 4.6 Damage phase-field standard deviation after crack propagation in univariate uncertainty quantification. Fatigue parameter a and viscous damping b do not propagate uncertainty as much as Griffith energy g_c and rate of change of damage parameter c. Since the crack path is defined by geometry, the majority of uncertainty is related to the speed of crack propagation, controlled mostly by g_c and c.

Fig. 4.7 shows expectation of damage sensitivity fields at final time $T = 0.5\ s$. We make two observations here. First, the sensitivity fields show that damage increases at the crack tip when we increase γ, a and c. By increasing b we damp vibrations, thus, reducing damage. Again, the effect of g_c is visible, where increasing its value we require higher levels of fatigue to drive damage towards 1, thus the sensitivity around the crack tip is negative. Second, if we order sensitivity and standard deviation by their maximum absolute values, we obtain the same decreasing order for both: c, g_c, b, γ, a, with similar proportionality between them in the two cases. Both results evidence that with a clear crack location and path, speed of propagation drives uncertainty and sensitivity.

From the univariate uncertainty propagation and the stochastic sensitivity analysis, we identify



(a) Phase-field layer width γ.

(b) Griffith fracture energy g_c.

(c) Rate of change of fatigue a.

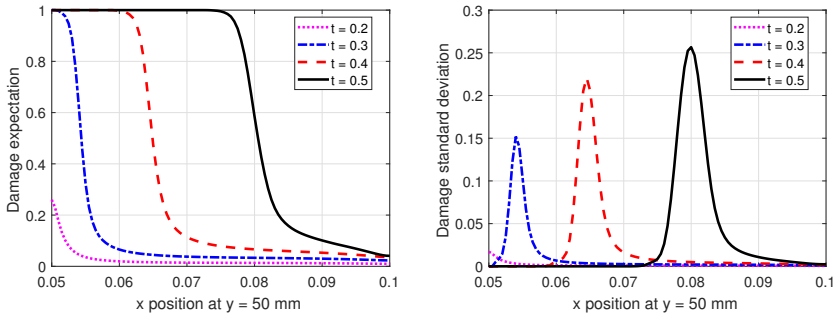
(d) Viscous damping b.

(e) Rate of change of damage c.

Figure 4.7 Expected sensitivity fields with respect to each input parameter. Similarly to standard deviation fields, local sensitivity results also point to c and g_c, related to propagation speed, as parameters with more sensitive output, since we have a specific crack initiation location and path.

the most influential parameters when we assume all others to be known. Now we show results of a 5-D PCM simulation, where we consider all parameter combinations through tensor product, using 6 points in each dimension. Fig. 4.8 shows the evolution of damage phase-field profiles at the crack path over time. The maximum standard deviation at final time (0.257) is comparable to peak uncertainty of g_c (0.201) and c (0.214).

In order to quantify the relative participation of each parameter in 5-D uncertainty, we compute the Sobol indices S^j and S_T^j from Eq. (4.24) and (4.25), respectively. Fig. 4.9 shows the total deviation field, as a reference, and sensitivity index fields S^j at final time for all parameters. Despite γ having high maximum value of its sensitivity index, we point out that most of its influence is around the crack, where the uncertainty is small. Ahead of the crack, γ has no influence in uncertainty in this case where the path is defined. We see again that at the region of high uncertainty g_c and c play an important role when we only consider their sole effects. We plot the total sensitivity index fields S_T^j in Fig. 4.10. We see that high order interactions between the



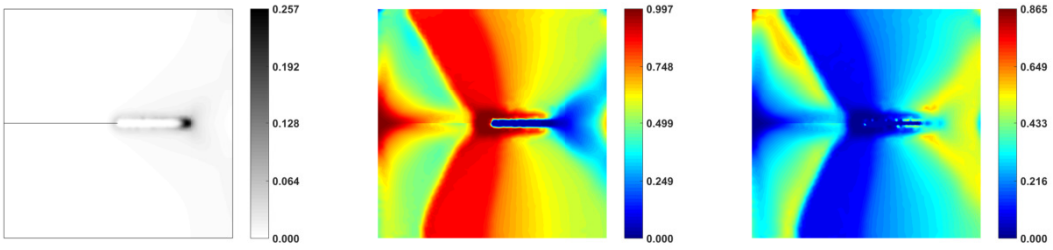
(a) Damage expectation profile.

(b) Damage standard deviation profile.

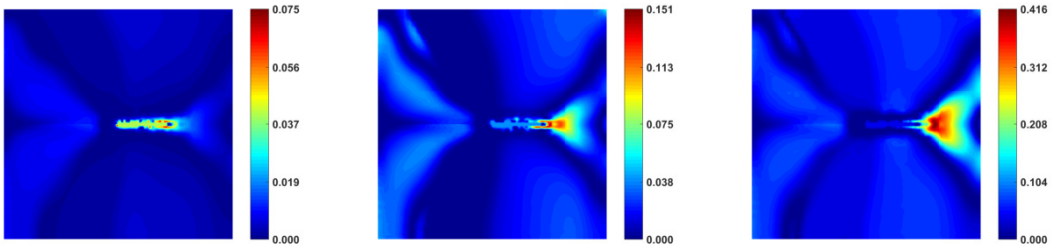
Figure 4.8 Time evolution of damage phase-field expectation and standard deviation profiles at the crack path line when propagating the uncertainty of all 5 random parameters. We observe that the combined effect of all parameters results in a larger standard deviation around the crack tip at final time, comparable to peak values of c and g_c uncertainties.

parameters have great impact over the sensitivity indices. In the majority of the geometry, the fields are uniform with base value around 0.4, except for γ. The influence of g_c and c is carried over the other parameters in the region ahead of the crack, with g_c still being the parameter that has most impact ahead of the crack.

From single-edge notched tensile test uncertainty and sensitivity analyses it is clear that the load conditions and geometric singularity define the position of crack initiation and its path. Parameters that control rate of change of damage, such as g_c or c, are more sensitive and contribute more to solution uncertainty. For general geometries and load situations, we may expect a shift in the relative contribution of total uncertainty. Take the sensitivity indices from Fig. 4.9, for example. Uncertainty that is not around the crack tip in the direction of crack propagation is dominated almost entirely by γ and g_c. In cases where the uncertainty is not concentrated around a specific region of the geometry, we can expect γ and g_c to have stronger influence. Not surprisingly, both parameters are multiplying the Laplacian in damage equation, which arises from the squared gradient, a local interaction term in the free-energy potential.



(a) Total deviation field. **(b)** Phase-field layer width γ. **(c)** Griffith fracture energy g_c.



(d) Rate of change of fatigue a. **(e)** Viscous damping b. **(f)** Rate of change of damage c.

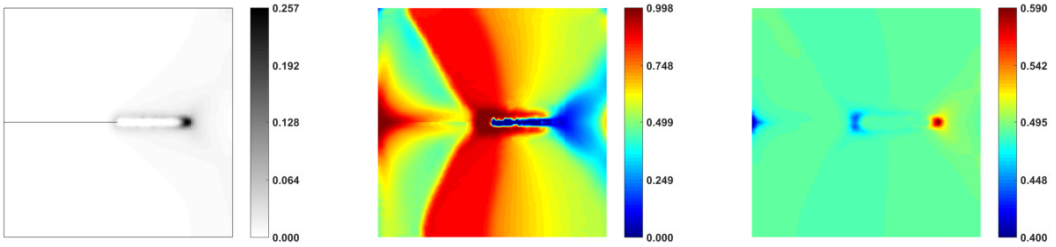
Figure 4.9 Notched tensile total damage deviation field and sensitivity indices (S^j) fields for all parameters using 6 points in each dimension at final time $T = 0.5\ s$. Ahead of the crack, g_c and c are the most influential parameters to total damage field variance. The remaining parameters have little participation at the most uncertain region of the geometry.

Table 4.2 Expected value of stochastic parameters for tensile test specimen.

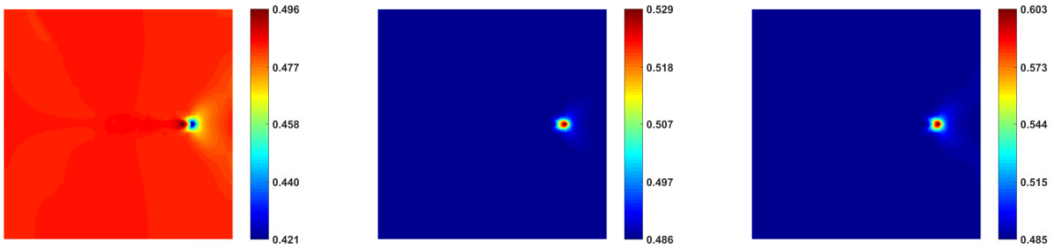
Parameter	Value
a (rate of change of fatigue)	$5 \times 10^{-7}\ m^2$
b (viscous damping)	$1 \times 10^{8}\ Ns/m^2$
c (rate of change of damage)	$2 \times 10^{-6}\ m/Ns$
g_c (Griffith energy)	$2700\ N/m$
γ (phase-field layer width)	$3 \times 10^{-4}\ m$

4.4.2 Tensile Test Specimen

The tensile test geometry without notch from Fig. 4.11 is the standard design with a mesh of 3912 nodes and 7236 linear triangle elements, where we also constrain one of its ends and apply a prescribed displacement of $4.5 \times 10^{-4}\ m/s$ until $T = 0.5\ s$, with time increments of $\Delta t = 5 \times 10^{-4}\ s$. The smallest element size in the mesh is $0.614\ mm$. We consider the same material properties and thickness from the notched case, with expected values for the stochastic parameters shown on Table 4.2.



(a) Total deviation field. **(b)** Phase-field layer width γ. **(c)** Griffith fracture energy g_c.



(d) Rate of change of fatigue a. **(e)** Viscous damping b. **(f)** Rate of change of damage c.

Figure 4.10 Notched tensile total damage deviation field and total effect sensitivity indices (S_T^j) fields for all parameters using 6 points in each dimension at final time $T = 0.5\ s$. When we combine parameter effects and include their interactions the dominant sensitivity at the crack tip gets carried out to all parameter indices. In the remaining regions, the sensitivity index is uniform except for γ: the diffusion coefficient is more influential throughout the specimen.

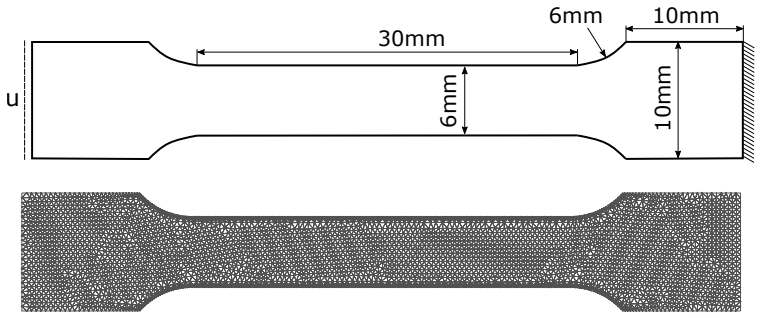
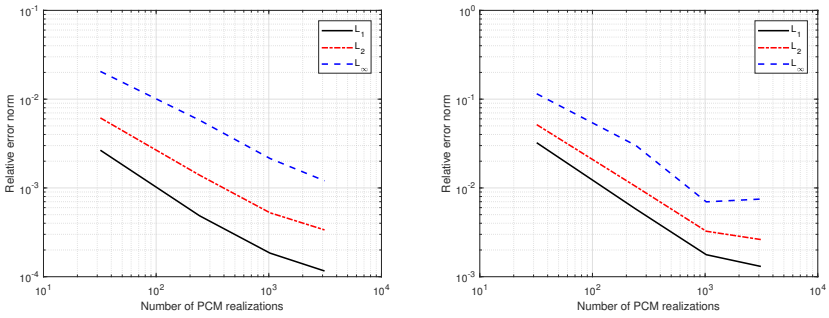


Figure 4.11 Top: Geometry and boundary conditions for tensile test specimen. Bottom: finite element mesh.



(a) Expectation. **(b)** Standard deviation.

Figure 4.12 Convergence of damage field in multivariate PCM simulations for tensile test specimen, where the reference for error computation is a solution with 6 collocation points in each dimension. We have lower convergence rates when compared to notched geometry due to due to more uncertainty of crack location.

We investigate the multivariate uncertainty propagation in the tensile test case. Fig. 4.12 shows the convergence of 5D PCM using the solution of 6 points in each dimension as a reference. Due to the absence of an initial crack, the convergence of the expected solution and deviation has a lower rate when compared to the single-edge notched test. Figs. 4.13 and 4.14 show the expectation and standard deviation of the damage field when we consider 5-dimensional PCM simulations with 6 integration points in each dimension. Unlike the notched case, here we have different expected locations for crack initiation that propagate from the surface to the interior of the body following the stress field concentration. Moreover, the final uncertainty in this case is more than 30% of the maximum damage, higher than any value from the notched case.

Fig. 4.15 shows the stochastic sensitivity fields for all parameters on the tensile test case using 8 PCM points to compute the expected sensitivity. We can observe that 3 parameters are more sensitive, namely γ, g_c and a, and their absolute range are equivalent, going from 5 to 30, approximately. We can argue that, since we do not have a preferential crack path nor a specific crack initiation position, all the uncertain parameters have the same sensitivity. The c parameter in this case is not as sensitive compared to the notched geometry, since here the crack location and path are not defined, so damage increase rate becomes less important compared to crack position. Last, we can see that b has little sensitivity on damage field in this case.

(a) $t = 0.46\ s$.

(b) $t = 0.48\ s$.

(c) $t = 0.5\ s$.

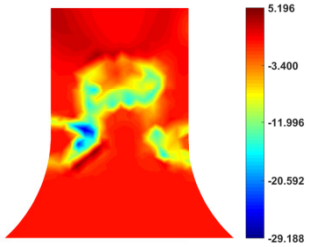
Figure 4.13 Damage phase-field expectation taking all parameters in ξ as random inputs. From tensile load we see the appearance of 4 possible crack initiation points, based on the stress concentration profile from the geometry. The expected solution at final time gives a curved crack path at both sides of the geometry.

(a) $t = 0.46\ s$.

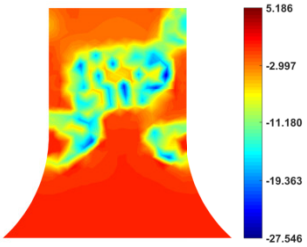
(b) $t = 0.48\ s$.

(c) $t = 0.5\ s$.

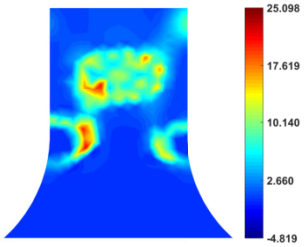
Figure 4.14 Damage phase-field deviation taking taking all parameters in ξ as random inputs. We have regions of uncertainty around all 4 points of possible crack initiation. At final time, the the uncertainty vanished where the crack propagated, and the maximum deviation around the crack is more than 30% of maximum damage.



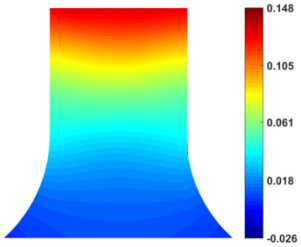
(a) Phase-field layer width γ.



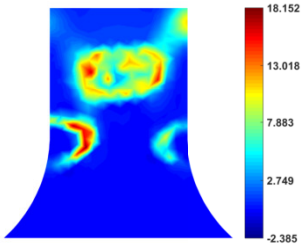
(b) Griffith fracture energy g_c.



(c) Rate of change of fatigue a.

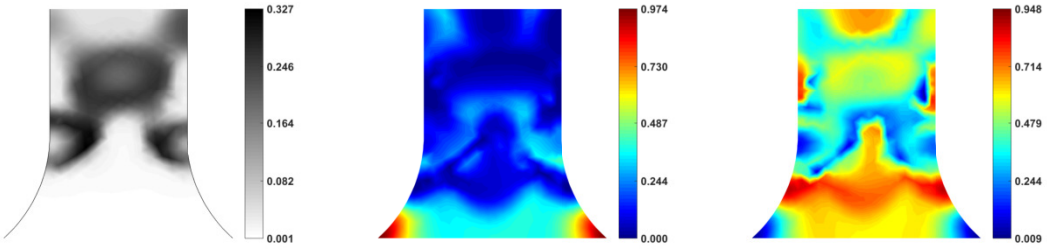


(d) Viscous damping b.

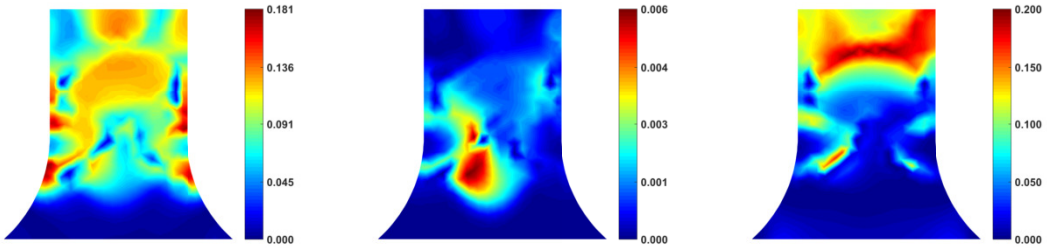


(e) Rate of change of damage c.

Figure 4.15 Local sensitivity expectation fields with respect to each input parameter.γ, g_c and a are the most sensitive parameters, with the same absolute range. b is not sensitive in the range considered and c is less sensitive than in the notched case.



(a) Total deviation field. **(b)** Phase-field layer width γ. **(c)** Griffith fracture energy g_c.



