GLOBAL-LOCAL TECHNIQUES FOR FRACTURE

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Matthias Birner

aus

München

Bonn 2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Marc Alexander Schweitzer

2. Gutachter: Prof. Dr. Joscha Gedicke

Tag der Promotion: 04.12.2023
Erscheinungsjahr: 2023

ABSTRACT

In the numerical simulation of components and structures it is necessary to incorporate the effect of localized damage and its propagation. However, simulating fracture on top of linear elasticity in three space dimensions significantly increases the computational cost. The need to choose a law to predict the crack path introduces further modeling decisions that are still an open research topic. Other approaches such as the phase-field method or peridynamics handle fracture naturally but are even more expensive to simulate. In this thesis, we demonstrate how the global-local enrichments method can improve this by either separating the degrees of freedom required for the resolution of the crack, or by moving a whole expensive method onto a different, local discretization whose solution is then used as a basis function on the global problem. To this end, we use the flat-top partition of unity method (PUM) due to its flexibility in the application of enrichments.

Specifically, this thesis contributes the following. First, we bring the simulation of three-dimensional fracture to the PUM. We address the challenges introduced by complex crack geometries and validate our implementation in numerical experiments. Second, we adapt the global-local enrichments method, such that its parallelization opportunity can be exploited efficiently with the flat-top cover construction in the PUM. We evaluate the accuracy, parallel scaling and performance of the resulting method in numerical experiments in three space dimensions. Parallel scaling of the resulting method is excellent and especially weak scaling is optimal. It further achieves the same accuracy as applying the degrees of freedom from the local problem directly on the global problem. The global-local method is thus a viable alternative for direct h-refinement in the simulation of fracture. Lastly, we generalize the global-local method to a coupling method that allows the flexible combination of the PUM with different simulation frameworks. Here, we present numerical results with peridynamics and the phase-field method on the local problem, which serve as a crack growth law for the global linear elasticity simulation. The global-local coupling approach produces similar results as traditional direct coupling along an interface and the obtained crack paths are in good agreement with real world experimental data.

ACKNOWLEDGMENTS

Before we begin, I would like to take the opportunity to thank all those who have made this work possible. First, I want to thank my supervisor Marc Alexander Schweitzer for introducing me to the field of Numerical Simulation and for the freedom to pursue my own interests during this journey. I owe special thanks to colleague Ziegenhagel, who mentored me through much of the implementation work that formed the base of this thesis. This said, I want to thank the entire PUMA group at Fraunhofer SCAI and the Institute for Numerical Simulation at the University of Bonn for the numerous discussions and valuable feedback. Thank you for sharing this journey with me – it was a pleasure. Moreover, I want to express my appreciation to Patrick Diehl and Robert Lipton for our collaboration on the peridynamics coupling. Last but not least, I thank my family for the continuous support of my studies.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CIM	contour integral method
DCM	displacement correlation method
FEM	finite element method
FEMGL	global-local finite element analysis
GECO	generic expression compiler for PUMA
GFEM	generalized finite element method
GFEM^{gl}	generalized finite element method using global-local enrichments
MPI	message passing interface
PDE	partial differential equation
PU	partition of unity
PUM	partition of unity method
PUMA	Fraunhofer software framework implementing the PUM
XFEM	extended finite element method
XFIELD	hybrid XFEM phase-field method

LIST OF SYMBOLS

$\mathbb{N}$	Natural numbers.
$\mathbb{R}$	Real numbers.
H^1	Sobolev space, p=2, k=1.
L^2	Lebesgue space, p=2.
Ω	Computational domain.
Γ	Boundary of a domain.
n	Normal to a boundary.
$B_r(x)$	Ball with radius r around x.
I	Identity matrix.
card	Cardinality of a set.
span	Space spanned by vectors or functions.
supp	Support of a function.
$\varepsilon(u)$	Strain.
$\sigma(u)$	Stress.
$\mathcal{T}$	Traction on a boundary.
ν	Poisson's ratio.
E	Young's modulus.
G	Shear modulus.
ϑ	Basis function.
V	Function space.
C_Ω	Cover of domain Ω.
ω	Patch in a cover C_Ω.
W	Weight function.
φ	PU function.
$\mathcal{P}$	Space of polynomials.
$\mathcal{E}$	Space of enrichment functions.
η	enrichment function
V^{PU}	Ansatz space of the PUM.
$\mathcal{H}$	Heaviside function.
τ	Buffersize factor in global-local method.
κ	Parameter in Robin boundary conditions.

1 INTRODUCTION

Numerical analysis of the simple model of linear elasticity is computationally cheap enough to be a common step in the design process of vehicles and structures. In this process, it is of further interest to incorporate the effect of localized damage present in components on their toughness as well as predicting the propagation of the damage [16]. The industry standard to perform such simulations is the mesh based finite element method (FEM), where the mesh serves two purposes: First of all, it represents the geometry and cracks, which are encoded in it as an actual boundary of the domain. Consequently, the crack geometry is limited by the resolution of the mesh and the mesh has to be updated if the crack grows. Second, it defines the incorporated polynomial basis functions and thus has direct impact on the quality of the computed solution. In linear elastic fracture mechanics, the area around the crack front is of special interest since the solution exhibits a singularity there. In order to accurately predict the crack growth direction, this singularity has to be captured in the numerical approximation. This introduces a lot of degrees of freedom locally, which increases the computational cost of the simulation. Furthermore, when the crack propagates, the mesh has to be adapted to the moving crack front, which introduces additional refinement and coarsening steps. The number of required degrees of freedom and remeshing steps are thus challenges in the application of FEMs to the simulation of fracture propagation in three space dimensions [71, 133].

Several methods exist that can work around those issues by employing special basis functions, so-called enrichments. In fracture mechanics, for instance, enrichments can be used instead of the mesh to model the crack and its singularity, thereby reducing the number of required degrees of freedom significantly and further avoiding the need to update the mesh. Examples of such methods are the element free Galerkin method [7], the reproducing kernel particle method [78] and hp-clouds [24], all of which are based on ideas of the moving least squares method. In practice, however, the concept of a partition of unity method [83] has prevailed to allow for enriched approximations. At the core of this method is a partition of unity (PU) $\{\varphi_i\}$ subordinate to some cover of the computational domain, which allows to create an ansatz space by multiplying basis functions to each PU function separately

$$V = \sum_i \varphi_i(\mathcal{P}_i + \mathcal{E}_i).$$

By this construction, we can decide locally at each φ_i which basis functions to use, where the spaces $\mathcal{P}_i$ denote polynomial basis functions and the spaces $\mathcal{E}_i$ consist of problem dependent basis functions added to the simulation to improve accuracy, called enrichments. Popular examples of partition of unity methods that have widely and successfully been applied to the simulation of fracture mechanics are the generalized finite element method (GFEM) [22] and the extended finite element method (XFEM) [86, 134]. However, both methods