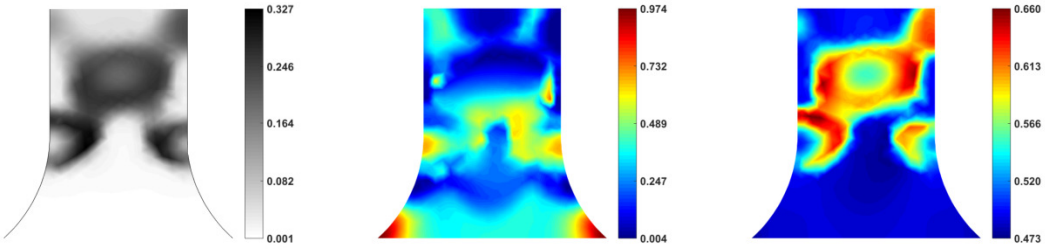
(d) Rate of change of fatigue a. **(e)** Viscous damping b. **(f)** Rate of change of damage c.

Figure 4.16 Tensile test specimen total damage deviation field and sensitivity indices (S^j) fields for all parameters using 6 points in each dimension at final time $T = 0.5\ s$. Differently than in the notched case, here γ and g_c are the most influential parameters in the region of higher uncertainty.

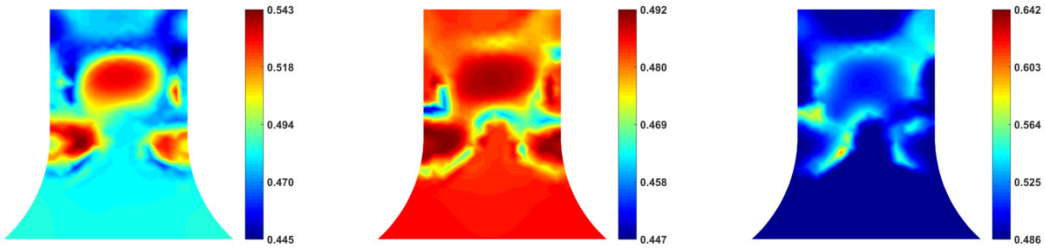
Another observation is that, contrary to the notched case, here the sensitivity of γ around the crack is negative, being positive elsewhere. Since we do not have a defined crack path, the increase of diffusion coefficient should smoothen the field. In other words, this case makes the γ associated with the Laplacian in the damage equation more sensitive than γ parameters in the remaining terms.

We present the total uncertainty field and sensitivity indices S_j for the tensile test specimen in Fig. 4.16. Similarly to the notched case, here we see that the order of most influential parameters to the uncertainty regions are equivalent to the most sensitive parameters from Fig. 4.15. In the regions of high uncertainty around the possible crack paths, the deviation field is influenced most by γ and g_c. Fatigue parameter a comes next, with relatively less importance than what was found in the local sensitivity analysis. Viscous damping b has little participation in total variance, while damage rate c has less importance when compared to the notched case. In summary, participation in regions of high uncertainty is dominated by γ and g_c.

Total sensitivity indices S_T^j for tensile specimen are presented in Fig. 4.17. We can observe that



(a) Total deviation field. **(b)** Phase-field layer width γ. **(c)** Griffith fracture energy g_c.



(d) Rate of change of fatigue a. **(e)** Viscous damping b. **(f)** Rate of change of damage c.

Figure 4.17 Tensile test specimen total damage deviation field and total effect sensitivity indices (S_T^j) fields for all parameters using 6 points in each dimension at final time $T = 0.5\ s$. With the combined effect of all parameters, we still have γ and g_c as having most influence in the uncertainty regions.

when we include interactions between the parameters, a, b and c present sensitivity fields that are almost uniform between 0.44 and 0.5. The other 2 parameters, γ and g_c also present uniform fields, except for the regions around the crack where the total sensitivity indices are higher than 0.6.

From these examples it is clear that geometry and the existence of initial cracks, notches or singularities in the geometry plays an important role on sensitivity and uncertainty. In the notched case, the existence of a crack and the application of tensile stress makes the crack propagate straight to the right. The only uncertainty remains with the speed of damage increase ahead of the crack path, hence the sensitivities of g_c and c.

In the tensile specimen without notch on singularity, the uncertainty is driven by γ, g_c and a. From Eq. 4.1, γ and g_c are multiplying the local interaction term, namely $g_c\frac{\gamma}{2}|\nabla\varphi|^2$ which later becomes the Laplacian in damage evolution, $\frac{\gamma(\omega)g_c(\omega)}{\lambda}\Delta\varphi$. Moreover, they are also associated with the tensor product on the equation of motion, also originated from the local interaction, and are associated to fatigue potentials $\mathcal{H}$ and $\mathcal{H}_f$, arbitrarily chosen. The complete symmetry of the tensile

specimen geometry with no surface roughness or material imperfections, associated with the local operators and *ad hoc* modeling, leads to potential crack appearances in the 4 stress concentration regions around the fillets. Any numerical effect such as artificial damping or residuals may lead to perturbation in the solution and crack initiation at any corner.

Furthermore, the traditional disregard of nonlocal interactions in phase-field models excludes the possibility of modeling many phenomena experimentally observed, such as intermittent dislocation avalanches [11–13, 181] and fractal characteristics of fracture[4, 118], which show scale-free distributions and power-law scaling, that can be successfully modeled through fractional calculus [182]. Recent works have addressed fractional-order Cahn-Hilliard equation [183, 184]. Phase-field models derived from free-energy potentials with nonlocal effects were first discussed by Giacomin and Lebowitz [185, 186] with recent contributions from Abels et al [187], and Ainsworth and Mao [184]. Fractional-order models for structural analysis have been developed [46], to which corresponding fractional uncertainty/sensitivity analyses can be formulated via operator-based uncertainty quantification [188] and Fractional Sensitivity Equation Method (FSEM) [189].

CHAPTER 5

DATA-DRIVEN FAILURE PREDICTION IN BRITTLE MATERIALS: A PHASE-FIELD BASED MACHINE LEARNING FRAMEWORK

5.1 Introduction

Predictability is essential to any mathematical model for failure and fracture. From early linear elastic fracture mechanic models from [2], to failure analysis through damage mechanics by [153], numerical models have improved in scope and complexity to provide realistic simulations of material failure to meet industry goals of safety, and to reduce component weight and production costs. The accurate simulation of the failure process, from crack initiation to propagation until final failure, in a consistent way, while respecting the physics and developing robust numerical methods, is still a challenging task.

During the last decade, phase-field models have been successfully established as a powerful tool in the study of damage and fatigue. By modeling sharp interfaces through smooth continuous fields, the dynamics of moving boundaries using phase-fields emerged in diverse physical applications, including fluid separation [54], solidification [55], tumor growth [56], two-phase complex fluid flow [57], and fluid-structure interaction [58]. In failure analysis, crack sharpness is modeled through a smooth phase-field indicating the state of the material among fractured, virgin, and intermediate damaged zones, evolving through Allen-Cahn type equations. Examples ranging from brittle [31, 156, 157], ductile [32, 158], and dynamic fracture [33, 159] successfully described phenomenological effects such as crack initiation, branching and coalescence. The inclusion of fatigue effects was initially attempted with Ginsburg-Landau free-energy potentials [160] and fractional derivatives [161]. A more general framework for damage and fatigue was later developed in a non-isothermal and thermodynamically consistent approach [34, 162, 163], followed by the emergence of further phase-field models for fatigue [190, 191]. Within this myriad of different models, solution uncertainty and parametric sensitivity are still influential, and the predictability

of phase-field models for arbitrary conditions is yet a withstanding effort [104]. One promising approach to address the predictability of numerical models is to use artificial intelligence (AI), which has been consistently expanding its applicability over the years.

AI and machine learning (ML) have been widely used in different engineering applications such as structural health monitoring [192–202] and fatigue crack detection [203–205]. ML algorithms, in the context of failure analysis, have been used for numerous applications, including phase-field models of polymer-based dielectrics [206], phase-field models of solidification [207], and crystal plasticity [208]. Another interesting application of ML is to obtain a data-driven representation of free-energy potentials in the atomic scale and upscale it to a phase-field model, using Integrable Deep Neural Networks [209]. Specifically for brittle failure, ML has been recently used to build surrogate models based on explicit crack representation [210], and in failure prediction using a discrete crack representation model for high-fidelity simulations that feed an artificial neural networks (ANN) algorithm [211, 212]. Nonetheless, the noted studies have only shown the applicability of ML in failure analysis. Therefore, ML methods have not yet been explored in the context of phase-field models for damage. It is noted that the use of ML leads to a new paradigm of phase-field modeling, where we establish the basis for novel data-driven frameworks, allowing systematic infusion of statistical information and corresponding uncertainty propagation from micro-scale models and experiments into continuum macroscopic failure models.

In this work, we develop an ML algorithmic framework for failure detection and classification merging a pattern recognition (PR) scheme and ML algorithms applied to a damage and fatigue phase-field model. We consider an isothermal, linear elastic and isotropic material under the hypothesis of small deformations and brittle fracture. We simulate the phase-field model using Finite Element Method (FEM) and a semi-implicit time-integration scheme to generate time-series data of damage phase-field φ and degradation function $g(\varphi) = (1-\varphi)^2$ from virtual sensing nodes positioned at different locations across a test specimen. We introduce a PR scheme as part of the ML framework, in which time-series data from FEM node responses are considered as a pattern with a corresponding label. We define multiple labels for "no failure", "onset of failure" and

"failure" of the test specimen based on tensile test load-displacement curve and damage threshold concept. Once the patterns representing different states of the material are identified, the proposed ML framework employs k-nearest neighbor (k-NN) and ANN algorithms to detect the presence and location of failure using such patterns. The use of ML algorithms makes the failure prediction framework practical even in cases of complex loading conditions, beyond the canonical example of monotonic loads presented here.

In this study, we consider different failure types to further assess the performance of the framework. In addition, by introducing noise to the time-series data, we ascertain the robustness of the proposed framework with noise-polluted data, leading to the effective use in failure analysis under high sensitive/uncertain parameters and operators. The idea of propagating uncertainty through the solution, combined with the inherent nonlinearity of the damage phase-field model make the choice of ANN particularly attractive in the construction of the framework. The findings from this study will pave a way for the development of novel data-driven failure prediction frameworks, which are able to efficiently establish a link among the classification results (i.e., accuracy) and different phase-field model parameters, thus enabling the computational framework to identify those parameters affecting model's accuracy and updating them to achieve the best performance.

This chapter is organized as follows: in 5.2 we present the damage and fatigue phase-field model, which is used to generate time-series data for the ML framework. We introduce the data generation procedure and corresponding label definitions in 5.3. We present the ML framework in 5.4, where we describe the integration of a pattern recognition scheme with the applied classification algorithms, k-NN and ANN. We present and discuss the numerical results in 5.5.

5.2 Damage and Fatigue Phase-Field Model

5.2.1 Governing Equations

We consider a isothermal phase-field framework for structural damage and fatigue, modeled by a system of coupled differential equations for the evolution of displacement $\boldsymbol{u}$, velocity $\boldsymbol{v} = \dot{\boldsymbol{u}}$, damage φ and fatigue $\mathcal{F}$. The damage phase-field φ describes the volumetric fraction of degraded

material, and takes $\varphi = 0$ for virgin material, $\varphi = 1$ for fractured material, varying between those states, $0 \leq \varphi \leq 1$, as a damaged material. The evolution equation for the damage field is of Allen-Cahn type since the damage and aging effects are non-conservative and non-decreasing, and is derived along with the equations of motion for $\boldsymbol{u}$ and $\boldsymbol{v}$ through the principle of virtual power and entropy inequalities with thermodynamic consistency [34]. The fatigue field $\mathcal{F}$ is associated to the presence of micro-cracks, and is treated as an internal variable, whose evolution equation is obtained through constitutive relations that must satisfy the entropy inequality for all admissible processes. The geometry is defined over a spatial domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, at time $t \in (0, T]$.

The final form of the governing equations will be defined by the choice of free-energy potentials related to elasticity, damage and fatigue. We consider a linear elastic isotropic material, where the phase-field free-energy takes the usual gradient form:

$$\Psi(\boldsymbol{E}, \varphi, \mathcal{F}) = \frac{1}{2}(1-\varphi)^2 \boldsymbol{E}^T C \boldsymbol{E} + g_c \frac{\gamma}{2} |\nabla \varphi|^2 + \mathcal{K}(\varphi, \mathcal{F}), \tag{5.1}$$

where $\boldsymbol{E} = \nabla^S \boldsymbol{u}$ is the strain tensor, where $\nabla^S \boldsymbol{q} = \text{sym}(\nabla \boldsymbol{q})$ represents the symmetric part of the gradient of a given vector field $\boldsymbol{q}$. Also, C is the elasticity tensor written in terms of the Young modulus E and Poisson coefficient ν, g_c is the Griffith energy, $\gamma > 0$ is the phase-field layer width parameter, and $\mathcal{K}(\varphi, \mathcal{F})$ is a function that models the damage evolution due to fatigue effects. The first term in 5.1 represents the degraded elastic response, modeled by the choice of degradation function $g(\varphi) = (1-\varphi)^2$. The final set of governing equations, defined over $\Omega \times (0, T]$, becomes:

$$\begin{cases} \dot{\boldsymbol{u}} = \boldsymbol{v}, \\ \dot{\boldsymbol{v}} = \text{div}\left((1-\varphi)^2 \dfrac{C}{\rho} \boldsymbol{E}\right) + \dfrac{b}{\rho} \text{div}\,(\boldsymbol{D}) - \dfrac{\gamma g_c}{\rho} \text{div}\,(\nabla \varphi \otimes \nabla \varphi) + \boldsymbol{f}, \\ \dot{\varphi} = \dfrac{\gamma g_c}{\lambda} \Delta \varphi + \dfrac{1}{\lambda}(1-\varphi) \boldsymbol{E}^T C \boldsymbol{E} - \dfrac{1}{\lambda \gamma}[g_c \mathcal{H}'(\varphi) + \mathcal{F} \mathcal{H}_f'(\varphi)], \\ \dot{\mathcal{F}} = -\dfrac{\hat{F}}{\gamma} \mathcal{H}_f(\varphi), \end{cases} \tag{5.2}$$

subjected to appropriate initial and boundary conditions, which depend on the physical problem. Either displacement or stress are known at the boundaries, in addition to considering $\nabla \varphi \cdot \boldsymbol{n} = 0$ on $\partial\Omega$. Moreover, the $\otimes$ operator denotes the outer product, the infinitesimal strain rate tensor is represented by $\boldsymbol{D} = \nabla^S \boldsymbol{v}$, and parameters b and ρ are the material's viscous damping and density,

respectively. We construct λ such that the rate of change of damage increases with damage (see e.g., [153]):

$$\frac{1}{\lambda} = \frac{c}{(1+\delta-\varphi)^{\varsigma}}, \tag{5.3}$$

where $c,\ \varsigma > 0$ are material dependent, and $\delta > 0$ is a small constant to avoid numerical singularity.

The potentials $\mathcal{H}(\varphi)$ and $\mathcal{H}_f(\varphi)$ model the damage transition from 0 to 1 as fatigue changes from zero to g_c. We take their (ordinary) derivatives with respect to φ to obtain potentials $\mathcal{H}'(\varphi)$ and $\mathcal{H}'_f(\varphi)$. Further details on fatigue potentials can be found in [34]. Choosing the transition to be continuous and monotonically increasing, suitable choices for the potentials are:

$$\mathcal{H}(\varphi) = \begin{cases} 0.5\varphi^2 & \text{for } 0 \le \varphi \le 1, \\ 0.5 + \delta(\varphi - 1) & \text{for } \varphi > 1, \\ -\delta\varphi & \text{for } \varphi < 0. \end{cases} \tag{5.4}$$

$$\mathcal{H}_f(\varphi) = \begin{cases} -\varphi & \text{for } 0 \le \varphi \le 1, \\ -1 & \text{for } \varphi > 1, \\ 0 & \text{for } \varphi < 0. \end{cases} \tag{5.5}$$

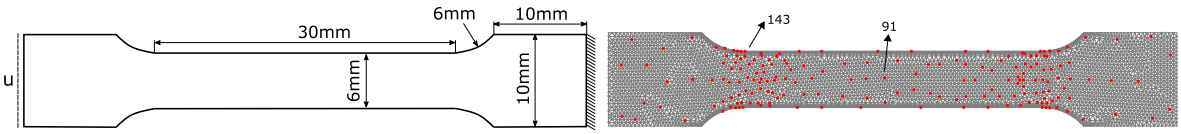
The evolution of fatigue $\mathcal{F}$ is controlled by $\hat{F}$, related to the formation and growth of micro-cracks that occur in cyclic loadings. We note that being a measure of energy accumulated in the microsctructure, fatigue variable $\mathcal{F}$ grows even under monotonic loading. The form of $\hat{F}$ depends on the absolute value of the power related to stress in the virgin material:

$$\hat{F} = a(1-\varphi)\left|(\boldsymbol{C}\boldsymbol{E} + b\boldsymbol{D}) : \boldsymbol{D}\right|, \tag{5.6}$$

where the parameter a in this case is chosen to give a linear dependence of the power of stress.

5.2.2 Discretization

We discretize 5.2 in space using linear finite element method (FEM), where the semi-discrete form is obtained through Galerkin method. For detailed derivation of the spatial discretization in 2D,



(a) Geometry and boundary conditions. **(b)** Mesh and virtual sensor nodes.

Figure 5.1 Description of geometry and boundary conditions for the tensile test specimen, along with finite element mesh and sensor layout for time-series generation. We highlight two sensor nodes that show different time-series behaviors.

please refer to Appendix A.

We consider the tensile test specimen without notch depicted in Fig. 5.1. We discretize it with a finite element mesh consisting of 3912 nodes and 7236 linear triangle elements, with smallest element size of 0.614 mm. We constrain one end, and apply a prescribed displacement of $4.5 \times 10^{-4}\ m/s$, with time increments of $\Delta t = 5 \times 10^{-4}\ s$, at the other end. We study a material with Young modulus $E = 160\ GPa$, Poisson coefficient $\nu = 0.3$, and density $\rho = 7800\ kg/m^3$, under plane stress conditions with thickness of $h = 5\ mm$. The rate of change of fatigue a is $5 \times 10^{-7}\ m^2$, and viscous damping b is $1 \times 10^8\ Ns/m^2$. The remaining parameters γ (phase-field layer width), g_c (Griffith energy), and c (rate of change of damage), are chosen in order to construct a set of different representative cases. We focus on those parameters to build the cases because they are the most sensitive and give more uncertainty in damage evolution [104].

5.3 Data Processing

In this section, we highlight how to obtain time-series data from phase-field simulations to train and test the learning algorithms. Further, we explore different possibilities of label definitions in the context of failure prediction based on the simulation results.

5.3.1 Time-Series Data Generation

To generate time-series data, virtual sensing nodes are considered at different locations of the specimen, as shown in Fig. 5.1. The sensor layout in our tests simply chosen to provide a coarse-to-fine (variable) resolution for the ML framework to calibrate/train and classify time-series data.

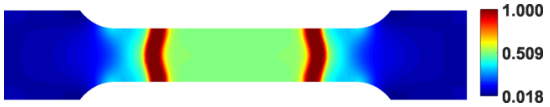
Table 5.1 presents the parameters used to construct each representative failure case or type, for

Table 5.1 Parameters used in the representative cases.

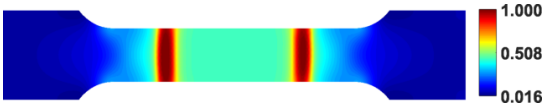
Case	$\gamma\ (m)$	$g_c\ (N/m)$	$c\ (\frac{m}{Ns})$
1	3.00×10^{-4}	2700	2.00×10^{-6}
2	2.00×10^{-3}	2700	2.00×10^{-6}
3	5.00×10^{-4}	5400	2.00×10^{-6}
4	5.00×10^{-4}	10800	2.00×10^{-6}
5	2.00×10^{-3}	5400	1.00×10^{-6}
6	2.50×10^{-4}	5400	1.00×10^{-6}

(a) Case 1, $t = 0.48\ s$.

(b) Case 2, $t = 0.48\ s$.



(c) Case 3, $t = 0.62\ s$.



(d) Case 4, $t = 0.78\ s$.

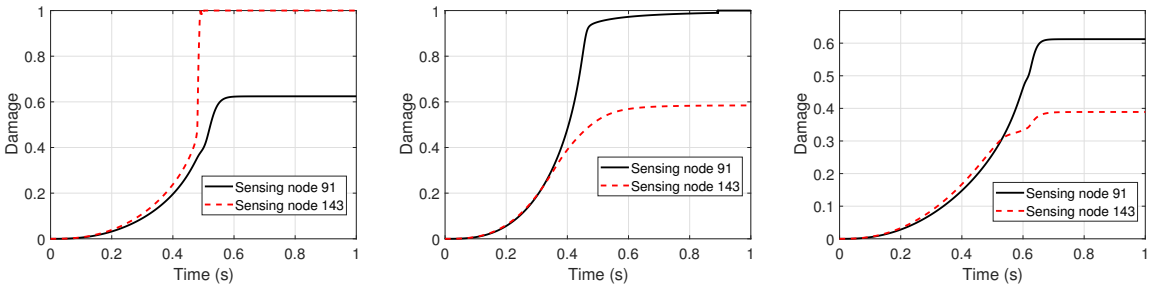
(e) Case 5, $t = 0.62\ s$.

(f) Case 6, $t = 0.64\ s$.

Figure 5.2 Damage phase-field for each representative failure case. By changing the parameters γ, g_c, and c, we observe different failure types (distinct crack positions and paths), as well as varying dynamics.

which we plot the damage phase-field at failure time in Fig. 5.2. We also observe three different failure types (i.e., around the fillets, at the middle of the specimen, and in an intermediate region between those), where the effect of changing parameters is clearly noticeable. We observe the different damage evolution in the highlighted sensor nodes shown in Fig. 5.1 from their time-series data. We plot time-series data from sensor nodes 91 (at the middle of the specimen) and 143 (at one of the fillets), for cases 1, 2 and 3, where we observe the different evolution profiles based on each failure type (see Fig. 5.3).

From the damage phase-field time-series data φ at sensing nodes, we then form a feature vector of a pattern as the degradation function $g(\varphi) = (1 - \varphi)^2$. Thus, patterns are generated using $g(\varphi)$, extracted at sensing nodes for a given time step. Accordingly, a label is assigned to identified



(a) Case 1. **(b)** Case 2. **(c)** Case 3.

Figure 5.3 Damage phase-field time-series data for three cases, showing the different evolution of φ depending on the virtual sensor node position.

patterns at each time step.

Remark 5.3.1. *Damage φ is a proper measure of material failure. However, by defining the feature vector based on the degradation function $g(\varphi)$ instead, we directly measure the material softening, since $g(\varphi)$ is the field variable that degrades the constitutive model, thus reducing the component's load bearing capabilities.*

5.3.2 Label Definitions

In the domain of ML, label is defined as the output of the classification algorithm. We outline different criteria used to generate the labels for the supervised ML algorithms. In the context of failure analysis, labels should reflect the material's capacity to withstand loads, so a first rational choice is to define labels based on load-displacement curve. Besides, we generate labels based on a damage threshold concept with degradation function $g(\varphi)$. Based on noted criteria, each pattern is given a label corresponding to one of multiple classes, namely, no failure (class 1), onset of failure (class 2), and failure (class 3).

5.3.2.1 Label definition according to load-displacement curve

We start by defining the labels in a binary fashion, in order to observe the damage phase-field corresponding to a specific label transition. At each time step, a pattern is assigned 0 if there is no failure,

and 1 if the specimen has fractured, such that we have a label vector $L = \{0\ 0\ \ldots\ 0\ 1\ \ldots\ 1\ 1\}^T$. We assign the labels based on load-displacement curve of the tensile test:

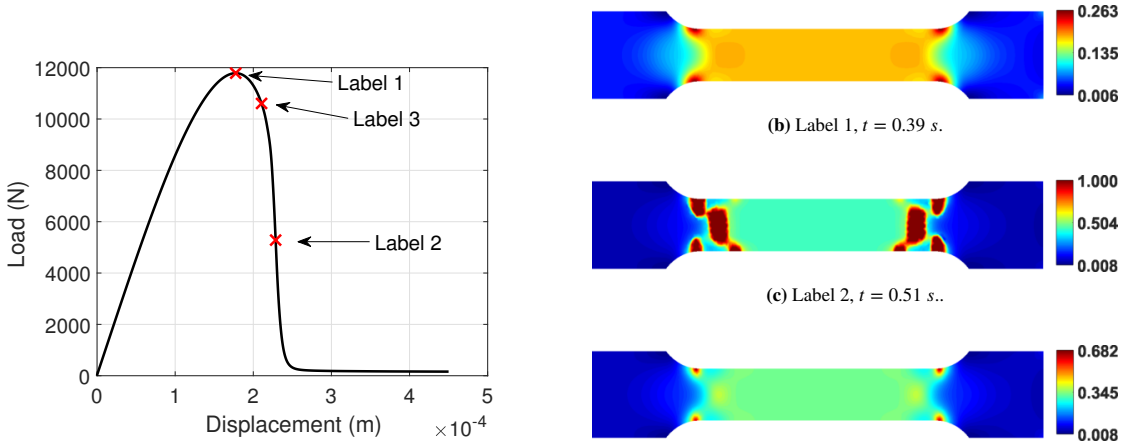
- **Label Type 1**: labels are generated based on the maximum force at the load-displacement curve, which may induce to a failure criterion too soon.
- **Label Type 2**: labels are generated according to the minimum derivative df/du, which could detect damage too late.
- **Label Type 3**: labels are generated based on 85%, 90%, and 95% of maximum force at the load-displacement curve, which yields an intermediate behavior compared to Label types 1 and 2.

Fig. 5.4 illustrates the different points, where failure is defined according to the label types, with corresponding damage phase-fields. The label types 1 and 2 are too extreme and lead to early and late prediction of failure, respectively. To address this issue, we further improve binary label type 3 by including an intermediate state, the onset of failure, based on percentages of maximum load. Label type 3 then becomes a multiple label definition, stated here as:

- **Multiple Label Type 3**: given the labels created based on 90% and 95% of maximum force at the load-displacement curve (l_{90} and l_{95}), a pattern x_i is assigned to class 1 (no failure) if label of the pattern based on l_{90} and l_{95} is zero, class 2 (onset of failure) when label of the pattern is zero based on l_{90} and one based on l_{95}, and class 3 if label of the pattern based on l_{90} and l_{95} is one.

5.3.2.2 Label definition according to damage threshold concept

We also propose a label definition based on a damage threshold of degradation function $g(\varphi)$, where three different thresholds (i.e., R_1=1, R_2=0.92, and R_3=0.85) are empirically selected based on the simulations. Accordingly, we generate labels by tracking $g(\varphi)$ on all sensing nodes, and following the rule:



(a) Load-displacement curve.

(b) Label 1, $t = 0.39\ s$.

(c) Label 2, $t = 0.51\ s$..

(d) Label 3, $t = 0.47\ s$..

Figure 5.4 (Left) Load-displacement curve for case 1, where we identify the three points where the labels change from 0 to 1, according to different criteria. (Right) Respective damage phase-fields corresponding to the positions indicated in the curve. We note that Label type 3, based on a threshold of 90% of maximum force, lies between the first two criteria. In Label 1, damage field is still too smooth, while in Label 2, failure is far too advanced.

- **Multiple Label Type 4**: a given sensor node S_i is shown with index a when $R_1 \geq g(\varphi) > R_2$, index b if $R_2 \geq g(\varphi) > R_3$, and index c if $R_3 \geq g(\varphi)$. Once the noted indices for all sensor nodes at a given time step (i.e., features of pattern x_i) are determined, sum of each index a to c is computed. Pattern x_i is then classified as class 1, if summation of index a is larger than that of indices b and c, class 2 when summation of index b is greater than summation of indices a and c, etc. This label definition is motivated by the neighboring effect concept (i.e., group of sensor nodes), allowing to eliminate the effect of uncertainty and faulty sensor.

5.4 ML Algorithmic Framework

We develop a supervised ML algorithmic framework for interpretation of time-series data generated from the phase-field model. The proposed ML framework presented in Fig. 5.5 is based on the integration of a PR scheme and ML algorithms. According to PR scheme, sensor nodes responses (i.e., time-series data of degradation function $g(\varphi) = (1-\varphi)^2$) at each time step are represented as a pattern, along with corresponding label. The input to the learning framework is thus a matrix M

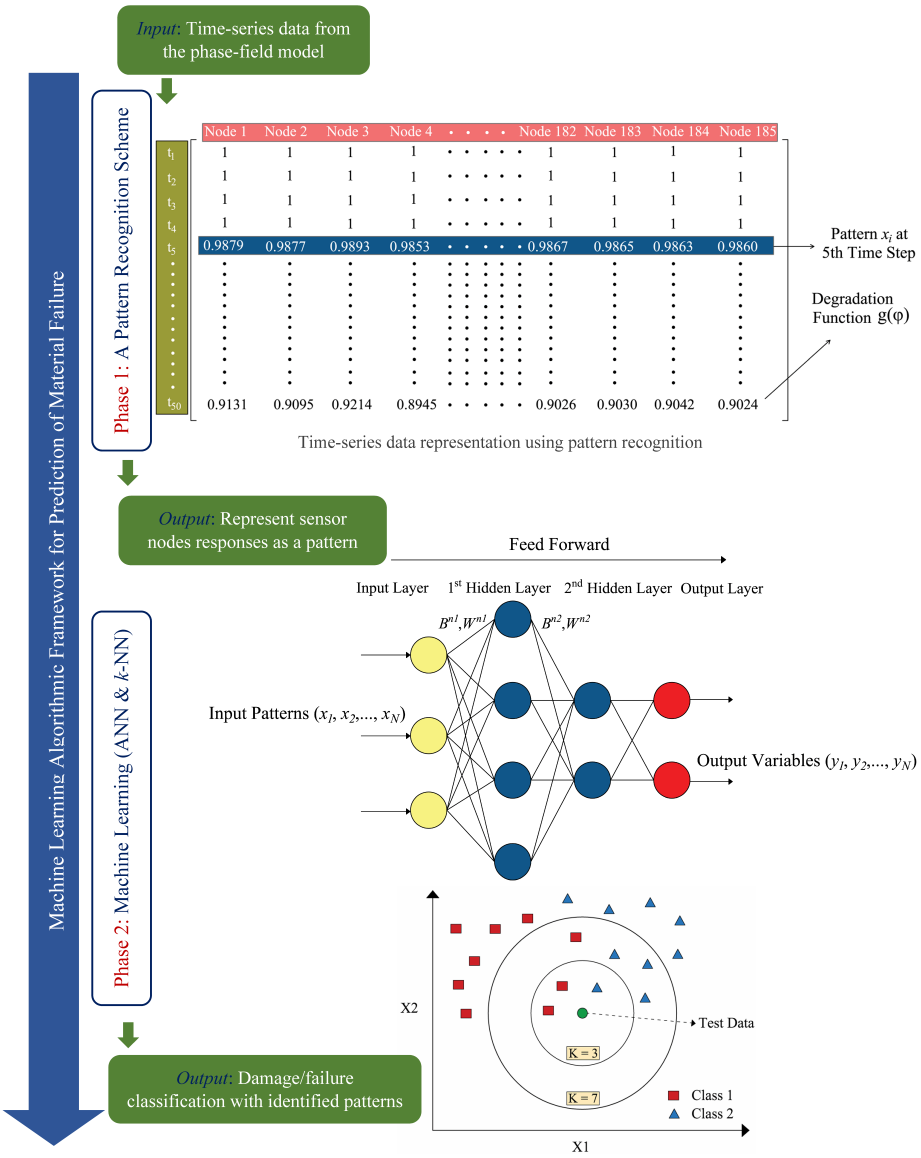


Figure 5.5 Schematic illustration of the proposed ML framework. A pattern recognition scheme is introduced to represent time-series data of damage degradation function $g(\varphi) = (1-\varphi)^2$ extracted at sensing nodes as a pattern. The k-NN and ANN algorithms are employed for failure classification using recognized patterns. In k-NN analysis the classification is performed by determining the k-nearest vote vector. An ANN provides a map between the inputs and outputs through determination of the weights using input and output patterns.

with dimension $m \times n$, where m denotes the number of time steps and n represents the number of sensor nodes. Consequently, each row of the matrix denotes a pattern, where the dimension of the PR problem is n (i.e., a pattern with n features).

After generating the noted matrix M, ML algorithms, k-NN and ANN, are used for fail-

ure/damage classification with m identified patterns. By introducing ML algorithms for pattern recognition, as opposed to relying exclusively on the load-displacement curve traditionally employed in failure mechanics, we allow the framework to be generalized, allowing failure detection in complex loading conditions, not restricted to the canonical examples presented, and in cases where load-displacement data is not even present. For the same reason, we underline the importance of Multiple Label Type 4, which exclusively depends on damage sensor data.

It should be noted that k-NN and ANN are used in this research due to their effective and reliable performance, while these algorithms are computationally efficient. ANN is specially advantageous in this application, where we have a phase-field damage model that is nonlinear, and associated with salient sources of parametric and model-form uncertainties which could be broadcast to the solution. Therefore, we use ANN over other comparable methods to permit a systematic study of uncertainty propagation and sensitivity of failure detection under noisy data in future works. The theoretical and mathematical details of k-NN and ANN can be found in the published literature [213–216].

The dataset for the k-NN and ANN analysis is divided into three subsets; namely, training, validation, and test. The training set is used to fit the ML classifiers, while the validation set is used to compute the optimal learning parameters. Performance of the ML classifiers with optimal parameters is then assessed on the test set. The performance of k-NN and ANN algorithms is measured using the detection performance rate defined in following equation:

$$\text{Classification accuracy} = \frac{\text{Number of patterns correctly classified}}{\text{Total number of identified patterns}}. \tag{5.7}$$

Different size of data subsets are considered herein to evaluate the effect of such factor on the performance of the ML algorithms. Accordingly, five different combinations listed below are defined, where the accuracy of k-NN and ANN is determined based on each combination.

- Comb 1: training & validation 65% and test 35%.
- Comb 2: training & validation 70% and test 30%.
- Comb 3: training & validation 75% and test 25%.
- Comb 4: training & validation 80% and test 20%.
- Comb 5: training & validation 85% and test 15%.

5.5 Results and Discussion

The performance of the developed ML framework in terms of predicting the presence and location/pattern of failure is evaluated with time-series data of degradation function $g(\varphi) = (1 - \varphi)^2$ generated from the phase-field model. To this aim, the framework is initially trained and tested using each one of six representative failure cases (see 5.3.1), where the presence of failure is detected for each case, along with corresponding accuracy. In the next analysis phase, to detect the location of failure, ML algorithms (i.e., k-NN and ANN) are trained using data from all six failure cases, and the classification accuracy is determined on test data, leading to identification of the pattern of failure. The following subsections present the classification results of the ML framework employing multiple labels generated according to multiple label types 3 and 4 (see 5.3.2).