are finite element and thus mesh based, although cracks can be modeled by the employed enrichment functions. Besides other consequences, the mesh based construction affects properties of the underlying PU, especially regarding stability of the enriched basis. Adding enrichments as basis functions can introduce linear dependencies in the resulting ansatz space: either locally with respect to the polynomial spaces $\mathcal{P}_i$, between different enrichments themselves, or globally due to the overlap of the PU functions φ_i. The latter can occur in the GFEM/XFEM since neighboring PU functions overlap completely and thus have no area in which they are constant one, that is, no so-called flat-top zone. One technique to improve this has been introduced as the stable GFEM [3, 47, 48]. Here, its interpolation in the polynomial part of the basis is subtracted from an enrichment function. Yet, this only partially addresses the above mentioned issues, as for example dependencies between different enrichments are not considered and only individual enrichments are near-orthogonalized with respect to the polynomial basis [45]. In this construction, it is further difficult to decide whether an enrichment is actually in the span of the polynomial basis on some patch, due to only evaluating on the mesh nodes. For this reason, we use the meshfree PUM introduced in [114, 115] in this thesis. Due to construction of the cover of the computational domain, the PUM has a flat-top partition of unity which allows to efficiently generate a globally stable and numerically orthogonal basis [117]. The PUM thus allows the flexible use of enrichment functions.

Dramatically reducing the required degrees of freedom by the use of enrichment functions, however, is only possible if such functions are available analytically for the given problem. For the singularity arising in fracture simulations, such functions are known in two and for trivial crack geometries in three space dimensions, yet unavailable for complex crack geometries in three space dimensions. Here, adaptive h-refinement towards the crack front is again necessary to be locally back in the case of a simple straight crack front, where good enrichment functions are known. So while enrichments still provide some benefit in reducing the required refinement, the number of degrees of freedom increases substantially around the crack front. Suitable enrichments, however, can always be generated by computing them either prior to or during a simulation. In the first case, the goal is to generate functions capturing some local feature preferably agnostic of, for example, the specific material by solving a simplified or generalized problem [2]. Then, one could choose from a handbook of functions for a given simulation, as envisioned in [132]. Fracture simulation, however, yields complex and a priori unpredictable crack geometries, making it unlikely to be able to precompute enrichments for them. In these cases, it is still possible to compute an enrichment on the fly, during a simulation, by solving a localized version of the global, overall problem. In fracture simulation, this moves the degrees of freedom resolving the crack front onto a different discretization, thereby separating scales. This reduces the computational cost associated with the global discretization for all further global operations.

On the fly computation of enrichments was introduced as the generalized finite element method using global-local enrichments (GFEM^{gl}) [23] and has since been successfully applied to a variety of problems, including fracture. We refer to the method abstracted from the specifics of the GFEM context as the global-local enrichments method. It builds on the global-local or zooming methods [27, 70, 88, 136] developed in the FEM context since the 1970s. These are

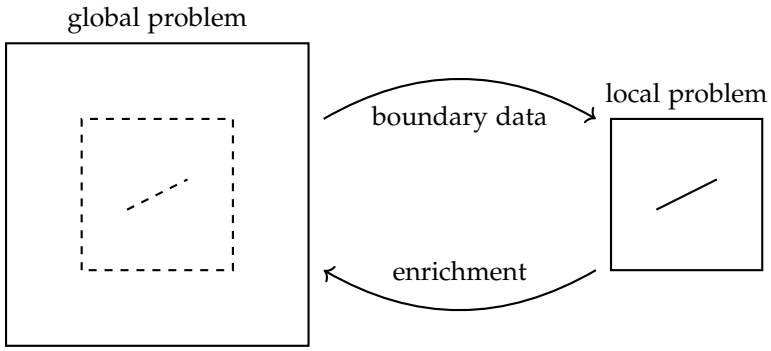


Figure 1.1: The global-local enrichments method for partition of unity methods as introduced in [23].

more akin to classical domain decomposition methods without overlap [37], as the local problem directly constitutes the solution in its domain. In the global-local enrichments method, however, the final solution is completely given by a coarse global discretization, with the fine scale feature captured by the computed enrichment. To achieve this, the global problem is initially solved disregarding the local feature to provide boundary data for the local problem. The solution of the local problem now capturing the local feature is used as an enrichment on the global problem, which then has to be solved again, as depicted in Figure 1.1. Multiple iterations of this may be required to achieve the full accuracy that is possible with the invested degrees of freedom, by successively improving the boundary data on the local problem and thereby the computed enrichment. As a result, it is possible to have the approximation power provided by the local ansatz space while only adding one degree of freedom per space dimension and enriched patch to the global problem. All further global operations such as linear or nonlinear solvers then benefit from the reduced global problem size. For sequential simulations, the global-local method works mostly similar in the meshfree PUM, as we already demonstrated on simple two-dimensional problems in [9], leading up to the work on this thesis. Besides the rearranging of degrees of freedom, the global-local method additionally provides an excellent parallelization opportunity, as local problems are completely independent of each other. However, due to the construction of the ansatz spaces in the meshfree PUM, h-refined spaces are not nested in their parents and adaption of the global-local method is required for it to work efficiently in parallel. In this thesis, we now propose an efficient parallel global-local method in the meshfree PUM and evaluate it in numerical experiments in three space dimensions.

Another issue with three-dimensional fracture simulation mentioned in [133] is the lack of a 'widely accepted' crack growth law in linear elastic fracture mechanics. However, other models exist that handle fracture initiation and propagation naturally [16]. In this thesis, we focus on two methods that seem especially promising [21]. First, peridynamics [126], because it includes aspects of atomistic models not present in linear elasticity and because it has been successfully applied to the simulation of a wide range of fracture processes [58]. And second, the phase-field method [28], because it is a framework to represent fracture, with the above-mentioned advantages, that can be formulated with different material models with different material properties [74]. We expand on

this in Section 2.1, but in short, the two models differ from linear elasticity in the following ways: The peridynamic model is given by an integral equation and not a partial differential equation (PDE), which naturally allows for the discontinuities across cracks as no derivatives are involved. Furthermore, force is transferred between material points that are not immediate neighbors, inside the so-called horizon of a point. Peridynamics is thus a non-local method. In the phase-field method, a second nonlinear PDE that models the energy dissipation by the crack is coupled with the original PDE of linear elasticity. In both methods the crack location is only determined up to the mesh width and accurate results thus require a fine discretization. Combined with the interaction across the horizon and the staggered nonlinear iteration respectively, both methods are computationally very expensive [21] and it is therefore desirable to use them only locally where needed [16, 58]. On this topic, we want to highlight that the global-local method depicted in Figure 1.1 also provides an excellent coupling opportunity as the local problem does not need to be solved with the same method or even has to discretize the same material model as the global. To be applicable, a method only needs to accept boundary data from a global PUM simulation and be able to provide some form of enrichment in return. In this thesis, we thus generalize global-local enrichments to a coupling method and present results with peridynamics and the phase-field method used on the local problem. We want to note, though, that the presented coupling approach is not limited to those two methods.