5.5.1 Results with k-NN

5.5.1.1 Detection of the Presence of Failure

As noted in 5.4, the proposed ML framework is trained and tested using different size of data subsets. For the k-NN analysis, k-fold cross validation with $k = 10$ is used. Patterns identified with the PR scheme, along with corresponding labels, are used as input to the algorithmic framework in order to predict the presence of failure. To further explore the performance of the k-NN algorithm, the optimal number of k needs to be determined. In this context, cases 1 to 3 representing

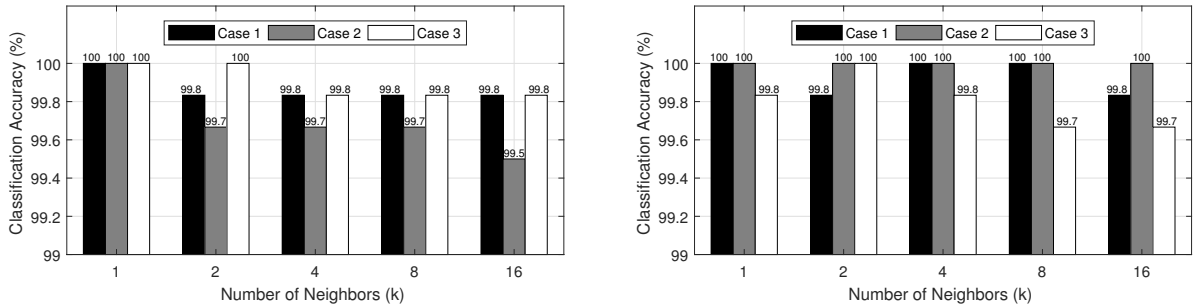


Figure 5.6 K-NN classification accuracy with different number of k: (a) Accuracy based on multiple label Type 3, (b) Accuracy based on multiple label Type 4.

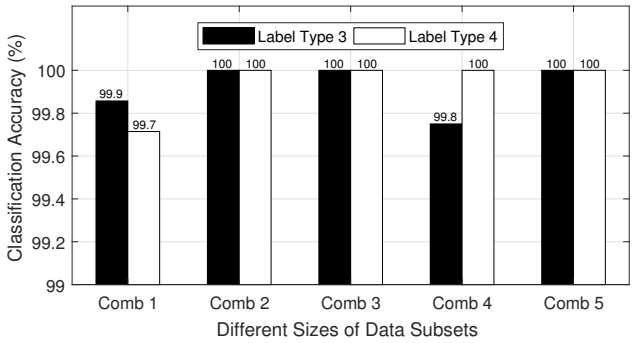


Figure 5.7 K-NN classification results for failure case 3 with different size of data subsets and multiple label Types 3 & 4.

different failure types/locations (see Fig. 5.2) are considered, based on which the performance of the algorithm is evaluated with varying k. The k-NN classification results based on multiple label types 3 and 4 are shown in Fig. 5.6. As can be seen, by increasing the number of neighbors (k), accuracy decreases. We choose $k = 2$ for subsequent analyses, and we will later check that this is the optimal number of neighbors in detection of failure location. Furthermore, the optimal distance is found to be “Cosine", which results in better accuracy compared to other distance functions. Classification results with different size of data subsets are presented in Fig. 5.7 for failure case 3, from which it can be observed that the highest accuracy is achieved based on combinations 2, 3 and 6, so we choose Comb 2, i.e., training & validation 70% , and test 30% for all further results.

The performance of the proposed ML framework employing k-NN with multiple labels is assessed for all failure cases (i.e., cases 1 to 6), where the classification accuracy on test data is reported for each case (see Fig. 5.8). Clearly, the performance of the framework is acceptable

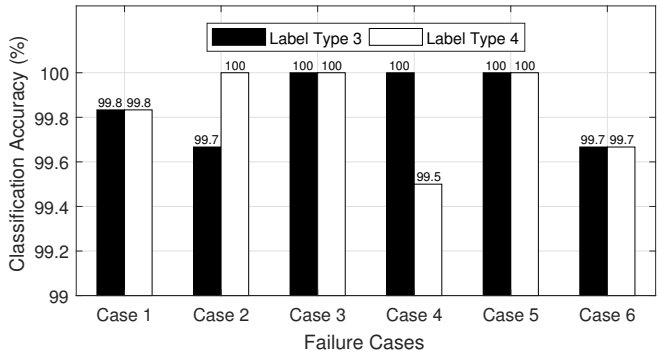


Figure 5.8 K-NN classification results for different failure cases based on label Types 3 & 4.

Figure 5.9 Confusion matrix on test data with k-NN: (a) Case 5 and multiple label Type 3, (b) Case 1 and multiple label Type 4.

such that the highest accuracy based on multiple label types 3 and 4 is 100% for failure cases 5 and 3, respectively. To better visualize the classification results, a confusion matrix containing information about actual and predicted classification is determined. Each column of the confusion matrix represents the patterns in a predicted class, whereas each row denotes the patterns in an actual class. The confusion matrix containing detailed classification results for cases 5 and 1 are depicted in Fig. 5.9. As can be seen, the k-NN method performs well on all classes, including class 2 denoting onset of failure, which is of primary interest for early detection of failure in real-world applications.

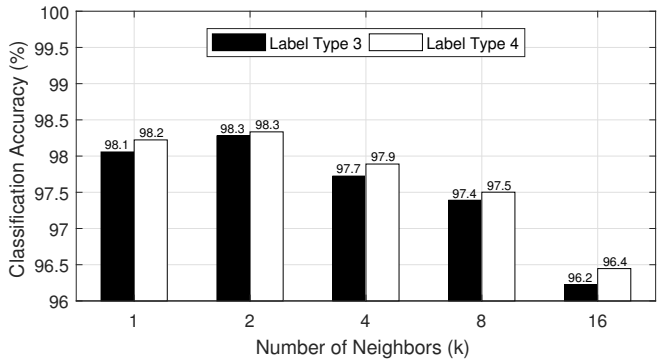


Figure 5.10 K-NN classification accuracy with different number of k based on multiple label Types 3, and 4.

5.5.1.2 Detection of the Location/Pattern of Failure

An attempt is made to detect the location of failure, enabling the framework to predict the pattern of failure. In regard to this, three different failure cases 1 to 3, representing different failure types, are considered for the analysis. Accordingly, nine classes/labels are defined, as shown in Table 5.2, using multiple label type 4 (multiple labels based on damage threshold concept). We also study the effect of different number of neighbors k in this context. We present the accuracy results in Fig. 5.10, in which we observe that $k = 2$ is indeed the optimal number of neighbors, corroborating the choice made in the previous section.

The confusion matrix showing the classification results is presented in Fig. 5.11. As can be observed, the total accuracy is reported as 98.3% using multiple labels/classes shown in Table 5.2 (k-NN detects the location of failure with 98.3% accuracy). Results indicate the overall efficient performance of k-NN to detect the onset of failure (classes 2, 5 and 8) and failure (classes 3, 6 and 9). The incorrect classifications are more concentrated in classes denoting no failure (classes 1, 4 and 7), where the method incorrectly identifies the location of failure in a few data points, because in early stages of the simulation the damage field is similar among the different cases. This is not an issue, since the critical part is the onset of failure. Moreover, the lowest classification accuracy is 86.5% for class 7 (see Fig. 5.11), which is an acceptable performance for a classification algorithm.

Table 5.2 Illustration of label/class definition for detection of location/pattern of failure.

Label/Class	Failure Case	Label based on Label Type 4	New Label/ Class
Label of a Pattern at a Given Time Step	1	0	1
	1	0.5	2
	1	1	3
	2	0	4
	2	0.5	5
	2	1	6
	3	0	7
	3	0.5	8
	3	1	9

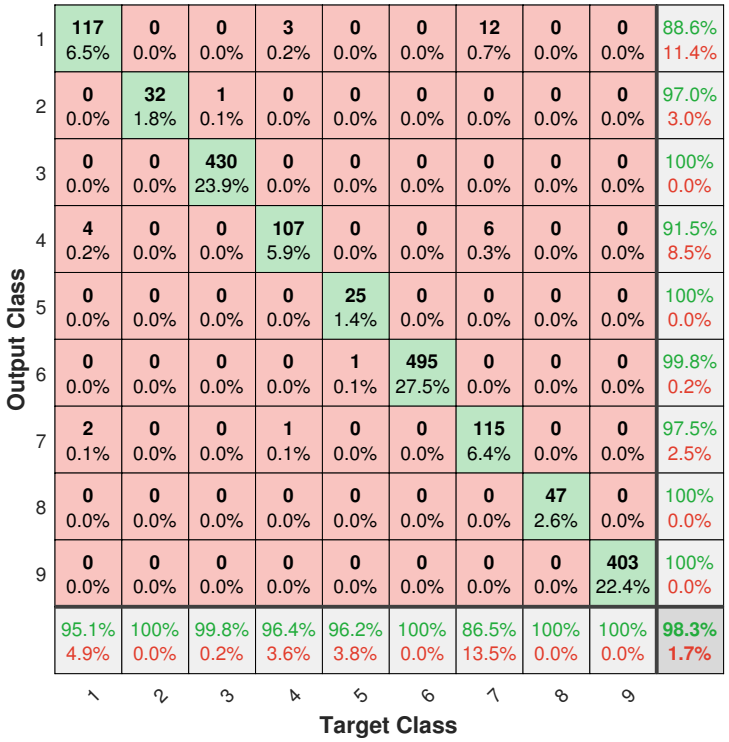


Figure 5.11 Confusion matrix with k-NN classification results for detection of location/pattern of failure based on multiple label Type 4.

5.5.2 Results with ANN

5.5.2.1 Detection of the Presence of Failure

The performance of the proposed ML framework employing ANN algorithm is evaluated in terms of detecting the presence of failure. On this basis, a two-layer (i.e., one hidden-layer with 5 neurons

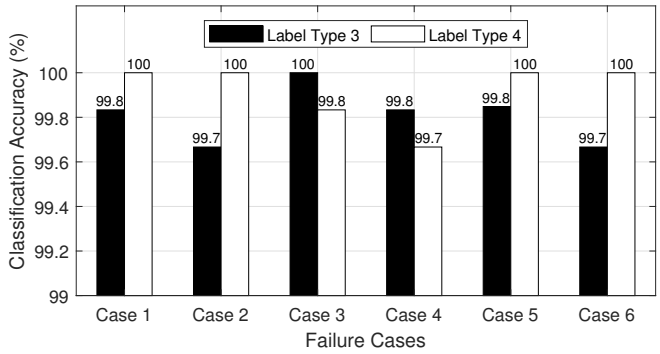


Figure 5.12 ANN classification results for different failure cases based on multiple label Types 3 & 4.

Figure 5.13 Confusion matrix on test data with ANN: (a) Case 5 and multiple label Type 3, (b) Case 1 and multiple label Type 4.

and output layer) feed-forward neural network with Sigmoid classifier/activation function is used. Classification results based on multiple label types 3 and 4 are presented in Fig. 5.12, from which it can be seen that ANN leads to comparable accuracy compared to k-NN (see Fig. 5.8). Detailed classification results with ANN is presented in a confusion matrix shown in Fig. 5.13. Results indicate that ANN effectively detects the presence of failure with total accuracy of 99.90% and 100% using multiple label types 3 and 4, respectively.

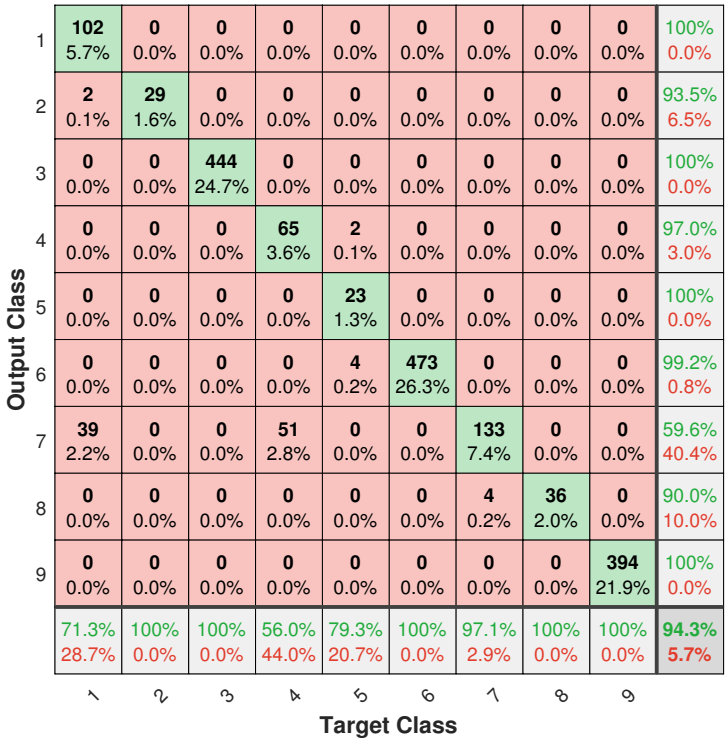


Figure 5.14 Confusion matrix with ANN classification results for detection of location/pattern of failure based on multiple label Type 4.

5.5.2.2 Detection of the Location/Pattern of Failure

Once the presence of failure is detected, ANN algorithm is employed to identify the location/pattern of failure. On this basis, multiple labels defined in Table 5.2 are used for supervised classification with ANN algorithm. A two-layer (i.e., one hidden-layer with 5 neurons and output layer) feed-forward neural network with Sigmoid activation function is used as the ANN architecture to detect the location of failure. The confusion matrix with detailed classification results is presented in Fig. 5.14, from which it can observed that the location/pattern of failure can be successfully detected using ANN with high accuracy. Similarly to k-NN, the majority of misclassifications in ANN belong to classes representing no failure (classes 1 and 4 in Fig. 5.14), due to the similarity of damage field prior to damage localization and crack initiation. In the other classes, ANN still performs successfully, with minimum accuracy of 79.3% when detecting the onset of failure (class 5).

5.5.3 Discussion of Deterministic Results

From observation of classification results employing k-NN and ANN to predict failure in damage phase-field models, we see consistent results with high accuracy. This is specially the case of k-NN, which shows less uncertainty in classifying the patterns. Yet, we must highlight that the nature of data used in this work plays a major role in such satisfactory results. Several nuances and complications of classification algorithms, such as variance-bias trade-off, sensitivity, and distribution/number of data points, are either justified, or less critical, due to characteristics of phase-field models.

In essence, phase-field models are smooth representations of sharp, discontinuous interfaces. In the case of damage modeling, cracks are smoothened in the phase field as an intrinsic approximation of the model. Therefore, as opposed to real data measurements, phase-field solutions evaluated at virtual sensing nodes are naturally smooth, introduce low noise, and consequently lead to less variance in the predictions.

In this regard, the choice of number and position of virtual sensing nodes was performed by an *a priori* study of failure patterns from Fig. 5.2. Different configurations of training data led to little change of classification results. The distribution of sensors considered expert opinion, as to provide the framework with data points associated with different modes of failure. Such assessment considers that placement of sensors in regions around the bases of the specimen will not contribute to accuracy, since there is no damage accumulation in such areas. We remark that for a specific failure case, with model parameters appropriately inferred from experiments, fewer sensor nodes would be required, placed preferably in critical areas (e.g. around the corners, and in the central region).

With respect to the specific algorithms used in this work, the nature of phase-field data aids in the trade-off between bias and variance. Both k-NN and ANN have been extensively studied in the literature, and the bias-variance behavior is well understood. For k-NN, it has been demonstrated analytically that bias increases, and variance decreases with increasing number of neighbors [217, 218]. Since the smoothness of phase-field solutions does not introduce substantial variance, the

choice of $k = 2$ makes the trade-off by reducing the bias. In the case of ANN, we reduce bias by increasing the number of hidden units [217], so the number of neurons used is also justified.

Another source of potential errors is related to the number of time-steps, and their distribution among the classes. This issue shows specifically when predicting onset of failure, the most critical part of the failure process. Classification errors in this case may be due to fewer data points available, since the onset of failure occurs rapidly. A possible remedy would be to refine the time-integration steps around the onset of failure. Additionally, we have shown that the algorithm incorrectly classifies cases of no failure between different fracture patterns due to enhanced smoothness early on the simulation. However, this issue is less catastrophic when compared to erroneously detecting the onset of failure, regardless of the fracture pattern.

Overall, the framework worked effectively with deterministic data, due to the nature of phase-field models, and reasonable choice of k-NN and ANN parameters to compromise the relation between bias and variance. Since phase-field models present smooth features with low variance, we further assess the robustness of the framework by adding noise to the damage field solutions.

5.5.4 Uncertainty Quantification

The results presented in previous sections consisted in smooth, deterministic input data in a single run of the ML algorithms. In this section we propagate the uncertainty associated to data sampling and randomness related to the algorithms, and we assess the robustness and accuracy of both methods to variability in data through the addition of Gaussian noise. This approach aims to verify the effectiveness of the framework to handle real-world data.

5.5.4.1 Algorithmic randomness

We first study the propagation of uncertainties related to the algorithms, still using deterministic data. Such randomness appears in both $k-$NN and ANN due to the random division of time-series data into training, validation and test sets. We need to choose random division to avoid bias, specially in damage data that shows pronounced temporal evolution trends. Furthermore, ANN

Table 5.3 Total classification accuracy mean and standard deviation from algorithmic randomness (%).

	k-NN		ANN	
	Mean	Std. Dev	Mean	Std. Dev
Failure presence - Case 1	99.86	0.16	99.82	1.53
Failure presence - Case 2	99.86	0.16	99.88	0.14
Failure presence - Case 3	99.87	0.14	99.84	0.27
Failure presence - Case 4	99.86	0.16	99.82	0.31
Failure presence - Case 5	99.87	0.14	99.89	0.15
Failure presence - Case 6	99.86	0.15	99.88	0.16
Failure location - Cases 1/2/3	98.28	0.31	84.58	6.00

also presents another source of uncertainty, related to random initialization of weights and biases in each neuron.