Several approaches to couple the non-local peridynamic model to the PDE model of linear elasticity mostly discretized by finite element based methods have already been proposed and [18] provides a good overview over them. However, all these coupling approaches have in common that the final global solution is given by a combination of the two methods, the peridynamics solution locally around the crack and the linear elasticity discretization in the remainder of the computational domain. This introduces an artificial interface, across which information has to travel from one model to the other, which usually causes problems such as wave reflection [34]. As it is based on the finite element based partition of unity method XFEM, we explicitly mention the approach presented in [35]. In this particular approach, a local peridynamic region is set up around the crack tip, while the rest of the crack is modeled by the standard enrichment functions. Yet, here too, an interface to the global XFEM discretization is created. A coupling method based on global-local enrichments gets rid of this interface by solving the global solution with the cheap linear elasticity model exclusively, while still benefitting from the natural crack propagation in the peridynamic model through the solution of the local problem. During the work on this thesis, we already presented such a coupling with peridynamics in [8] in joint work with P. Diehl and R. Lipton. In this thesis, we present selected results of this publication together with new experiments, where we reproduce physical experiments with the coupled method for the first time and compare the results.

Several coupling approaches of the phase-field method particularly with the XFEM have been proposed too [90]. We especially want to highlight the hybrid XFEM phase-field method (XFIELD) [36], as it is a coupling approach very similar to the one presented in this thesis. The global-local method has further been applied to an uncoupled phase-field formulation [31] to improve its runtime

performance. In this thesis, we present initial results highlighting the possibility of this combination in the global-local method.

In summary, the contributions of this monograph to the simulation of fracture in the meshfree PUM and to the application of the global-local method to fracture simulation thus are the following.

- We bring the simulation of three-dimensional fracture to the meshfree PUM and address the challenges introduced by complex crack geometries.
- We adapt the global-local enrichments method, such that its parallelization opportunity can be exploited efficiently with the flat-top cover construction in the PUM.
- We generalize global-local enrichments to a coupling method, to benefit from other material and fracture models locally in simulations of linear elasticity.

The remainder of this thesis is structured as follows.

In Chapter 2, we start with an overview over fracture modeling in both linear elastic fracture mechanics as well as peridynamics. We then expand on the former model and detail the singular displacement functions dominating the linear elasticity solution close to a crack front. Building on them, we explain the concept of stress intensity factors that are used to predict the rate and direction of crack growth. To be able to use the stress intensity factors in a simulation, we present the analysis part of their extraction from a displacement function. Next, we introduce the PUM at the core of this thesis, which we use to discretize and solve linear elastic fracture mechanics. Specifically, we describe the construction of the flat-top partition of unity that enables the stable transformation, which in turn allows the flexible use of enrichment functions. We outline the parallelization of the method as well as the construction of integration cells for the assembly of the linear system, as we need that context in our parallel implementation of the global-local method.

Chapter 3 then is dedicated to challenges in the simulation of linear elastic fracture mechanics in the PUM introduced by complex three-dimensional crack geometries. Here, we especially detail the implementation of fracture simulation in the Fraunhofer software framework implementing the PUM (PUMA) [112, 120], which is used for all PUM simulations in this thesis. The implementation is along the lines of what was presented in the GFEM context [44, 104]. However, handling of complex three-dimensional fracture had to be realized in PUMA, as prior to this thesis only two-dimensional cracks were supported. Specifically, we discuss the geometric representation of cracks chosen in our implementation. This representation is used to generate integration cells resolving the crack surface and has to provide efficient queries on the crack geometry for the evaluation of the crack specific enrichment functions. For those enrichment functions we present and compare methods to handle the geometric challenge introduced by kinks in the crack front, where the singularity is unknown. Regarding the extraction of stress intensity factors, we detail the parallel implementation of the contour integral method (CIM) and the displacement correlation method (DCM) and test its soundness in numerical experiments. Lastly, we discuss the update of the geometry representation during crack growth. We conclude this chapter with numerical experiments to validate our implementation.

In Chapter 4 we then present the global-local method and adapt it to the meshfree PUM. The primary challenge in the adaptation is that it is difficult to construct nested ansatz spaces in the meshfree PUM, which complicates efficient evaluation of the computed enrichments compared to the GFEM^{gl}, especially if local problems overlap. As a consequence, we restrict ourselves to having non-overlapping local problems, but in contrast to the GFEM^{gl} [66] solve them in parallel instead of sequentially. Parallel local problems complicate the implementation, since we now have to exchange the neighbor relationships between the parallel distributions of different function spaces. In order to exploit the inherent parallelism of the global-local method, it is further necessary to restrict individual local problems to subsets of the overall available processes. To this end, coordination of different communication units is necessary in the implementation. As a result of these design choices, we present an efficient and distributed memory parallel global-local method in the meshfree PUM in this thesis. We conclude this chapter by evaluating the method and our implementation in numerical experiments. We assess its accuracy, especially in comparison to direct, h-refined solutions. Furthermore, we measure the impact of several techniques to improve the accuracy of the global-local method in the first iterations, such as a bufferzone [49] or Robin boundary conditions [68]. Lastly, we evaluate the parallel scaling and performance of our adapted implementation in the PUM.

In Chapter 5 we generalize the global-local method to a coupling method, where different methods or discretizations can be used on the local problem easily. While the approach is not limited to those, we present results with peridynamics and with the phase-field method on the local problem. We address the following challenges. First of all, we want to note that by combining linear elastic fracture mechanics with either method, but especially with peridynamics, we combine two different material models, that can differ in application of load and the response thereto. Hence, the first step is to verify that under similar load conditions and in the linear regime of the peridynamic model both models yield comparable displacements. This is necessary for them to be compatible along the boundary of the local domain. The second challenge is the lack of a discrete, sharp boundary in peridynamics, in contrast to the linear elastic PDE model. In the global-local method, we thus assign boundary conditions on a layer of peridynamic nodes. Moreover, we do not directly apply the global conditions on overlapping parts of the local boundary, but instead transmit it through the global solution. Third, it might not be possible to use the computed displacement as an enrichment function, since the solution of the different local peridynamic or phase-field material model may not minimize the global energy. An alternative is to only extract the crack path from the local solution and to model it with the standard crack enrichments in the global PUM problem, thereby using the local model as a crack growth law. We chose the latter approach, as crack path extraction has to be performed anyways as soon as we move the local domain with the propagating crack. The initial results in [8] on uncracked and cracked media look promising and we can further present a so far unpublished comparison with real world experimental data. We conclude this chapter with initial results of coupling the PUM with a local phase-field model also discretized by the PUM. To this end, we run a coupled experiment with automatic crack

path extraction in two space dimensions and show a simulation in three space dimensions as an outlook.