We use Monte Carlo (MC) method to run multiple classification problems for each algorithm, and compute the expected total classification accuracy and standard deviation. We use multiple label Type 3, and run 1000 simulations for detection of failure presence in each case (cases 1 to 6), and run additional 1000 classifications for the detection of failure location, using classes from Table 5.2. We show the results in Table 5.3 . We observe that k-NN performs better than ANN in this setting, since ANN incorporates another level of uncertainty from the random guesses of neuron parameters. The randomness of data division does not affect the performance of neither method in the failure location problem. For failure location, ANN is less accurate, but still within acceptable range.

5.5.4.2 Noisy data

To assess the performance of the proposed ML framework with noisy data, time-series data are corrupted by adding Gaussian distributed noise with different standard deviations to damage φ data. We run 1000 MC simulations for k-NN and ANN algorithms using multiple label Type 3, and compute the total accuracy expectation and standard deviation. We propagate uncertainty for failure presence detection in case 3, and for detection of failure location, and show the results in

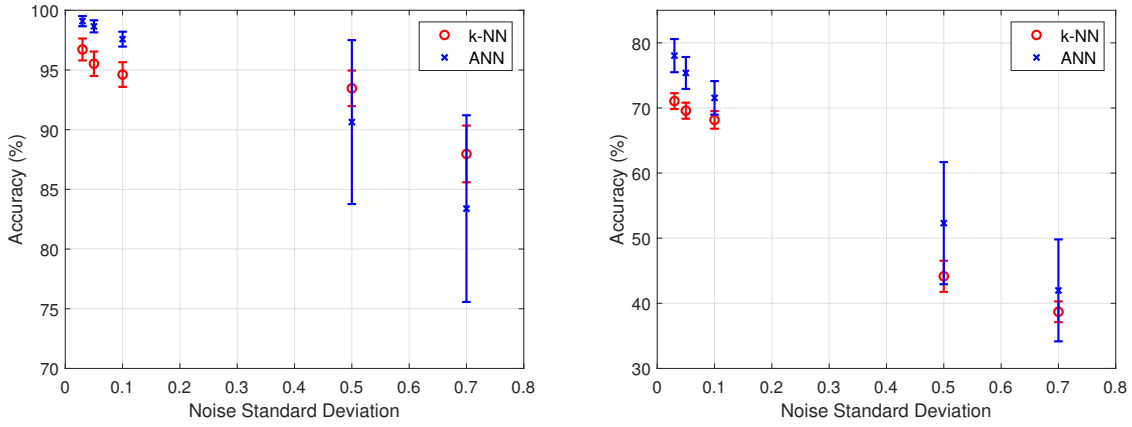


Figure 5.15 Mean total classification accuracy and standard deviation for (a) Detection of failure presence, case 3 and , (b) Detection of failure location.

Fig. 5.15.

We observe that increasing noise levels decreases the mean accuracy, while enlarging the uncertainty range, for both methods. We see that even small intensity noise drops the accuracy to lower levels when compared to mean values based only on algorithmic randomness. In other words, input data dominates over algorithms' uncertainty. For failure presence detection in case 3, we see that ANN performs better than k-NN for low noise levels, while k-NN is more robust under higher noise magnitudes, resulting in higher mean accuracy and lower standard deviation. For case 3, the lowest accuracy was still above 75%, showing robustness with 3 classes. Similarly, when we look at detection of failure location, ANN is superior k-NN for low noise levels, and shows higher mean accuracy even for high noise intensity, yet with more uncertainty. However, we cannot claim good performance of the algorithms with 9 classes under high noise levels, under this specific choice of algorithmic setup. This motivates a more systematic approach to uncertainty and sensitivity of ML algorithms under noisy phase-field data, and it shall be the focus of future studies.

CHAPTER 6

SUMMARY AND FUTURE WORKS

The main objective of this work was to develop a multi-scale, data-driven framework for the modeling, analysis, and simulation of material failure. The framework considers the stochastic processes occurring in the small scales as sources of uncertainty that must be propagated and quantified at the component level. We developed a methodology to obtain fast estimates of uncertainty in dislocation mobility at the nano-scale through a surrogate model that learns a stochastic process from high-fidelity molecular dynamics simulations. At the meso-scale, we studied the collective dislocation dynamics as a probabilistic model governed by nonlocal equations. We developed a machine learning framework to learn the parameters of the nonlocal kernel using discrete dislocation dynamics simulation data, confirming through the resulting operator that failure processes have anomalous character. We confirmed the need for alternative operators through the inspection of uncertainties from stochastic damage and fatigue phase-field models at the continuum level, where parameters associated with free-energy density show more sensitivity in the solution. Then, we developed a Machine Learning framework at the macro-scale that predicts failure even under the presence of noise.

6.1 Concluding Remarks

In **Chapter 2**, we developed a data-driven framework for constructing a surrogate model of dislocation glide. Atomistic simulations of dislocation motion provide the statistics that inform the underlying stochastic process of the surrogate. This is achieved firstly through the coarse-graining of MD domain using a graph-theoretical representation. Over this network, the dislocation is idealized as a random walker jumping between the nodes, where the waiting time distribution is parameterized directly from time-series data obtained in MD simulation. The random walk over the network is simulated through a KMC algorithm based on the waiting times obtained empirically. By tracking the dislocation position over we computed the dislocation velocity for each applied

shear stress, which in turn leads to the estimation of dislocation mobility.

We highlight the following observations from the model and its numerical results:

- The construction followed the assumption of a memoryless, Markovian process governing the dislocation motion, which was a sufficient description based on an estimate of average waiting times from empirical data.
- The estimation of rate constants, often a major difficulty in the application of KMC, was performed directly through MD data. We compared three different methods that yielded nearly identical results.
- From the computed rates, the actual simulation of the stochastic process resulted in dislocation motion in agreement with trajectories simulated by MD. Next, computation of mobility through the surrogate also had excellent agreement with original atomistic estimates.
- Simulation through the surrogate achieved remarkable speedup compared with MD computation times.
- Uncertainty levels dependent on the number of data points used to construct the surrogate. We provided uncertainty estimates for the mobility through the surrogate, taking into account the variance of the underlying stochastic process.

The proposed framework establishes a meaningful bridge for coupling scales, where not only the value of mobility is provided, but its associated uncertainty. Through the description of dislocation motion as a stochastic process informed by high-fidelity data, we can propagate the uncertainty associated with mobility estimations or any other quantity of interest, even with a limited number of MD samples. As a consequence, this framework acts as a tool for more predictive multi-scale material characterization.

In **Chapter 3**, we proposed a data-driven nonlocal model for the simulation of dislocation position probability densities. We generated dislocation shifted position data in the form of particle trajectories from high-fidelity two-dimensional DDD simulations under creep with different load

levels, with and without multiplication mechanisms. From the Lagrangian particle trajectories we estimated the evolution of PDFs through and Adaptive Kernel Density Estimation method. Last, we developed a bi-level ML algorithm to obtain the kernel parameterization for the proposed nonlocal operator that describes that PDF's evolution. The developed approach integrates the high-fidelity dynamics of dislocations at the meso-scale with a continuum probabilistic frame in a fluid-limit sense.

We make the following observations from the integrated framework:

- We recovered the dislocation velocity statistics available in the literature from our two-dimensional DDD simulations. We identified the same exponent of around 2.4 for the power-law decay velocity distribution tail when dislocation multiplication is present, and a sharper, close to 3 exponent without the effect of multiplication mechanisms. The statistics from DDD show similarities among the studied cases.

- The PDF estimation from dislocation trajectories makes evident that the presence of multiplication sources greatly impacts the probability distributions, increasing the heaviness of the tails and implying greater nonlocality.

- Our bi-level algorithm based on a Least-Squares approach for the computation of the optimal nonlocal diffusion coefficient for every pair of α and δ performed well with a manufactured solution, and proved to be robust for data-driven PDFs, considering different train-test splits and initial guess combinations.

- The large horizon parameter δ found among the cases confirms the nonlocal nature of dislocation dynamics, even from a probabilistic perspective. Furthermore, the nonlocal kernel power-law exponent obtained matches the tail decay from dislocation velocity distributions computed in DDD simulations. This establishes a well-defined path between the anomalous behavior observed in particle meso-scale dynamics and the upscaling of anomalous effects to a continuum, macro-scale frame of reference.

Although we used single-glide mechanisms, we note that shifted particle positions may be obtained in more general, multi-slip systems. Since the goal of this framework is to obtain the probability densities in the fluid-limit, the same procedure could be applied to each slip-system in a complex crystal. We further point to the fact that the bulk dynamics adopted here can also be extended to dislocation motion near free surfaces, crack tips, or grain boundaries. In such cases, one could expect the PDFs to show non-zero skewness, which could be naturally accommodated by a different choice of (nonsymmetric) kernel in the nonlocal operator, suitable for skewed, possibly one-sided distributions.

The nonlocal model of dislocation motion at the meso-scale proposed in this work opens up the opportunity of fast computations of quantities of interest compared to the high-fidelity simulations. The implications of nonlocal dislocation models are readily applicable to the study of visco-elasticity and visco-plasticity, where fractional-order models have been successfully applied to model the power-law relaxation including damage effects [47]. One of the main connections to be established is the ultimate effect of different regimes of dislocation dynamics on the evolution of macro-scale free-energy potentials during failure, in phase-field models [104] for instance. Around crack tips and other dislocation generation objects, such as holes, pores, or other micro-cracks, we expect the macro-scale behavior to also be anomalous. Substantially, the methods proposed here can be essential tools to connect other physical processes from a wider range of applications to the generation of corresponding nonlocal operators.

Finally, the proposed bi-level optimization approach is an effective way of reducing the computational burden of optimizing in a high-dimensional parameter space and proves to be robust with both manufactured and simulated datasets.

In **Chapter 4**, we developed an uncertainty quantification and sensitivity analysis framework for stochastic damage and fatigue phase-field equations. We used Monte Carlo sampling and Probabilistic Collocation to compute expectation and standard deviation of damage field, and expected local sensitivity. To compute the local sensitivity at each collocation point, complex-step differentiation was used. Probabilistic Collocation method poses a great advantage over

random sampling methods such as MC, reducing significant computational costs with a simple implementation.

We presented two representative examples to study the uncertainty propagation in the model. We detected two different behaviors of the model based on geometry:

- In the single-edge notched tensile test case, where we already know the crack location and direction, the uncertainty is reduced to the speed of crack propagation. Interestingly, that is not only controlled by the rate of change of damage parameter c, but also indirectly by the Griffith energy g_c. Uncertainty can be inferred by local sensitivity analysis, which shows the same order of parameter influence. When we compute the global sensitivity indices, uncertainty around the crack tip is also controlled by g_c and c;

- In a geometry with no unique crack initiation location nor a determined crack path, such as the tensile test specimen, fatigue coefficient a, and most importantly g_c and γ are the most sensitive parameters. High uncertainty is dominated by influence of g_c and γ, which help determine speed and mostly direction of damage transport, due to the lack of unique and well-known crack path.

The framework has shown that in undefined crack path or location, uncertainty is concentrated around parameters involved with local interactions. Specifically, γ and g_c are multiplying the local interaction term in the free-energy potential, and affect the equation of motion, the Laplacian in damage evolution and are also related to fatigue potentials, which are chosen arbitrarily. The higher sensitivity and uncertainty of those parameters related to local terms motivate the use of different operators that include nonlocal interactions as a way to mitigate model form uncertainty.

In **Chapter 5**, we presented a phase-field based machine learning (ML) framework developed to predict failure of brittle materials. Time-series data are generated according to nodal damage results from finite element simulations of a tensile test specimen. We assessed the performance of the proposed ML framework employing PR scheme and ML algorithms (k-NN and ANN) for

different failure types, and with multiple labels generated based on load-displacement curve and damage threshold concept. We draw the following conclusions from the carried out study:

- Results indicate the acceptable performance of the proposed framework with multiple labels, in which a PR scheme is effectively used to represent time-series data of degradation function $g(\varphi) = (1 - \varphi)^2$ as a pattern. This choice of time-series data is effective since it directly complies with the material softening behavior.

- Both k-NN and ANN were efficient to predict the presence and location of failure. The majority of errors in detection of failure location were concentrated in classes representing no failure, due to smoothness and similarity of damage field early in the simulations.

- Uncertainty related to input data noise dominates over algorithmic randomness uncertainty. The framework showed robustness to noise when detecting failure presence, and showed acceptable accuracy with low noise levels when predicting failure location. In general, with noisy data ANN outperforms k-NN.

Results of this study demonstrate the satisfactory performance of the developed algorithmic framework and the applicability of ML for failure prediction with damage phase-field time-series data.

6.2 Future Works

Following the research directions initiated in this work, we believe that the proposed framework has potential applications in multi-scale failure analysis in different disciplines under the perspective of uncertainty propagation and stochastic modeling. A potential direction is to use the probabilistic model of meso-scale dislocation dynamics to learn macro-scale free-energy potentials associated to failure processes. A natural consequence of learning free-energy functions from the nonlocal dynamics is the development of variable-order nonlocal methods, where the nonlocality only manifests in critical regions with presence of failure precursors, such as dislocation multiplication. This approach has the potential of mitigating the model-form uncertainty observed in the global

sensitivity of damage models. Finally, the propagation of uncertainty across the scales has the potential of solving practical engineering problems where the knowledge of lower-scale statistics provide more predictable failure models. The combination of multi-scale uncertainty propagation, with machine learning algorithms for failure detection and reliability analysis using real-time data could enhance the accuracy of failure predictions in diverse applications.

APPENDIX

APPENDIX

DISCRETIZATION OF THE 2-D PHASE-FIELD MODEL OF DAMAGE AND FATIGUE

A.1 Spatial Discretization

We approximate a deterministic solution of the damage and fatigue phase-field model over its spatial domain Ω^d with finite element method, where the semi-discrete form of Equations (4.2) is obtained from the weak Galerkin form after multiplication by test functions and integration over the domain. A more detailed derivation of the spatial discretization in 2D can be found in Chiarelli et al (2017) [162]. Denoting $\ddot{\hat{\mathbf{u}}} = \dot{\hat{\mathbf{v}}}$ we write the semi-discrete form for an element k as

$$\begin{cases} \mathbf{M}^k \, \ddot{\hat{\mathbf{u}}}^k = \mathbf{K}_u^k \, \hat{\mathbf{u}}^k + \mathbf{K}_v^k \, \hat{\mathbf{v}}^k + \mathbf{w}_a^k + \mathbf{M}^k \, \hat{\mathbf{f}}^k, \\ \mathbf{M}_\varphi^k \, \dot{\hat{\varphi}}^k = \left(\mathbf{P}_\varphi^k + \mathbf{K}_c^k \right) \hat{\varphi}^k + \mathbf{w}_b^k + \mathbf{w}_c^k, \\ \mathbf{M}_\mathcal{F}^k \, \dot{\hat{\mathcal{F}}}^k = \mathbf{w}_d^k, \end{cases} \tag{A.1}$$

where $\mathbf{M}$, $\mathbf{M}_\varphi$ and $\mathbf{M}_\mathcal{F}$ are mass matrices related to displacement, damage and fatigue. In the equation of motion, $\mathbf{K}_u$ is the elasticity stiffness matrix degraded by damage, $\mathbf{K}_v$ is associated to viscous damping and $\mathbf{w}_a$ is a term related to gradient of damage that affects the displacement field. Term $\mathbf{P}_\varphi$ in the damage evolution equation includes the Laplacian and potential $\mathcal{H}'(\varphi)$. The influence of displacement in damage is represented by $\mathbf{K}_c$ and $\mathbf{w}_b$. Effect of potential $\mathcal{H}'_f(\varphi)$ is considered in term $\mathbf{w}_c$. $\mathbf{w}_d$ is the operator on the right-hand side of fatigue evolution equation.

We apply the standard assembly operation to obtain the global form of the operator matrices and we will drop the superscript k in the global sense.

From the semi-discrete system of equations (A.1), the solution at each element is written as a linear combination of local nodal basis functions such that

$$\mathbf{u}^k = \mathbf{N}^k \, \hat{\mathbf{u}}^k, \tag{A.2}$$

$$\mathbf{v}^k = \mathbf{N}^k \, \hat{\mathbf{v}}^k, \tag{A.3}$$

$$\varphi^k = \mathbf{N}_\varphi^k \, \hat{\varphi}^k, \tag{A.4}$$

$$\mathcal{F}^k = \mathbf{N}_\mathcal{F}^k \, \hat{\mathcal{F}}^k, \tag{A.5}$$

Constructing a mesh of linear triangles, the nodal solutions are defined as:

$$\hat{\mathbf{u}}^k = \begin{bmatrix} u_{1x}^k & u_{1y}^k & u_{2x}^k & u_{2y}^k & u_{3x}^k & u_{3y}^k \end{bmatrix}^T, \tag{A.6}$$

$$\hat{\mathbf{v}}^k = \begin{bmatrix} v_{1x}^k & v_{1y}^k & v_{2x}^k & v_{2y}^k & v_{3x}^k & v_{3y}^k \end{bmatrix}^T, \tag{A.7}$$

$$\hat{\varphi}^k = \begin{bmatrix} \varphi_1^k & \varphi_2^k & \varphi_3^k \end{bmatrix}^T, \tag{A.8}$$

$$\hat{\mathcal{F}}^k = \begin{bmatrix} \mathcal{F}_1^k & \mathcal{F}_2^k & \mathcal{F}_3^k \end{bmatrix}^T \tag{A.9}$$

$$\tag{A.10}$$

with interpolation matrices

$$\mathbf{N}^k = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}, \tag{A.11}$$

$$\mathbf{N}_\varphi^k = \mathbf{N}_\mathcal{F}^k = \begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix}, \tag{A.12}$$

$$\tag{A.13}$$

Gradients of displacement, velocity and damage are approximated by linear combinations of shape function derivatives

$$\mathbf{E}^k = \mathbf{B}_u^k \, \hat{\mathbf{u}}^k, \tag{A.14}$$

$$\mathbf{D}^k = \mathbf{B}_v^k \, \hat{\mathbf{v}}^k, \tag{A.15}$$

$$\nabla\varphi^k = \mathbf{B}_\varphi^k \, \hat{\varphi}^k, \tag{A.16}$$

Derivative matrices are defined as:

$$\mathbf{B}_u^k = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} & N_{3,y} & N_{3,x} \end{bmatrix}, \tag{A.17}$$

$$\mathbf{B}_v^k = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} \\ \frac{1}{\sqrt{2}}N_{1,y} & \frac{1}{\sqrt{2}}N_{1,x} & \frac{1}{\sqrt{2}}N_{2,y} & \frac{1}{\sqrt{2}}N_{2,x} & \frac{1}{\sqrt{2}}N_{3,y} & \frac{1}{\sqrt{2}}N_{3,x} \end{bmatrix}, \tag{A.18}$$

$$\mathbf{B}_\varphi^k = \begin{bmatrix} N_{1,x} & N_{2,x} & N_{3,x} \\ N_{1,y} & N_{2,y} & N_{3,y} \end{bmatrix}. \tag{A.19}$$

From those definitions, we can express the mass, stiffness and remaining operator matrices from Equation (A.1) as

$$\mathbf{M}^k = \int_{\Omega^k} \mathbf{N}^T \mathbf{N} d\Omega^k \, ; \tag{A.20}$$

$$\mathbf{M}_\varphi^k = \int_{\Omega^k} \mathbf{N}_\varphi^T \mathbf{N}_\varphi \, d\Omega^k \, ; \tag{A.21}$$

$$\mathbf{M}_\mathcal{F}^k = \int_{\Omega^k} \mathbf{N}_\mathcal{F}^T \mathbf{N}_\mathcal{F} \, d\Omega^k \, ; \tag{A.22}$$

$$\mathbf{K}_u^k = -\int_{\Omega^k} \frac{1}{\rho_0} \left(1 - \mathbf{N}_\varphi^k \, \hat{\varphi}^k\right)^2 \left(\mathbf{B}_u^k\right)^T C \mathbf{B}_u^k \, d\Omega^k \, ; \tag{A.23}$$

$$\mathbf{K}_v^k = -\int_{\Omega^k} \frac{b}{\rho_0} \left(\mathbf{B}_v^k\right)^T \mathbf{B}_v^k \, d\Omega^k \, ; \tag{A.24}$$

$$\mathbf{P}_\varphi^k = -\int_{\Omega^k} \frac{\gamma g_c}{\lambda} \left(\mathbf{B}_\varphi^k\right)^T \mathbf{B}_\varphi^k \, d\Omega^k - \int_{\Omega^k} \frac{g_c}{\lambda\gamma} \left(\mathbf{N}_\varphi^k\right)^T \mathbf{N}_\varphi^k \, d\Omega^k \, ; \tag{A.25}$$

$$\mathbf{K}_c^k = -\int_{\Omega^k} \frac{1}{\lambda} \left(\mathbf{B}_u^k \, \hat{\mathbf{u}}^k\right)^T C \left(\mathbf{B}_u^k \, \hat{\mathbf{u}}^k\right) \left(\mathbf{N}_\varphi^k\right)^T \mathbf{N}_\varphi^k \, d\Omega^k \, ; \tag{A.26}$$

$$\mathbf{w}_a^k = \int_{\Omega^k} \frac{\gamma g_c}{\rho_0} \left(\mathbf{B}_\varphi^k \, \hat{\varphi}^k \otimes \mathbf{B}_\varphi^k \, \hat{\varphi}^k\right) : \mathbf{B}_u^k \, d\Omega^k \, ; \tag{A.27}$$

$$\mathbf{w}_b^k = \int_{\Omega^k} \frac{1}{\lambda} \left(\mathbf{B}_u^k \, \hat{\mathbf{u}}^k\right)^T C \left(\mathbf{B}_u^k \, \hat{\mathbf{u}}^k\right) \mathbf{N}_\varphi^k \, d\Omega^k \, ; \tag{A.28}$$

$$\mathbf{w}_c^k = \int_{\Omega^k} \frac{-1}{\lambda\gamma} \left(\mathbf{N}_\mathcal{F}^k\right)^T \mathbf{N}_\mathcal{F}^k \, \hat{\mathcal{F}}^k \, d\Omega^k \, ; \tag{A.29}$$

$$\begin{aligned} \mathbf{w}_d^k = \int_{\Omega^k} & \frac{a}{\gamma} \left(1 - \mathbf{N}_\varphi^k \, \hat{\varphi}^k\right) \\ & \left(\mathbf{N}_\varphi^k \, \hat{\varphi}^k\right) \left| \left[C \left(\mathbf{B}_u^k \, \hat{\mathbf{u}}^k\right) + b \left(\mathbf{B}_v^k \, \hat{\mathbf{v}}^k\right)\right] : \left(\mathbf{B}_v^k \, \hat{\mathbf{v}}^k\right) \right| d\Omega^k \, . \end{aligned} \tag{A.30}$$

We consider the plane stress elasticity matrix C given by

$$C = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{A.31}$$

We express the second-order tensor in Equation A.27 as a vector using Voigt notation.

A.2 Time Discretization

We adopt a semi-implicit time integration scheme, where we solve each equation separately using a suitable implicit method, treating nonlinear terms and other variable fields explicitly. The methodology is based on the work by Haveroth et al (2018) [163], where a detailed derivation can be found. We split the solution time interval $[0, T]$ in discrete time steps t_n with time increments given by $\Delta t = t_{n+1} - t_n > 0$, $n = 0, 1, \ldots$. We denote the global approximations for the variables at t_{n+1} as

$$\mathbf{u}_{n+1} = \hat{\mathbf{u}}(t_{n+1}), \tag{A.32}$$

$$\mathbf{v}_{n+1} = \hat{\dot{\mathbf{u}}}(t_{n+1}), \tag{A.33}$$

$$\varphi_{n+1} = \hat{\varphi}(t_{n+1}), \tag{A.34}$$

$$\mathcal{F}_{n+1} = \hat{\mathcal{F}}(t_{n+1}). \tag{A.35}$$

We first discuss damage time integration. We use a backward Euler scheme to compute φ_{n+1} from Equation (A.1). Parameter λ, displacement and fatigue are treated explicitly using values from time step t_n. This simplifies the solution and avoids the use of iterative methods to treat the nonlinearity. Evolution of damage is then obtained by solving the linear system

$$\left[\mathbf{M}_\varphi - \Delta t\left(\mathbf{P}_\varphi + \mathbf{K}_c\right)\right]\varphi_{n+1} = \mathbf{M}_\varphi \varphi_n + \Delta t(\mathbf{w}_b + \mathbf{w}_c). \tag{A.36}$$

With the updated damage field, we use Newmark method to solve displacement and velocity in the equation of motion. In Newmark scheme, acceleration and velocity at time t_{n+1} are approximated by

$$\ddot{\mathbf{u}}_{n+1} = \alpha_1 \left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) - \alpha_2 \dot{\mathbf{u}}_n - \alpha_3 \ddot{\mathbf{u}}_n \tag{A.37}$$

$$\dot{\mathbf{u}}_{n+1} = \alpha_4 \left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) + \alpha_5 \dot{\mathbf{u}}_n + \alpha_6 \ddot{\mathbf{u}}_n, \tag{A.38}$$

Algorithm A.1 Semi-implicit time integration scheme

1: **for** $n = 0 \rightarrow N - 1$ **do**
2: Given $\mathbf{u}_n$, $\mathbf{v}_n$ and λ_n, solve Eq. (A.36) for φ_{n+1}.
3: Solve Eq. (A.45) for $\mathbf{u}_{n+1}$.
4: Update acceleration $\mathbf{a}_{n+1}$ and velocity $\mathbf{v}_{n+1}$ using Eq. (A.37) and (A.38).
5: Update the fatigue $\mathcal{F}_{n+1}$.
6: Update the time step by adding the time increment Δt.
7: **end for**

with α_i, $i = 1, 2, \ldots, 6$ written in terms of standard Newmark coefficients $\tilde{\gamma}$ and $\tilde{\beta}$:

$$\alpha_1 = \frac{1}{\tilde{\beta}\Delta t^2},\ \alpha_2 = \frac{1}{\tilde{\beta}\Delta t},\ \alpha_3 = \frac{1-2\tilde{\beta}}{2\tilde{\beta}},\ \alpha_4 = \frac{\tilde{\gamma}}{\tilde{\beta}\Delta t},\ \alpha_5 = 1 - \frac{\tilde{\gamma}}{\tilde{\beta}}\ \text{and}\ \alpha_6 = \left(1 - \frac{\tilde{\gamma}}{2\tilde{\beta}}\right)\Delta t. \tag{A.39-A.44}$$

The discrete form is then

$$\begin{aligned}\left[\alpha_1\mathbf{M} - \mathbf{K}_u - \alpha_4\mathbf{K}_v\right]\mathbf{u}_{n+1} = \mathbf{M}\left[\alpha_3\ddot{\mathbf{u}}_n + \alpha_2\dot{\mathbf{u}}_n + \alpha_1\mathbf{u}_n\right] \\ + \mathbf{K}_v\left[\alpha_6\ddot{\mathbf{u}}_n + \alpha_5\dot{\mathbf{u}}_n - \alpha_4\mathbf{u}_n\right] + \mathbf{w}_a + \mathbf{M}\mathbf{f}_{n+1}.\end{aligned} \tag{A.45}$$

with coefficients α_i, $i = 1, 2, \ldots, 6$.

After the solution of Equation (A.45) we update the current acceleration and velocity fields using Equations (A.37) and (A.38), respectively. When imposing prescribed displacement $\bar{\mathbf{u}}(t_{n+1})$ we should also prescribe appropriate velocity and acceleration at the boundaries using

$$\bar{\ddot{\mathbf{u}}}_{n+1} = \frac{d^2}{dt^2}\bar{\mathbf{u}}(t_{n+1}) \quad \text{and} \quad \bar{\dot{\mathbf{u}}}_{n+1} = \frac{d}{dt}\bar{\mathbf{u}}(t_{n+1}), \tag{A.46-A.47}$$

where the bar symbol represents the prescribed degrees of freedom.

Finally, we update the fatigue variable using a Trapezoidal method given by

$$\mathcal{F}_{n+1} = \mathcal{F}_n + \frac{\Delta t}{2}\mathbf{M}_{\mathcal{F}}^{-1}\left[\mathbf{w}_d\left(\mathbf{u}_{n+1}, \mathbf{v}_{n+1}, \varphi_{n+1}\right) + \mathbf{w}_d\left(\mathbf{u}_n, \mathbf{v}_n, \varphi_n\right)\right]. \tag{A.48}$$

Algorithm A.1 presents the final semi-implicit time integration scheme.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Derek Hull and David J Bacon. Introduction to dislocations. Butterworth-Heinemann, 2001.

[2] Alan Arnold Griffith. Vi. the phenomena of rupture and flow in solids. Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character, 221(582-593):163–198, 1921.

[3] Daniel Bonamy. Intermittency and roughening in the failure of brittle heterogeneous materials. Journal of Physics D: Applied Physics, 42(21):214014, November 2009.

[4] Benoit B. Mandelbrot, Dann. E. Passoja, and Alvin J. Paullay. Fractal character of fracture surfaces of metals. Nature, 308:721, April 1984.

[5] J. J. Mecholsky, D. E. Passoja, and K. S. Feinberg-Ringel. Quantitative Analysis of Brittle Fracture Surfaces Using Fractal Geometry. Journal of the American Ceramic Society, 72(1):60–65, January 1989.

[6] E Bouchaud, G Lapasset, and J Planès. Fractal Dimension of Fractured Surfaces: A Universal Value? Europhysics Letters (EPL), 13(1):73–79, September 1990.

[7] Q Y Long, Li Suqin, and C W Lung. Studies on the fractal dimension of a fracture surface formed by slow stable crack propagation. Journal of Physics D: Applied Physics, 24(4):602–607, April 1991.

[8] R.E. Williford. Fractal fatigue. Scripta Metallurgica et Materialia, 25(2):455–460, February 1991.

[9] Carpinteri, A., Chiaia, B. Multifractal scaling law for the fracture energy variation of concrete structures. Fracture mechanics of concrete structures, 1995.

[10] Carpinteri, A and Yang, GP. Fractal dimension evolution of microcrack net in disordered materials. Theoretical and applied fracture mechanics, 25(1):73–81, 1996.

[11] M.-Carmen Miguel, Alessandro Vespignani, Stefano Zapperi, Jérôme Weiss, and Jean-Robert Grasso. Intermittent dislocation flow in viscoplastic deformation. Nature, 410(6829):667–671, April 2001.

[12] Marisol Koslowski, Richard LeSar, and Robb Thomson. Avalanches and Scaling in Plastic Deformation. Physical Review Letters, 93(12), September 2004.

[13] Thiebaud Richeton, Jérôme Weiss, and François Louchet. Breakdown of avalanche critical behaviour in polycrystalline plasticity. Nature Materials, 4(6):465–469, June 2005.

[14] D. M. Dimiduk. Scale-Free Intermittent Flow in Crystal Plasticity. Science, 312(5777):1188–1190, May 2006.

[15] Jérôme Weiss, Thiebaud Richeton, François Louchet, Frantisek Chmelik, Patrick Dobron, Denis Entemeyer, Mikhail Lebyodkin, Tatiana Lebedkina, Claude Fressengeas, and Russell J. McDonald. Evidence for universal intermittent crystal plasticity from acoustic emission and high-resolution extensometry experiments. Physical Review B, 76(22), December 2007.

[16] A. Petri, G. Paparo, A. Vespignani, A. Alippi, and M. Costantini. Experimental Evidence for Critical Dynamics in Microfracturing Processes. Physical Review Letters, 73(25):3423–3426, December 1994.

[17] A. Garcimartín, A. Guarino, L. Bellon, and S. Ciliberto. Statistical Properties of Fracture Precursors. Physical Review Letters, 79(17):3202–3205, October 1997.

[18] D. Bonamy, S. Santucci, and L. Ponson. Crackling Dynamics in Material Failure as the Signature of a Self-Organized Dynamic Phase Transition. Physical Review Letters, 101(4):045501, July 2008.

[19] Ashivni Shekhawat, Stefano Zapperi, and James P. Sethna. From Damage Percolation to Crack Nucleation Through Finite Size Criticality. Physical Review Letters, 110(18), April 2013.

[20] Purusattam Ray. Statistical physics perspective of fracture in brittle and quasi-brittle materials. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 377(2136):20170396, January 2019.

[21] SJ Zhou, DM Beazley, PS Lomdahl, and BL Holian. Large-scale molecular dynamics simulations of three-dimensional ductile failure. Physical Review Letters, 78(3):479, 1997.

[22] Vasily Bulatov, Farid F Abraham, Ladislas Kubin, Benoit Devincre, and Sidney Yip. Connecting atomistic and mesoscale simulations of crystal plasticity. Nature, 391(6668):669–672, 1998.

[23] Arttu Lehtinen, Fredric Granberg, Lasse Laurson, Kai Nordlund, and Mikko J Alava. Multiscale modeling of dislocation-precipitate interactions in fe: from molecular dynamics to discrete dislocations. Physical Review E, 93(1):013309, 2016.

[24] Saikumar R Yeratapally, Michael G Glavicic, Christos Argyrakis, and Michael D Sangid. Bayesian uncertainty quantification and propagation for validation of a microstructure sensitive model for prediction of fatigue crack initiation. Reliability Engineering & System Safety, 164:110–123, 2017.

[25] Anh V Tran and Yan Wang. Reliable molecular dynamics: Uncertainty quantification using interval analysis in molecular dynamics simulation. Computational Materials Science,

127:141–160, 2017.

[26] Aleksandr Chernatynskiy, Simon R Phillpot, and Richard LeSar. Uncertainty quantification in multiscale simulation of materials: A prospective. Annual Review of Materials Research, 43:157–182, 2013.

[27] A. Arsenlis, W. Cai, M. Tang, M. Rhee, T. Oppelstrup, G. Hommes, T.G. Pierce, and V.V. Bulatov. Enabling strain hardening simulations with dislocation dynamics. Modeling and Simulation in Materials Science and Engineering, 15(6), 2007.

[28] Hythem Sidky and Jonathan K. Whitmer. Learning free energy landscapes using artificial neural networks. The Journal of Chemical Physics, 148(10):104111, March 2018.

[29] Xiaoxuan Zhang and Krishna Garikipati. Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures. Computer Methods in Applied Mechanics and Engineering, 372:113362, December 2020.

[30] Hengxu Song, Nina Gunkelmann, Giacomo Po, and Stefan Sandfeld. Data-mining of dislocation microstructures: concepts for coarse-graining of internal energies. Modelling and Simulation in Materials Science and Engineering, 29(3):035005, 2021.

[31] Christian Miehe, Martina Hofacker, and Fabian Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. Computer Methods in Applied Mechanics and Engineering, 199(45-48):2765–2778, November 2010.

[32] M. Ambati, T. Gerasimov, and L. De Lorenzis. Phase-field modeling of ductile fracture. Computational Mechanics, 55(5):1017–1040, May 2015.

[33] Michael J. Borden, Clemens V. Verhoosel, Michael A. Scott, Thomas J.R. Hughes, and Chad M. Landis. A phase-field description of dynamic brittle fracture. Computer Methods in Applied Mechanics and Engineering, 217-220:77–95, April 2012.