To conclude the thesis, we present several practical use cases in Chapter 6, which can now be simulated with PUMA due to the implementation carried out for this thesis. Especially notable here is the new possibility of three-dimensional fracture simulation in PUMA, which enables a variety of new research possibilities. Lastly, we discuss the findings of this thesis in Chapter 7.

2 FOUNDATION

In this thesis, we apply global-local enrichments to the simulation of fracture. To lay the foundation, we first introduce ways of modeling fracture in continuum mechanics in Section 2.1 and provide more specifics of linear elastic fracture mechanics in Section 2.2. In Section 2.3, we introduce the partition of unity method that enables the use of enrichments and thus the global-local method. Throughout this thesis we use the PUM to discretize the equations of linear elasticity. Other models of fracture are revisited in Chapter 5, where the global-local method is used to couple different material and thus fracture models.

2.1 FRACTURE MODELING

In this section, we are concerned with the modeling of fracture especially in linear elasticity. To get a clear overview over the modeling assumptions, we start with (classical) continuum mechanics and linear elasticity. In continuum mechanics, physical bodies are assumed to be a continuous mass rather than interacting particles. Classical continuum mechanics further assumes that only directly neighboring material points transfer force, hence analysis can be restricted to infinitesimal small regions around points [58, 121]. In the following we motivate the model of linear elasticity. The presentation in this thesis is based on [13] and we refer to [129] for more details.

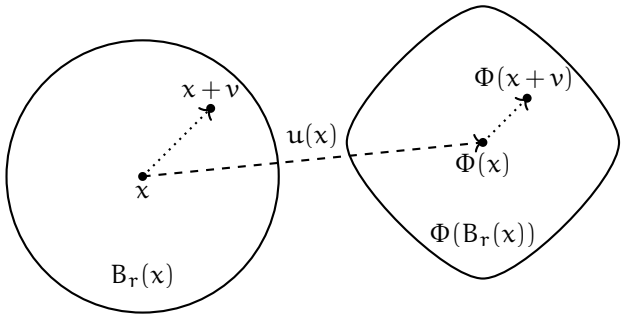


Figure 2.1: Deformation $\Phi(B_r(x))$ of a neighborhood $B_r(x)$ along the displacement u.

Consider a body Ω that is deformed into $\Phi(\Omega)$ under some load, where $\Phi(x) \coloneqq x + u(x)$ is assumed to be a smooth function and $u(x)$ is the displacement at each point, compare Figure 2.1. To measure the deformation or strain the body is experiencing, we compare the squared distance of two points x and $x+v$ with that of their images $\Phi(x)$ and $\Phi(x+v)$. Since we are only interested in points infinitesimally close to x, we can approximate $\Phi(x+v) - \Phi(x) \approx D\Phi(x)v =: F(x)v$ using Taylor expansion, where $F = I + Du$ is the deformation gradient. We can thus write

$$\|\Phi(x+v) - \Phi(x)\|^2 \approx v^T(F^TF)(x)v \quad \text{and} \quad \|x+v-x\|^2 = v^TIv. \tag{2.1}$$

We then use half the difference of these distances to define the strain

$$E(x) := \frac{1}{2}\Big[(F^T F)(x) - I\Big] = \frac{1}{2}\Big[Du + (Du)^T + (Du)^T Du\Big]. \tag{2.2}$$

Note that this strain definition is inherently nonlinear due to the quadratic term on the right hand side, which, from a numerical perspective, is expensive to simulate and only necessary for large displacements or strains. In this thesis, we are only interested in infinitesimal strain theory, as opposed to e. g. finite strain [72] theory, and exclusively use the linear infinitesimal strain tensor

$$\varepsilon(u) = \frac{1}{2}\Big[Du + (Du)^T\Big] \tag{2.3}$$

given by dropping the quadratic term in (2.2). The strain is related to stress by the linear Hooke's law [129]

$$\sigma(u) = C : \varepsilon(u), \tag{2.4}$$

where Hooke's tensor C represents material parameters and $\sigma(u)$ is the materials stress due to the deformation. In general anisotropic materials, C has 21 independent components. If we restrict ourselves to isotropic materials, however, Hooke's tensor is determined by two material constants only [13, 129]: Using the Lamé parameters λ and G, the stress $\sigma(u)$ is given by

$$\sigma(u) = 2G\varepsilon(u) + \lambda \operatorname{tr}(\varepsilon(u))I, \tag{2.5}$$

where G is also referred to as the shear modulus. On a bounded domain $\Omega \subset \mathbb{R}^d$ requiring balance of external and internal forces, the governing equilibrium equation is then given by [14]

$$\begin{aligned} \nabla \cdot \sigma(u) &= b && \text{in } \Omega \\ u &= \bar{u} && \text{on } \Gamma^D \subset \partial\Omega \\ \sigma(u) \cdot n &= \bar{t} && \text{on } \Gamma^N = \partial\Omega \setminus \Gamma^D, \end{aligned} \tag{2.6}$$

where b are volume forces acting on the domain and $T(u) := \sigma(u) \cdot n$ is the traction with respect to the outward unit normal n on the boundary $\partial\Omega$. The boundary data $\bar{u}$ and $\bar{t}$ are the prescribed displacement and traction, respectively. The model (2.6) is called geometrically linear if a linear strain function is chosen, and has linear material if a linear stress is chosen. When interested in a dynamic problem in time and not in the equilibrium solution, the second derivative $\ddot{u}$ of the displacement in time t is not zero and the equation of motion reads

$$\begin{aligned} \rho(x)\ddot{u}(x,t) &= -\nabla \cdot \sigma(u)(x,t) + b(x,t) && \text{in } \Omega \times [0,T] \\ u(x,0) &= u_0(x) && \text{in } \Omega \\ u(x,t) &= \bar{u}(x,t) && \text{on } \Gamma^D \times (0,T] \\ T(u)(x,t) &= \bar{t}(x,t) && \text{on } \Gamma^N \times (0,T], \end{aligned} \tag{2.7}$$

with material density ρ, initial state u_0 and boundary conditions $\bar{u}, \bar{t}$ in time.

Fracture is not easily representable in this model. First of all, the displacement has to be discontinuous across the crack, hence the PDE is not valid there.

In numerical simulation this is either bypassed by introducing an artificial boundary into the domain or by the use of specialized basis functions when solving the weak formulation. This of course requires exact knowledge of the crack location and thus crack initiation is not possible in this model. Furthermore, from a physics point of view we know that except for perfectly brittle fracture, a process zone occurs around the crack tip [99], where the material shows plastic deformations and the response is highly nonlinear. In the PDE model (2.6) with (2.5) as constitutive relation and (2.3) as the strain, one can show that a solution has infinite stress towards the crack tip [129], which does not occur in practice, though. From the PDE solution one can, however, extract coefficients K_I, K_{II} and K_{III}, the so-called stress intensity factors, which describe that singularity close to the crack front [55], see Section 2.2 for details.