[34] J.L. Boldrini, E.A. Barros de Moraes, L.R. Chiarelli, F.G. Fumes, and M.L. Bittencourt. A non-isothermal thermodynamically consistent phase field framework for structural damage and fatigue. Computer Methods in Applied Mechanics and Engineering, 312:395–427, December 2016.

[35] Mikko J Alava, Phani KVV Nukala, and Stefano Zapperi. Statistical models of fracture. Advances in Physics, 55(3-4):349–476, 2006.

[36] VV Mourzenko, J-F Thovert, and PM Adler. Percolation of three-dimensional fracture networks with power-law size distribution. Physical Review E, 72(3):036103, 2005.

[37] Qiang Du, Max Gunzburger, Richard B Lehoucq, and Kun Zhou. Analysis and approximation of nonlocal diffusion problems with volume constraints. SIAM review, 54(4):667–696, 2012.

[38] Qiang Du, Max Gunzburger, Richard B Lehoucq, and Kun Zhou. A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. Mathematical Models and Methods in Applied Sciences, 23(03):493–540, 2013.

[39] Marta D'Elia and Max Gunzburger. The fractional laplacian operator on bounded domains as a special case of the nonlocal diffusion operator. Computers & Mathematics with Applications, 66(7):1245–1260, 2013.

[40] S.A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. Journal of the Mechanics and Physics of Solids, 48(1):175–209, January 2000.

[41] John T Foster, Stewart Andrew Silling, and Wayne W Chen. Viscoplasticity using peridynamics. International journal for numerical methods in engineering, 81(10):1242–1258, 2010.

[42] Guanfeng Zhang, Quang Le, Adrian Loghin, Arun Subramaniyan, and Florin Bobaru. Validation of a peridynamic model for fatigue cracking. Engineering Fracture Mechanics, 162:76–94, 2016.

[43] Wenke Hu, Youn Doh Ha, and Florin Bobaru. Peridynamic model for dynamic fracture in unidirectional fiber-reinforced composites. Computer Methods in Applied Mechanics and Engineering, 217:247–261, 2012.

[44] Pranesh Roy, Deepak Behera, and Erdogan Madenci. Peridynamic simulation of finite elastic deformation and rupture in polymers. Engineering Fracture Mechanics, 236:107226, 2020.

[45] Yan Gao and Selda Oterkus. Non-local modeling for fluid flow coupled with heat transfer by using peridynamic differential operator. Engineering Analysis with Boundary Elements, 105:104–121, 2019.

[46] JL Suzuki, M Zayernouri, ML Bittencourt, and GE Karniadakis. Fractional-order uniaxial visco-elasto-plastic models for structural analysis. Computer Methods in Applied Mechanics and Engineering, 308:443–467, 2016.

[47] Jorge Suzuki, Yongtao Zhou, Marta D'Elia, and Mohsen Zayernouri. A thermodynamically consistent fractional visco-elasto-plastic model with memory-dependent damage for anomalous materials. Computer Methods in Applied Mechanics and Engineering, 373:113494, 2021.

[48] Guofei Pang, Marta D'Elia, Michael Parks, and George E Karniadakis. npinns: nonlocal physics-informed neural networks for a parametrized nonlocal universal laplacian operator. algorithms and applications. Journal of Computational Physics, 422:109760, 2020.

[49] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378:686–707, 2019.

[50] Xiao Xu, Marta D'Elia, and John T Foster. A machine-learning framework for peridynamic material models with physical constraints. arXiv preprint arXiv:2101.01095, 2021.

[51] Huaiqian You, Yue Yu, Nathaniel Trask, Mamikon Gulian, and Marta D'Elia. Data-driven learning of nonlocal physics from high-fidelity synthetic data. Computer Methods in Applied Mechanics and Engineering, 374:113553, 2021.

[52] Huaiqian You, Yue Yu, Stewart Silling, and Marta D'Elia. Data-driven learning of nonlocal models: from high-fidelity simulations to constitutive laws. arXiv preprint arXiv:2012.04157, 2020.

[53] Huaiqian You, Yue Yu, Stewart Silling, and Marta D'Elia. A data-driven peridynamic continuum model for upscaling molecular dynamics. arXiv preprint arXiv:2108.04883, 2021.

[54] John W. Cahn and John E. Hilliard. Free Energy of a Nonuniform System. I. Interfacial Free Energy. The Journal of Chemical Physics, 28(2):258–267, February 1958.

[55] Samuel M. Allen and John W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. Acta Metallurgica, 27(6):1085–1095, June 1979.

[56] E. A. B. F. Lima, J. T. Oden, and R. C. Almeida. A hybrid ten-species phase-field model of tumor growth. Mathematical Models and Methods in Applied Sciences, 24(13):2569–2599, December 2014.

[57] Pengtao Yue, James J. Feng, Chun Liu, and Jie Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. Journal of Fluid Mechanics, 515:293–317, 2004.

[58] Pengtao Sun, Jinchao Xu, and Lixiang Zhang. Full Eulerian finite element method of a phase field model for fluid–structure interaction problem. Computers & Fluids, 90:1 – 8, 2014.

[59] Ralph C Smith. Uncertainty quantification: theory, implementation, and applications, volume 12. Siam, 2013.

[60] Zeev Schuss. Singular perturbation methods in stochastic differential equations of mathematical physics. Siam Review, 22(2):119–155, 1980.

[61] Ivo Babuška and Panagiotis Chatzipantelidis. On solving elliptic stochastic partial differential

equations. Computer Methods in Applied Mechanics and Engineering, 191(37-38):4093–4122, 2002.

[62] Roger G Ghanem and Pol D Spanos. Stochastic finite elements: a spectral approach. Courier Corporation, 2003.

[63] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. Computer methods in applied mechanics and engineering, 191(43):4927–4948, 2002.

[64] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. SIAM Journal on Numerical Analysis, 42(2):800–825, 2004.

[65] Ivo Babuška, Raúl Tempone, and Georgios E Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. Computer methods in applied mechanics and engineering, 194(12-16):1251–1294, 2005.

[66] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. SIAM Journal on Scientific Computing, 27(3):1118–1139, 2005.

[67] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Journal on Numerical Analysis, 45(3):1005–1034, 2007.

[68] Sergei Abramovich Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In Doklady Akademii Nauk, volume 148, pages 1042–1045. Russian Academy of Sciences, 1963.

[69] Anindya Ghoshal, Muthuvel Murugan, Michael J. Walock, Luis Bravo, Jeffrey J. Swab, Clara Hofmeister-Mock, Samuel G. Hirsch, Robert J. Dowding, M. Pepi, Andy Nieto, Larry Fehrenbacher, Kelvin Wong, Acree Technologies, Victor Grubsky, Matthew T. Webster, Nishan Jain, Alison B. Flatau, Andrew Wright, and Jian Luo. Advanced high temperature propulsion materials research project: An update. 2019.

[70] Ed Habtour, Daniel P Cole, Jaret C Riddick, Volker Weiss, Mark Robeson, Raman Sridharan, and Abhijit Dasgupta. Detection of fatigue damage precursor using a nonlinear vibration approach. Structural Control and Health Monitoring, 23(12):1442–1463, 2016.

[71] J. Chang, W. Cai, V. Bulatov, and S. Yip. Dislocation motion in bcc metals by molecular dynamics. Materials Science and Engineering: A, 309-310:160–163, 2001.

[72] F. Maresca, D. Dragoni, G. Csányi, N. Marzari, and W.A. Curtin. Screw dislocation structure and mobility in body centered cubic fe predicted by a gaussian approximation potential. npj Computational Materials, 4, 2018.

[73] S.J. Zhou, D.L. Preston, P.S. Lomdahl, and D.M. Beazley. Large-scale molecular dynamics simulations of dislocation intersection in copper. Science, 279(5356):1525–1527, 1998.

[74] B. Chen, Suzhi. Li, H. Zong, X. Ding, J. Sun, and E. Ma. Unusual activated processes controlling dislocation motion in body-centered-cubic high-entropy alloys. Proceedings of the National Academy of Sciences of the United States of America, 117(28):16199–16206, 2020.

[75] S. Queyreau, J. Marian, M.R. Gilbert, and B.D. Wirth. Edge dislocation mobilities in bcc fe obtained by molecular dynamics. Physical Review B, 84:064106, 2011.

[76] A. Lehtinen, L. Laurson, F. Granberg, K. Nordlund, and M.J. Alava. Effects of precipitates and dislocation loops on the yield stress of irradiated iron. Scientific Reports, 8(1):6914, 2018.

[77] V. Bulatov and W. Cai. Computer Simulations of Dislocations. Oxford University Press, 2006.

[78] V. Bulatov, L. Hsiung, M. Tang, A. Arsenlis, M. Bartelt, W. Cai, J. Florando, M. Hiratani, M. Rhee, G. Hommes, T. Pierce, and T. Diaz de la Rubia. Dislocation multi-junctions and strain hardening. Nature, 440:1174–1178, 2006.

[79] Arthur F Voter. Introduction to the kinetic monte carlo method. In Radiation effects in solids, pages 1–23. Springer, 2007.

[80] Tim P Schulze. Efficient kinetic monte carlo simulation. Journal of Computational Physics, 227(4):2455–2462, 2008.

[81] WM Young and EW Elcock. Monte carlo studies of vacancy migration in binary ordered alloys: I. Proceedings of the Physical Society, 89(3):735, 1966.

[82] Alfred B Bortz, Malvin H Kalos, and Joel L Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. Journal of Computational Physics, 17(1):10–18, 1975.

[83] B Meng and WH Weinberg. Dynamical monte carlo studies of molecular beam epitaxial growth models: interfacial scaling and morphology. Surface Science, 364(2):151–163, 1996.

[84] Stephan A Baeurle, Takao Usami, and Andrei A Gusev. A new multiscale modeling approach for the prediction of mechanical properties of polymer-based nanomaterials. Polymer, 47(26):8604–8617, 2006.

[85] Mie Andersen, Chiara Panosetti, and Karsten Reuter. A practical guide to surface kinetic monte carlo simulations. Frontiers in chemistry, 7:202, 2019.

[86] Wei Cai, Vasily V Bulatov, Sidney Yip, and Ali S Argon. Kinetic monte carlo modeling of dislocation motion in bcc metals. Materials Science and Engineering: A, 309:270–273, 2001.

[87] Wei Cai, Vasily V Bulatov, and Sidney Yip. Kinetic monte carlo method for dislocation glide in silicon. Journal of computer-aided materials design, 6(2-3):175–183, 1999.

[88] Wei Cai, Vasily V Bulatov, João F Justo, Ali S Argon, and Sidney Yip. Intrinsic mobility of a dissociated dislocation in silicon. Physical review letters, 84(15):3346, 2000.

[89] S Scarle, CP Ewels, MI Heggie, and N Martsinovich. Linewise kinetic monte carlo study of silicon dislocation dynamics. Physical Review B, 69(7):075209, 2004.

[90] Yue Zhao and Jaime Marian. Direct prediction of the solute softening-to-hardening transition in w–re alloys using stochastic simulations of screw dislocation motion. Modelling and Simulation in Materials Science and Engineering, 26(4):045002, 2018.

[91] Shuhei Shinzato, Masato Wakeda, and Shigenobu Ogata. An atomistically informed kinetic monte carlo model for predicting solid solution strengthening of body-centered cubic alloys. International Journal of Plasticity, 122:319–337, 2019.

[92] Alexander Stukowski, David Cereceda, Thomas D Swinburne, and Jaime Marian. Thermally-activated non-schmid glide of screw dislocations in w using atomistically-informed kinetic monte carlo simulations. International Journal of Plasticity, 65:108–130, 2015.

[93] Wei Cai, Vasily V Bulatov, João F Justo, Ali S Argon, and Sidney Yip. Kinetic monte carlo approach to modeling dislocation mobility. Computational materials science, 23(1-4):124–130, 2002.

[94] Pak Yuen Chan, Georgios Tsekenis, Jonathan Dantzig, Karin A Dahmen, and Nigel Goldenfeld. Plasticity and dislocation dynamics in a phase field crystal model. Physical review letters, 105(1):015502, 2010.

[95] Ebrahim Asadi and Mohsen Asle Zaeem. A review of quantitative phase-field crystal modeling of solid–liquid structures. Jom, 67(1):186–201, 2015.

[96] Mohsen Asle Zaeem and Ebrahim Asadi. Phase-field crystal modeling: Integrating density functional theory, molecular dynamics, and phase-field modeling. Integrated Computational Materials Engineering (ICME) for Metals: Concepts and Case Studies, page 49, 2018.

[97] Mark Ainsworth and Zhiping Mao. Fractional phase-field crystal modelling: analysis, approximation and pattern formation. IMA Journal of Applied Mathematics, 85(2):231–262, 2020.

[98] Douglas Brent West et al. Introduction to graph theory, volume 2. Prentice hall Upper

Saddle River, NJ, 1996.

[99] Michael A Webb, Jean-Yves Delannoy, and Juan J De Pablo. Graph-based approach to systematic molecular coarse-graining. Journal of chemical theory and computation, 15(2):1199–1208, 2018.

[100] Michail Stamatakis and Dionisios G Vlachos. A graph-theoretical kinetic monte carlo framework for on-lattice chemical kinetics. The Journal of chemical physics, 134(21):214115, 2011.

[101] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. Physics reports, 716:1–58, 2017.

[102] Liming Xiong, Garritt Tucker, David L McDowell, and Youping Chen. Coarse-grained atomistic simulation of dislocations. Journal of the Mechanics and Physics of Solids, 59(2):160–177, 2011.

[103] Jorge L Suzuki, Ehsan Kharazmi, Pegah Varghaei, Maryam Naghibolhosseini, and Mohsen Zayernouri. Anomalous nonlinear dynamics behavior of fractional viscoelastic beams. Journal of Computational and Nonlinear Dynamics, 16(11):111005, 2021.

[104] Eduardo A Barros de Moraes, Mohsen Zayernouri, and Mark M Meerschaert. An integrated sensitivity-uncertainty quantification framework for stochastic phase-field modeling of material damage. International Journal for Numerical Methods in Engineering, 122(5):1352–1377, 2021.

[105] Eduardo A Barros de Moraes, Hadi Salehi, and Mohsen Zayernouri. Data-driven failure prediction in brittle materials: A phase field-based machine learning framework. Journal of Machine Learning for Modeling and Computing, 2(1), 2021.

[106] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. J. Comp. Phys., 117:1–19, 1995.

[107] KOE Henriksson, C Björkas, and Kai Nordlund. Atomistic simulations of stainless steels: a many-body potential for the fe–cr–c system. Journal of Physics: Condensed Matter, 25(44):445401, 2013.

[108] P.M. Larsen, S. Schmidt, and J. Schiøtz. Robust structural identification via polyhedral template matching. Modelling Simul. Mater. Sci. Eng., 24(5), 2016.

[109] A. Stukowski. Visualization and analysis of atomistic simulation data with ovito-the open visualization tool. Modelling Simul. Mater. Sci. Eng., 18:015012, 2007.

[110] A Pérez Riascos and José L Mateos. Fractional dynamics on networks: Emergence of anomalous diffusion and lévy flights. Physical Review E, 90(3):032809, 2014.

[111] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[112] Geoffrey Grimmett and Dominic Welsh. Probability: an introduction. Oxford University Press, 2014.

[113] Michael J Evans and Jeffrey S Rosenthal. Probability and statistics: The science of uncertainty. Macmillan, 2004.

[114] Mark M Meerschaert and Alla Sikorskii. Stochastic models for fractional calculus, volume 43. Walter de Gruyter, 2011.

[115] Ting Zhu, Ju Li, and Sidney Yip. Atomistic study of dislocation loop emission from a crack tip. Physical review letters, 93(2):025503, 2004.

[116] K. Tanaka and T. Mura. A Dislocation Model for Fatigue Crack Initiation. Journal of Applied Mechanics, 48(1):97, 1981.

[117] Michael Zaiser. Scale invariance in plastic flow of crystalline solids. Advances in Physics, 55(1-2):185–245, January 2006.

[118] Peter Hähner, Karlheinz Bay, and Michael Zaiser. Fractal Dislocation Patterning During Plastic Deformation. Physical Review Letters, 81(12):2470–2473, September 1998.

[119] David L. Holt. Dislocation Cell Formation in Metals. Journal of Applied Physics, 41(8):3197–3201, July 1970.

[120] Daniel Walgraef and Elias C Aifantis. Dislocation patterning in fatigued metals as a result of dynamical instabilities. Journal of applied physics, 58(2):688–691, 1985.

[121] P Hähner. A theory of dislocation cell formation based on stochastic dislocation dynamics. Acta materialia, 44(6):2345–2352, 1996.

[122] Olga Kapetanou, Vasileios Koutsos, Efstathios Theotokoglou, Daniel Weygand, and Michael Zaiser. Statistical analysis and stochastic dislocation-based modeling of microplasticity. Journal of the Mechanical Behavior of Materials, 24(3-4):105–113, 2015.

[123] Thomas Hochrainer, Stefan Sandfeld, Michael Zaiser, and Peter Gumbsch. Continuum dislocation dynamics: towards a physical theory of crystal plasticity. Journal of the Mechanics and Physics of Solids, 63:167–178, 2014.

[124] Thomas Hochrainer. Multipole expansion of continuum dislocations dynamics in terms of alignment tensors. Philosophical Magazine, 95(12):1321–1367, 2015.

[125] Qiang Du, Zhan Huang, and Richard B Lehoucq. Nonlocal convection-diffusion volume-constrained problems and jump processes. Discrete & Continuous Dynamical Systems-B, 19(2):373, 2014.