The assumption in linear elastic fracture mechanics is that the process zone is small compared to, for example, the crack length and that the solution inside it is essentially determined by the stress intensity factors. This is also known as the K fracture model [41], compare Figure 2.2. Of course, modeling the plastic zone at the crack tip is possible, but expensive, since such models are nonlinear [64, 65].

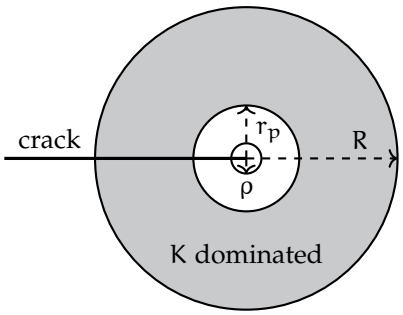
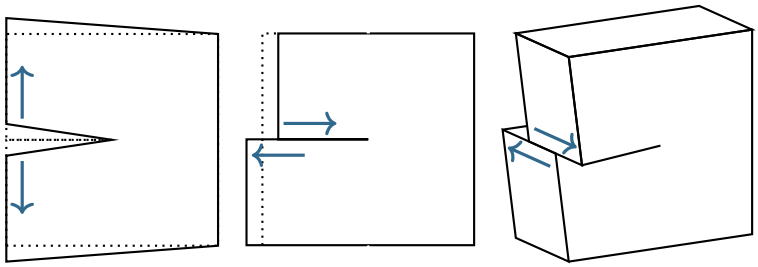


Figure 2.2: Plastic zone inside radius $0 < r_p$ and process zone inside radius $\rho < r_p$. Between r_p and R we have the K dominated zone. Further away from the tip than R, the singularity does not dominate the elasticity solution and other components have to be considered as well. Inside the plastic or even the process zone, the elastic model is not meaningful. However, in the K model, we assume the material response there to be governed by the stress intensity factors.

The stress intensity factors determine the modes of crack opening (I), sliding (II) and tearing (III), sketched in Figure 2.3. That is, they determine the respective contributions of those modes to the local stress. However, if the crack grows at all, in which direction and how far is still undefined. To this end, a crack growth law is required, which in a general form is not available in linear elastic fracture mechanics [16]. For two classes of crack growth, unstable and fatigue crack growth, several estimates that fit experimental data to varying degrees exist however.

Fatigue crack growth happens in the cyclic application of medium loads [109]. In pure mode I loading, it occurs if the stress intensity factor range

$$\Delta K_I := K_{I,max} - K_{I,min} \tag{2.8}$$



(a) Mode I. (b) Mode II. (c) Mode III.

Figure 2.3: Crack opening modes in three space dimensions.

in a load cycle is between a material specific threshold value $\Delta K_{I,th}$ and a critical value $\Delta K_{I,C}$. That is, when

$$\Delta K_{I,th} < \Delta K_I < \Delta K_{I,C}. \tag{2.9}$$

In this case, several crack growth equations are available to predict the crack growth rate, i. e. the crack update in each load cycle. Paris law [101] is probably the most prominent here. In mixed-mode loading a cyclic equivalent stress intensity factor $\Delta K_{eq}(K_I, K_{II}, K_{III})$ can be computed and used in the mode I models. Here, the update direction can be estimated as in unstable crack growth.

Unstable crack growth occurs [109] if K_I in mode I loading, or an equivalent stress intensity factor $K_{eq}(K_I, K_{II}, K_{III})$ in mixed-mode loading, is greater than the material specific critical fracture toughness K_{IC}, which needs to be determined experimentally. Here, the crack update length cannot be estimated and the crack path induced by a static load has to be solved incrementally. Several criteria to predict the crack growth direction based on the extracted stress intensity factors exist [109] and we present one in Section 2.2.3.

In summary, crack growth simulation based on the solution of linear elasticity is possible, but several modeling choices have to be made along the way. Crack initiation is also out of scope. Of course, different approaches of modeling fracture have been proposed [16]. In this thesis, we consider the phase-field method [28] and the peridynamics [126]. Both differ in their crack representation from linear elastic fracture mechanics in that they do not have a sharp discontinuity across the crack, but rather smear or regularize it across an area. By this, 'the crack is not realized as a topological discontinuity but rather through its constitutive behavior' [16], which avoids the singularity introduced in the PDE model, but requires fine resolution globally to obtain accurate crack paths.

In phase-field methods [11, 12, 28] for brittle fracture a second field is introduced that models the crack continuously from uncracked to cracked material. This phase field is the solution of a second nonlinear PDE that models the energy dissipation by the crack and is coupled to the original PDE of linear elasticity. As a consequence, cracks initiate and evolve naturally at the cost of a smeared crack location, different crack opening width compared to a discontinuous crack representation and high computational cost [36]. We want to note that the phase-field method is an approach to model fracture and not a specific material model, i. e. various models with different material properties exist in that framework, see [74] for a review.

In peridynamics [126], the force transmission in the material is modeled by an integral equation. Thereby, the model gets rid of derivatives and the associated problem of discontinuities across cracks. While peridynamics is a continuum mechanics model, it differs from classical continuum mechanics in that material points interact across a radius δ with each other. Such interactions between non-neighboring particles have been observed in theoretical and experimental studies [121]. Peridynamics is therefore a non-local model and models slightly different physics than classical continuum mechanics. However, many formulations recover the linear elastic material model in the limit of vanishing horizon $\delta \to 0$ under some assumptions, notably the absence of cracks [26, 128]. Note that we discuss bond-based peridynamics in this thesis only.

In bond-based peridynamics the equation of motion is given by [126]

$$\rho(x)\ddot{u}(x,t) = \int_{B_\delta(x)} f_{PD}\big(y-x,\, u(y,t)-u(x,t)\big)\,dy + b(x,t), \tag{2.10}$$

where ρ is the material density, u the displacement, b the external force density and the pair-wise force density f_{PD} encodes the material model. In this thesis, we consider the bond-based softening model presented in [75, 76], where the pair-wise force density

$$f_{PD}(\Delta x, \Delta u) \coloneqq \frac{\partial_S \psi_{PD}(\Delta x, \Delta u)}{|\Delta x|} e_{\Delta x} \tag{2.11}$$

is given by the derivative of the pair-wise force potential ψ_{PD} with respect to the bond stretch S and

$$e_{\Delta x} \coloneqq \frac{y-x}{|y-x|} \tag{2.12}$$

is a unit vector. The bond stretch S between two points x, y is determined by their difference $\Delta x = y - x$ and the difference between their respective displacements $\Delta u = u(y,t) - u(x,t)$ by

$$S(\Delta x, \Delta u) \coloneqq \frac{\Delta u}{|\Delta x|} \cdot e_{\Delta x} = \frac{u(y,t)-u(x,t)}{|y-x|} \cdot \frac{y-x}{|y-x|}. \tag{2.13}$$

The bond stretch derivative of the pair-wise force potential

$$\partial_S \psi_{PD}(\Delta x, \Delta u) = J^\delta(|\Delta x|) \frac{\partial_S g_{PD}\big(|\Delta x|\, S^2(\Delta x, \Delta u)\big)}{\delta\, \mu(B_\delta(0))} \tag{2.14}$$

is defined by the influence function J^δ, which is chosen to be piecewise constant

$$J^\delta(r) \coloneqq \begin{cases} 1 & \text{if} \quad 0 \leqslant r < \delta \\ 0 & \text{else} \end{cases} \tag{2.15}$$

in this thesis, and the double well potential

$$g_{PD}(r) \coloneqq C(1 - \exp[-\beta r]), \tag{2.16}$$

where C and β are material parameters. Note that no derivatives of the displacement u appear in the peridynamic formulation (2.10), which is why discontinuities across cracks pose no challenge.

Given the critical energy release rate G_c, Young's modulus E and Poisson's ration ν, the material parameters are given by

$$C \coloneqq \pi \frac{G_c}{4} \quad \text{and} \quad \beta \coloneqq \frac{4E\nu}{C(1-\nu)(1-2\nu)}. \tag{2.17}$$

Note, however, that in bond-based peridynamics Poisson's ratio is fixed to $1/3$ for plain-strain [69], hence our peridynamics material model is defined by the energy release rate and Young's modulus only. Figure 2.4 further shows the derivative of the double well potential g_{PD} (2.16) as used in the force density (2.11). Here, the force initially increases almost linearly with the bond stretch and then decays to zero after some critical bond stretch r_c. This implies that no bonds get broken and cracks will heal upon removing the load, unlike in other peridynamic models.

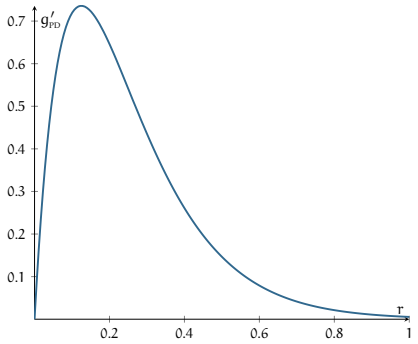


Figure 2.4: Derivative of double well potential g_{PD} with parameters $C = 2$ and $\beta = 8$. Force between points increases until material capacity is reached, where bonds begin to soften, but never break.

In Chapter 5 we discuss combining the advantages of the phase-field and peridynamics model with the linear elastic PDE model through the global-local method, by using the expensive models only locally where required. Until then, we only consider linear elastic fracture mechanics, on which we provide more details in the next section.

2.2 LINEAR ELASTIC FRACTURE MECHANICS

In linear elasticity, we consider only isotropic material in this thesis. We are thus interested in solving the PDE model (2.6) with (2.5) as the linear constitutive relation and (2.3) as the linear strain. However, we usually use Poisson's ratio ν and Young's modulus E to define Hooke's tensor, i. e. the material properties, where we have the relations

$$G = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{2.18}$$

Before we get into the specifics of linear elasticity solutions around cracks, we introduce the weak formulation of (2.6) for later reference. The weak form of linear elasticity is the actual problem we are concerned with in the remainder of the thesis. We state it in $H^1_0(\Omega)$ and assume that non-zero Dirichlet boundary data is treated separately in the discretization, as described in Section 2.3.4. Multiplying (2.6) by a test function $v \in H^1_0(\Omega)$ and using Gauss's theorem we arrive at the following problem [13]: find $u \in H^1_0(\Omega)$ such that

$$\int_\Omega \sigma(u) : \varepsilon(v)\,dx = \int_{\Gamma^N} \bar{t} v\,ds + \int_\Omega b\,v\,dx \tag{2.19}$$

for all test functions $v \in H^1_0(\Omega)$. Trial and test functions employed in our PUM are given in Section 2.3.2. Regarding boundary conditions, we refer to fixing the displacement u on Γ^D as setting Dirichlet boundary conditions. We refer to setting a traction $\bar{t}$, i. e. to applying a force on the domain Ω, as setting Neumann boundary conditions. For problem (2.6) to be well posed, we need Γ^D to have positive (d-1)-dimensional Lebesgue measure. We thus have to set at least some Dirichlet boundary conditions or exclude rigid body motions separately.

In the following, we focus on the arising singularities around crack fronts in linear elasticity and definition and extraction of so-called stress intensity factors. The mathematical analysis here is limited to simple crack geometries in three space dimensions. We discuss further challenges of complex crack geometries in Chapter 3.

2.2.1 *Crack Singularity*

In this section, we are concerned with the shape of solutions of (2.6) around crack tips in two or crack fronts in three space dimensions. Note that a real cracked body will have a region of inelastic deformation around the crack tip due to stress concentration there. Hence, elastic analysis is of practical utility only if we assume the region of inelastic deformation to be small compared to other characteristic lengths of the problem at hand. This section closely follows [137].

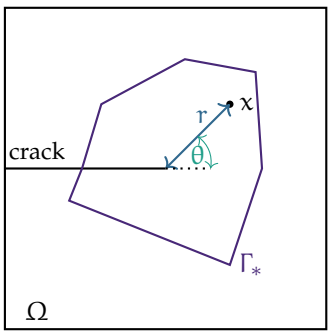


Figure 2.5: Two-dimensional crack problem with polar coordinate system (r, θ) with respect to crack tip and sketched extraction path Γ_*.

On a domain $\Omega \subset \mathbb{R}^2$ consider a two-dimensional crack problem in polar coordinates (r, θ) with respect to the crack tip as sketched in Figure 2.5. The

crack is parallel to the x_1 coordinate and we assume that the crack is traction free and we ignore rigid body motions. Consider the sequence $\{\lambda_i\}_{i\in\mathbb{N}}$ of real numbers given by

$$\lambda_i = \frac{1}{2}, \frac{3}{2}, 2, \frac{5}{2}, \dots \tag{2.20}$$

With

$$\Lambda_i := \frac{\lambda_i - 1}{\lambda_i + 1} \quad \text{and} \quad Q_i := \frac{-\Lambda_i \sin((\lambda_i - 1)\pi)}{\sin((\lambda_i + 1)\pi)} \tag{2.21}$$

we define the functions $\Psi^I_{\lambda_i}, \Psi^{II}_{\lambda_i} : \mathbb{R} \mapsto \mathbb{R}^2$ in terms of the angle θ by

$$\Psi^I_{\lambda_i}(\theta) := \begin{pmatrix} (k - Q_i(\lambda_i + 1))\cos(\lambda_i\theta) - \lambda_i \cos((\lambda_i - 2)\theta) \\ \\ (k + Q_i(\lambda_i + 1))\sin(\lambda_i\theta) + \lambda_i \sin((\lambda_i - 2)\theta) \end{pmatrix}, \tag{2.22}$$

where the Kolosov constant $k = 3 - 4\nu$ is a material parameter depending on Poisson's ratio ν and

$$\Psi^{II}_{\lambda_i}(\theta) := \begin{pmatrix} (k - Q_i(\lambda_i + 1))\sin(\lambda_i\theta) - \lambda_i \sin((\lambda_i - 2)\theta) \\ \\ -(k + Q_i(\lambda_i + 1))\cos(\lambda_i\theta) - \lambda_i \cos((\lambda_i - 2)\theta) \end{pmatrix}. \tag{2.23}$$