[126] Marta D'Elia, Qiang Du, Max Gunzburger, and Richard Lehoucq. Nonlocal convection-diffusion problems on bounded domains and finite-range jump processes. Computational Methods in Applied Mathematics, 17(4):707–722, 2017.

[127] Qiang Du, Robert Lipton, and Tadele Mengesha. Multiscale analysis of linear evolution equations with applications to nonlocal models for heterogeneous media. ESAIM: Mathematical Modelling and Numerical Analysis, 50(5):1425–1455, 2016.

[128] Ali Akhavan-Safaei, Mehdi Samiee, and Mohsen Zayernouri. Data-driven fractional subgrid-scale modeling for scalar turbulence: A nonlocal les approach. Journal of Computational Physics, 446:110571, 2021.

[129] S Hadi Seyedi and Mohsen Zayernouri. A data-driven dynamic nonlocal subgrid-scale model for turbulent flows. Physics of Fluids, 34(3):035104, 2022.

[130] Mehdi Samiee, Ali Akhavan-Safaei, and Mohsen Zayernouri. A fractional subgrid-scale model for turbulent flows: Theoretical formulation and a priori study. Physics of Fluids, 32(5):055102, 2020.

[131] Ali Akhavan-Safaei and Mohsen Zayernouri. A nonlocal spectral transfer model and new scaling law for scalar turbulence. arXiv preprint arXiv:2111.06540, 2021.

[132] Mehdi Samiee, Ali Akhavan-Safaei, and Mohsen Zayernouri. Tempered fractional les modeling. Journal of Fluid Mechanics, 932, 2022.

[133] Rina Schumer, David A Benson, Mark M Meerschaert, and Stephen W Wheatcraft. Eulerian derivation of the fractional advection–dispersion equation. Journal of contaminant hydrology, 48(1-2):69–88, 2001.

[134] Xiao Xu, Marta D'Elia, Christian Glusa, and John T Foster. Machine-learning of nonlocal kernels for anomalous subsurface transport from breakthrough curves. arXiv preprint arXiv:2201.11146, 2022.

[135] Jorge Suzuki, Mamikon Gulian, Mohsen Zayernouri, and Marta D'Elia. Fractional modeling in action: A survey of nonlocal models for subsurface transport, turbulent flows, and anomalous materials. arXiv preprint arXiv:2110.11531, 2021.

[136] SA Silling. Dynamic fracture modeling with a meshfree peridynamic code. In Computational Fluid and Solid Mechanics 2003, pages 641–644. Elsevier, 2003.

[137] Stewart Andrew Silling and Abe Askari. Peridynamic model for fatigue cracking. Technical

Report SAND2014-18590, 1160289, October 2014.

[138] Marta D'Elia, Mamikon Gulian, George Karniadakis, and Hayley Olson. A unified theory of fractional nonlocal and weighted nonlocal vector calculus. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.

[139] Henri Salmenjoki, Mikko J Alava, and Lasse Laurson. Machine learning plastic deformation of crystals. Nature communications, 9(1):1–7, 2018.

[140] Mika Sarvilahti, Audun Skaugen, and Lasse Laurson. Machine learning depinning of dislocation pileups. APL Materials, 8(10):101109, 2020.

[141] Henri Salmenjoki et al. Predicting the behaviour of dislocation systems with machine learning methods. 2017.

[142] Dominik Steinberger, Hengxu Song, and Stefan Sandfeld. Machine learning-based classification of dislocation microstructures. Frontiers in Materials, 6:141, 2019.

[143] Eduardo A Barros de Moraes, Jorge L Suzuki, and Mohsen Zayernouri. Atomistic-to-meso multi-scale data-driven graph surrogate modeling of dislocation glide. Computational Materials Science, 197:110569, 2021.

[144] Joseph Bakarji and Daniel M Tartakovsky. Data-driven discovery of coarse-grained equations. Journal of Computational Physics, 434:110219, 2021.

[145] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I Siettos, and Ioannis G Kevrekidis. Coarse-scale pdes from fine-scale observations via machine learning. Chaos: An Interdisciplinary Journal of Nonlinear Science, 30(1):013141, 2020.

[146] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. Science Advances, 3(4):e1602614, 2017.

[147] Rohit Supekar, Boya Song, Alasdair Hastewell, Alexander Mietke, and Jörn Dunkel. Learning hydrodynamic equations for active matter from particle simulations and experiments. arXiv preprint arXiv:2101.06568, 2021.

[148] Peter M Anderson, John P Hirth, and Jens Lothe. Theory of dislocations. Cambridge University Press, 2017.

[149] Erik Van der Giessen and Alan Needleman. Discrete dislocation plasticity: a simple planar model. Modelling and Simulation in Materials Science and Engineering, 3(5):689, 1995.

[150] Bernard W Silverman. Density estimation for statistics and data analysis. Routledge, 2018.

[151] Daniele Pedretti and Daniel Fernàndez-Garcia. An automatic locally-adaptive method to

estimate heavily-tailed breakthrough curves from particle distributions. Advances in water Resources, 59:52–65, 2013.

[152] Ian S Abramson. On bandwidth variation in kernel estimates-a square root law. The annals of Statistics, pages 1217–1223, 1982.

[153] Jean Lemaitre and Rodrigue Desmorat. Engineering damage mechanics: ductile, creep, fatigue and brittle failures. Springer Science & Business Media, 2005.

[154] Ted L Anderson. Fracture mechanics: fundamentals and applications. CRC press, 2017.

[155] Melvin F Kanninen and Carl L Popelar. Advanced fracture mechanics. 1985.

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

[157] Michael J. Borden, Thomas J.R. Hughes, Chad M. Landis, and Clemens V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. Computer Methods in Applied Mechanics and Engineering, 273:100–118, May 2014.

[158] Marreddy Ambati, Roland Kruse, and Laura De Lorenzis. A phase-field model for ductile fracture at finite strains and its experimental verification. Computational Mechanics, 57(1):149–167, January 2016.

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

[160] G. Amendola, M. Fabrizio, and J. M. Golden. Thermomechanics of damage and fatigue by a phase field model. Journal of Thermal Stresses, 39(5):487–499, May 2016.

[161] Michele Caputo and Mauro Fabrizio. Damage and fatigue described by a fractional derivative model. Journal of Computational Physics, 293:400–408, July 2015.

[162] L.R. Chiarelli, F.G. Fumes, E.A. Barros de Moraes, G.A. Haveroth, J.L. Boldrini, and M.L. Bittencourt. Comparison of high order finite element and discontinuous Galerkin methods for phase field equations: Application to structural damage. Computers & Mathematics with Applications, 74(7):1542–1564, October 2017.

[163] G.A. Haveroth, E.A. Barros de Moraes, J.L. Boldrini, and M.L. Bittencourt. Comparison of semi and fully-implicit time integration schemes applied to a damage and fatigue phase field model. Latin American Journal of Solids and Structures, 15(5):1–16, May 2018.

[164] George Fishman. Monte Carlo: concepts, algorithms, and applications. Springer Science & Business Media, 2013.

[165] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. SIAM journal on scientific computing, 24(2):619–644, 2002.

[166] OM Knio and OP Le Maitre. Uncertainty propagation in cfd using polynomial chaos decomposition. Fluid dynamics research, 38(9):616, 2006.

[167] George Stefanou. The stochastic finite element method: past, present and future. Computer methods in applied mechanics and engineering, 198(9-12):1031–1051, 2009.

[168] Paul G Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4):A1500–A1524, 2014.

[169] Paul G Constantine, Michael Emory, Johan Larsson, and Gianluca Iaccarino. Exploiting active subspaces to quantify uncertainty in the numerical simulation of the hyshot ii scramjet. Journal of Computational Physics, 302:1–20, 2015.

[170] Paul G Constantine and Paul Diaz. Global sensitivity metrics from active subspaces. Reliability Engineering & System Safety, 162:1–13, 2017.

[171] Khader M Hamdia, Mohammed A Msekh, Mohammad Silani, Nam Vu-Bac, Xiaoying Zhuang, Trung Nguyen-Thoi, and Timon Rabczuk. Uncertainty quantification of the fracture properties of polymeric nanocomposites based on phase field modeling. Composite Structures, 133:1177–1190, 2015.

[172] Ernesto A. B. F. Lima, Regina C. Almeida, and J. Tinsley Oden. Analysis and numerical solution of stochastic phase-field models of tumor growth: Stochastic Tumor Growth. Numerical Methods for Partial Differential Equations, 31(2):552–574, March 2015.

[173] Charbel Farhat, Adrien Bos, Philip Avery, and Christian Soize. Modeling and quantification of model-form uncertainties in eigenvalue computations using a stochastic reduced model. AIAA Journal, 56(3):1198–1210, 2018.

[174] Christian Soize and Charbel Farhat. Probabilistic learning for modeling and quantifying model-form uncertainties in nonlinear computational mechanics. International Journal for Numerical Methods in Engineering, 117(7):819–843, 2019.

[175] Joaquim RRA Martins, Peter Sturdza, and Juan J Alonso. The complex-step derivative approximation. ACM Transactions on Mathematical Software (TOMS), 29(3):245–262, 2003.

[176] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and

Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. Computer Physics Communications, 181(2):259–270, 2010.

[177] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3):37–52, 1987.

[178] Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.

[179] Mathilde Chevreuil, Régis Lebrun, Anthony Nouy, and Prashant Rai. A least-squares method for sparse low rank approximation of multivariate functions. SIAM/ASA Journal on Uncertainty Quantification, 3(1):897–921, 2015.

[180] Ilya M Sobol. Sensitivity estimates for nonlinear mathematical models. Mathematical modelling and computational experiments, 1(4):407–414, 1993.

[181] J. Weiss. Three-Dimensional Mapping of Dislocation Avalanches: Clustering and Space/Time Coupling. Science, 299(5603):89–92, January 2003.

[182] Alberto Carpinteri and Francesco Mainardi. Fractals and fractional calculus in continuum mechanics, volume 378. Springer, 2014.

[183] Mark Ainsworth and Zhiping Mao. Analysis and Approximation of a Fractional Cahn–Hilliard Equation. SIAM Journal on Numerical Analysis, 55(4):1689–1718, January 2017.

[184] Mark Ainsworth and Zhiping Mao. Well-posedness of the Cahn–Hilliard equation with fractional free energy and its Fourier Galerkin approximation. Chaos, Solitons & Fractals, 102:264–273, September 2017.

[185] Giambattista Giacomin and Joel L. Lebowitz. Phase segregation dynamics in particle systems with long range interactions. I. Macroscopic limits. Journal of Statistical Physics, 87(1-2):37–61, April 1997.

[186] Giambattista Giacomin and Joel L. Lebowitz. Phase Segregation Dynamics in Particle Systems with Long Range Interactions II: Interface Motion. SIAM Journal on Applied Mathematics, 58(6):1707–1729, December 1998.

[187] Helmut Abels, Stefano Bosia, and Maurizio Grasselli. Cahn–Hilliard equation with nonlocal singular free energies. Annali di Matematica Pura ed Applicata (1923 -), 194(4):1071–1106, August 2015.

[188] Ehsan Kharazmi and Mohsen Zayernouri. Operator-based uncertainty quantification of stochastic fractional partial differential equations. Journal of Verification, Validation and Uncertainty Quantification, 4(4), 2019.

[189] Ehsan Kharazmi and Mohsen Zayernouri. Fractional sensitivity equation method: Application to fractional model construction. Journal of Scientific Computing, 80(1):110–140, 2019.

[190] P Carrara, M Ambati, R Alessi, and L De Lorenzis. A framework to model the fatigue behavior of brittle materials based on a variational phase-field approach. Computer Methods in Applied Mechanics and Engineering, page 112731, 2019.

[191] Martha Seiler, Thomas Linse, Peter Hantschke, and Markus Kästner. An efficient phase-field model for fatigue fracture in ductile materials. arXiv preprint arXiv:1903.06465, 2019.

[192] Yi-Yan Liu, Yong-Feng Ju, Chen-Dong Duan, and Xue-Feng Zhao. Structure damage diagnosis using neural network and feature fusion. Engineering applications of artificial intelligence, 24(1):87–92, 2011.

[193] Adam Santos, Eloi Figueiredo, MFM Silva, CS Sales, and JCWA Costa. Machine learning algorithms for damage detection: Kernel-based approaches. Journal of Sound and Vibration, 363:584–599, 2016.

[194] Dia Al Azzawi, Hever Moncayo, Mario G Perhinschi, Andres Perez, and Adil Togayev. Comparison of immunity-based schemes for aircraft failure detection and identification. Engineering Applications of Artificial Intelligence, 52:181–193, 2016.

[195] Moisés Silva, Adam Santos, Eloi Figueiredo, Reginaldo Santos, Claudomiro Sales, and João CWA Costa. A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges. Engineering Applications of Artificial Intelligence, 52:168–180, 2016.

[196] Adam J Wootton, John B Butcher, Theocharis Kyriacou, Charles R Day, and Peter W Haycock. Structural health monitoring of a footbridge using echo state networks and narmax. Engineering Applications of Artificial Intelligence, 64:152–163, 2017.

[197] Arash Saeidpour, Mi G Chorzepa, Jason Christian, and Stephan Durham. Parameterized fragility assessment of bridges subjected to hurricane events using metamodels and multiple environmental parameters. Journal of Infrastructure Systems, 24(4):04018031, 2018.

[198] Hadi Salehi and Rigoberto Burgueno. Emerging artificial intelligence methods in structural engineering. Engineering structures, 171:170–189, 2018.

[199] Hadi Salehi, Saptarshi Das, Shantanu Chakrabartty, Subir Biswas, and Rigoberto Burgueño. Structural damage identification using image-based pattern recognition on event-based binary data generated from self-powered sensor networks. Structural Control and Health Monitoring, 25(4):e2135, 2018.

[200] Hadi Salehi, Saptarshi Das, Shantanu Chakrabartty, Subir Biswas, and Rigoberto Burgueño.

Damage identification in aircraft structures with self-powered sensing technology: A machine learning approach. Structural Control and Health Monitoring, 25(12):e2262, 2018.

[201] Hadi Salehi, Saptarshi Das, Shantanu Chakrabartty, Subir Biswas, and Rigoberto Burgueño. An algorithmic framework for reconstruction of time-delayed and incomplete binary signals from an energy-lean structural health monitoring system. Engineering Structures, 180:603–620, 2019.

[202] Hadi Salehi, Subir Biswas, and Rigoberto Burgueño. Data interpretation framework integrating machine learning and pattern recognition for self-powered data-driven damage identification with harvested energy variations. Engineering Applications of Artificial Intelligence, 86:136–153, 2019.

[203] Andrea Rovinelli, Michael D Sangid, Henry Proudhon, and Wolfgang Ludwig. Using machine learning and a data-driven approach to identify the small fatigue crack driving force in polycrystalline materials. npj Computational Materials, 4(1):35, 2018.

[204] Hyung Jin Lim, Hoon Sohn, and Yongtak Kim. Data-driven fatigue crack quantification and prognosis using nonlinear ultrasonic modulation. Mechanical Systems and Signal Processing, 109:185–195, 2018.

[205] Hyung Jin Lim and Hoon Sohn. Online fatigue crack quantification and prognosis using nonlinear ultrasonic modulation and artificial neural network. In Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, volume 10598, page 105981L. International Society for Optics and Photonics, 2018.

[206] Zhong-Hui Shen, Jian-Jun Wang, Jian-Yong Jiang, Sharon X Huang, Yuan-Hua Lin, Ce-Wen Nan, Long-Qing Chen, and Yang Shen. Phase-field modeling and machine learning of electric-thermal-mechanical breakdown of polymer-based dielectrics. Nature communications, 10(1):1843, 2019.

[207] Yuksel C Yabansu, Philipp Steinmetz, Johannes Hötzer, Surya R Kalidindi, and Britta Nestler. Extraction of reduced-order process-structure linkages from phase-field simulations. Acta Materialia, 124:182–194, 2017.

[208] Stefanos Papanikolaou, Michail Tzimas, Andrew CE Reid, and Stephen A Langer. Spatial strain correlations, machine learning, and deformation history in crystal plasticity. Physical Review E, 99(5):053003, 2019.

[209] GH Teichert, AR Natarajan, A Van der Ven, and K Garikipati. Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions. Computer Methods in Applied Mechanics and Engineering, 353:201–216, 2019.

[210] Abigail Hunter, Bryan A Moore, Maruti Mudunuru, Viet Chau, Roselyne Tchoua, Chandramouli Nyshadham, Satish Karra, Daniel O'Malley, Esteban Rougier, Hari Viswanathan,

et al. Reduced-order modeling through machine learning and graph-theoretic approaches for brittle fracture applications. Computational Materials Science, 157:87–98, 2019.

[211] Bryan A. Moore, Esteban Rougier, Daniel O'Malley, Gowri Srinivasan, Abigail Hunter, and Hari Viswanathan. Predictive modeling of dynamic fracture growth in brittle materials with machine learning. Computational Materials Science, 148:46–53, June 2018.

[212] Max Schwarzer, Bryce Rogan, Yadong Ruan, Zhengming Song, Diana Y Lee, Allon G Percus, Viet T Chau, Bryan A Moore, Esteban Rougier, Hari S Viswanathan, et al. Learning to fail: Predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks. Computational Materials Science, 162:322–332, 2019.

[213] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, (4):580–585, 1985.

[214] Mohamad H Hassoun et al. Fundamentals of artificial neural networks. MIT press, 1995.

[215] Guoqiang Zhang, B Eddy Patuwo, and Michael Y Hu. Forecasting with artificial neural networks:: The state of the art. International journal of forecasting, 14(1):35–62, 1998.

[216] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 10(Feb):207–244, 2009.

[217] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. Neural computation, 4(1):1–58, 1992.

[218] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001.