Sufficiently close to the crack, any solution u^* of linear elasticity can then be written [137] as

$$u^*(r, \theta) = \sum_{i=1}^{\infty} \frac{A^I_i}{2G} r^{\lambda_i} \Psi^I_{\lambda_i}(\theta) + \sum_{i=1}^{\infty} \frac{A^{II}_i}{2G} r^{\lambda_i} \Psi^{II}_{\lambda_i}(\theta), \tag{2.24}$$

where $G = E/2(1+\nu)$ is the shear modulus. Both infinite series converge absolutely for $r < r_0$ for some $0 < r_0$. Note that the series actually also converge for $-\lambda_i$ and the resulting function u^* is a valid solution to the crack problem, but with infinite strain energy. The first series describes mode I, the second mode II crack opening. The opening modes are visualized in Figure 2.3.

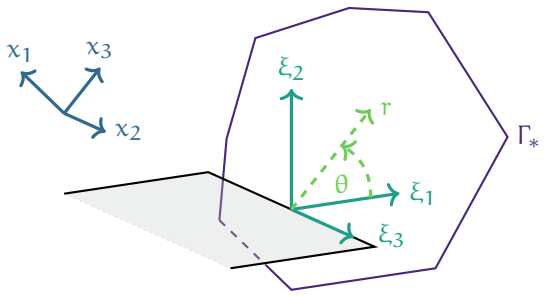


Figure 2.6: Three-dimensional crack problem with global coordinates (x_1, x_2, x_3), local coordinates (ξ_1, ξ_2, ξ_3) at a point on the crack front and corresponding cylindrical coordinates (r, θ, ξ_3). Sketched extraction path Γ_*.

In three dimensions, consider a single through the thickness crack with straight crack front, as depicted in Figure 2.6. Again, we assume that the crack is traction free and ignore rigid body motions. For the analysis we further assume

global (x_1, x_2, x_3) and local coordinates (ξ_1, ξ_2, ξ_3) to coincide, such that the crack lies in the (x_1, x_3)-plane. The general case then requires additional coordinate transformations. While we do not have a closed form solution like (2.24) in three space dimensions, the following statements about the displacement field u^* around the crack front are possible [61].

Considering only mode I and II crack opening first, that is $u_1^* = u_1^*(r, \theta)$, $u_2^* = u_2^*(r, \theta)$ and $u_3^* = 0$, we are essentially back in the two-dimensional case. The only singular contributions to the displacement field u^* around the crack front are

$$\frac{A_1^{\mathrm{I}}\sqrt{r}}{2G}\psi_{\lambda_1}^{\mathrm{I}} \quad \text{and} \quad \frac{A_1^{\mathrm{II}}\sqrt{r}}{2G}\psi_{\lambda_1}^{\mathrm{II}} \tag{2.25}$$

extended by zero in the third coordinate.

In the mode III case, with $u_1^* = u_2^* = 0$ and $u_3^* = u_3^*(r, \theta)$, the only singular contribution is given by

$$u_3^* = \frac{2A^{\mathrm{III}}\sqrt{r}}{G}\sin\left(\frac{\theta}{2}\right). \tag{2.26}$$

Hence we define

$$\psi_1^{\mathrm{III}}(\theta) \coloneqq \begin{pmatrix} 0 & 0 & \sin\left(\frac{\theta}{2}\right) \end{pmatrix}^{\mathsf{T}}. \tag{2.27}$$

2.2.2 *Stress Intensity Factors*

The generalized stress intensity factors A_i^{I}, A_i^{II} and A^{III} are unknown a priori and their respective displacement functions

$$u_{\mathrm{I}} \coloneqq \frac{\sqrt{r}}{2G}\psi_{\lambda_1}^{\mathrm{I}}(\theta), \quad u_{\mathrm{II}} \coloneqq \frac{\sqrt{r}}{2G}\psi_{\lambda_1}^{\mathrm{II}}(\theta) \quad \text{and} \quad u_{\mathrm{III}} \coloneqq \frac{2\sqrt{r}}{G}\psi_1^{\mathrm{III}}(\theta), \tag{2.28}$$

that depend only on the polar part r, θ of the coordinates around the crack front, have a singularity in their derivative due to the exponent on the radius. Thus, in a region close enough to the crack front but still outside the process zone, compare Figure 2.2, the functions (2.28) capture the material response to the applied load. In practice we are thus interested in computing the corresponding coefficients A_1^{I}, A_1^{II} and A^{III}, which are related to the stress intensity factors K_{I} (mode I), K_{II} (mode II) and K_{III} (mode III) of linear elastic fracture mechanics by

$$K_{\mathrm{I}} \coloneqq A_1^{\mathrm{I}}\sqrt{2\pi}, \quad K_{\mathrm{II}} \coloneqq A_1^{\mathrm{II}}\sqrt{2\pi} \quad \text{and} \quad K_{\mathrm{III}} = A^{\mathrm{III}}\sqrt{2\pi}. \tag{2.29}$$

In the following, we present two methods to extract these factors from a computed numerical solution. Note that extraction here is always at a point on the crack front, which we still assume to be straight and flat. As already noted earlier, we discuss challenges of complex crack and crack front geometries in Chapter 3.

2.2.2.1 *Contour Integral Method*

One method to extract the coefficients A_1^{I}, A_1^{II} and A^{III} from a displacement function is the CIM [30, 44, 137]. Here, extraction is performed by integration on a closed, counterclockwise path Γ_* in the (ξ_1, ξ_2)-plane perpendicular to the

crack fronts tangent ξ_3 around a point on the crack front, compare Figure 2.6. On that path and with functions $u, v : \mathbb{R}^3 \mapsto \mathbb{R}^3$, the integrals are given by

$$I_{\Gamma_*}(u, v) \coloneqq \int_{\Gamma_*} \mathcal{T}(u) \cdot v \, ds - \int_{\Gamma_*} \mathcal{T}(v) \cdot u \, ds, \tag{2.30}$$

where $\mathcal{T}$ is the traction vector on Γ_* in the extraction plane and in normal direction away from the point on the crack front. Note that the integral I_{Γ_*} is independent of the specific path Γ_*. We define the so-called extraction functions by

$$\begin{aligned} v_I \coloneqq \frac{1}{2G\sqrt{r}} \Psi^I_{-\lambda_1}(\theta), \quad v_{II} \quad &\coloneqq \frac{1}{2G\sqrt{r}} \Psi^{II}_{-\lambda_1}(\theta) \\ \text{and} \quad v_{III} &\coloneqq \frac{2}{G\sqrt{r}} \Psi^{III}_1(\theta), \end{aligned} \tag{2.31}$$

which in two space dimension correspond to terms of (2.24) with negative λ_i. Note that again we implicitly assume both displacement and extraction functions defined in the context of the two-dimensional analysis to be extended by zero in the third component. With constants

$$\begin{aligned} C_I \coloneqq I_{\Gamma_*}(u_I, v_I), \quad C_{II} \quad &\coloneqq I_{\Gamma_*}(u_{II}, v_{II}) \\ \text{and} \quad C_{III} &\coloneqq I_{\Gamma_*}(u_{III}, v_{III}) \end{aligned} \tag{2.32}$$

computed from the functions (2.28) and (2.31), the generalized stress intensity factors can be extracted from the crack front expansion (2.24) in two space dimensions by

$$A^I_1 = I_{\Gamma_*}\left(u^*, \frac{1}{C_I} v_I\right) \quad \text{and} \quad A^{II}_1 \quad = I_{\Gamma_*}\left(u^*, \frac{1}{C_{II}} v_{II}\right) \tag{2.33}$$

and in three space dimensions we have

$$A^{III} = I_{\Gamma_*}\left(u_{III}, \frac{1}{C_{III}} v_{III}\right). \tag{2.34}$$

Since the true solution to our problem is unknown, we extract from a computed numerical approximation u_{PU} by

$$\begin{aligned} K_I = \sqrt{2\pi}\, I_{\Gamma_*}\left(u_{PU}, \frac{1}{C_I} v_I\right), \quad K_{II} \quad &= \sqrt{2\pi}\, I_{\Gamma_*}\left(u_{PU}, \frac{1}{C_{II}} v_{II}\right) \\ \text{and} \quad K_{III} = \sqrt{2\pi}\, I_{\Gamma_*}&\left(u_{PU}, \frac{1}{C_{III}} v_{III}\right). \end{aligned} \tag{2.35}$$

Computing the integrals (2.35) numerically via the CIM is discussed in Section 3.2.1 in detail.

2.2.2.2 *Displacement Correlation Method*

A simple alternative to the CIM is the DCM [42, 44, 85], which estimates the stress intensity factors by the jump in displacement across the crack. While potentially less accurate than integral based methods when extracting from an analytic crack displacement function, it is computationally cheap to evaluate

and more robust to errors in the computed displacement as it does not evaluate the derivative.

Let $u(r, \theta, \xi_3)$ be the displacement in cylindrical coordinates around a straight crack front as in Figure 2.6. We define the jump across the crack in the i-th local coordinate (ξ_1, ξ_2, ξ_3) by

$$[\![u_i(r)]\!] \coloneqq u_i(r, \pi, 0) - u_i(r, -\pi, 0). \tag{2.36}$$

Then the approximation

$$K_{I,r} \coloneqq \sqrt{\frac{2\pi}{r}} \frac{G}{k+1} [\![u_2(r)]\!] \tag{2.37}$$

of the real stress intensity factor K_I is of order $\mathcal{O}(r)$ if we extract directly from the crack front displacement functions (2.28). Similarly we define

$$K_{II,r} \coloneqq \sqrt{\frac{2\pi}{r}} \frac{G}{k+1} [\![u_1(r)]\!] \tag{2.38}$$

and

$$K_{III,r} \coloneqq \sqrt{\frac{2\pi}{r}} \frac{G}{4} [\![u_3(r)]\!]. \tag{2.39}$$

The approximation can be further improved by using two extraction radii $r_a < r_b$ and Richardson extrapolation

$$\widehat{K}_i \coloneqq \frac{r_b}{r_b - r_a} \left(K_{i,r_a} - \frac{r_a}{r_b} K_{i,r_b} \right) \tag{2.40}$$

for $i \in [I, II, III]$, making $\widehat{K}_i$ an order $\mathcal{O}(r^2)$ approximation of the true stress intensity factors if we extract from the crack front expansion. Extracting from a computed displacement introduces additional error not easily captured in terms of r, though. In practice, we thus use several pairs of extraction radii and average the results to account for the displacement error.

2.2.3 *Crack Growth Criteria*

By the application of a crack growth law based on the stress intensity factors, a decision can be made whether and in which direction a crack grows. Unfortunately, there is not one definitive crack growth law in linear elastic fracture mechanics, but several exist in the literature. Such a law then estimates two growth angles, as sketched in Figure 2.7.

The comparison in [109] suggests that Schöllmann's criterion and the criterion of Richards are able to capture mixed-mode crack growth effects in three space dimensions. Based on the decision in [104] we chose Schöllmann's criterion introduced in [113]. Therein, it is argued that in a lot of cases, cracks in experiments grow radially from the crack front into the direction that is perpendicular to the maximum principal stress σ_1' given by

$$\sigma_1' \coloneqq \frac{\sigma_\alpha + \sigma_z}{2} + \frac{1}{2} \sqrt{(\sigma_\alpha - \sigma_z)^2 + 4w_{\alpha z}^2}, \tag{2.41}$$

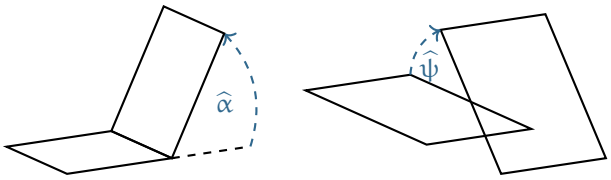


Figure 2.7: Crack growth directions for pure and mixed fracture modes in three space dimensions. Mode II introduces a deflection angle $\widehat{\alpha}$ and mode III a twisting angle $\widehat{\psi}$.

with

$$\begin{aligned}\sigma_\alpha =& \frac{K_I}{4\sqrt{2\pi r}}\left(3\cos\left(\frac{\alpha}{2}\right)+\cos\left(\frac{3\alpha}{2}\right)\right)\\ &-\frac{K_{II}}{4\sqrt{2\pi r}}\left(3\sin(\frac{\alpha}{2})+3\sin(\frac{3\alpha}{2})\right)\end{aligned} \tag{2.42}$$

and

$$\begin{aligned}w_{\alpha z} :=& \frac{K_I}{4\sqrt{2\pi r}}\left(\sin(\frac{\alpha}{2})+\sin(\frac{3\alpha}{2})\right)\\ &+\frac{K_{II}}{4\sqrt{2\pi r}}\left(\cos\left(\frac{\alpha}{2}\right)+3\cos\left(\frac{3\alpha}{2}\right)\right).\end{aligned} \tag{2.43}$$

Setting $\sigma_z = 0$ for plane stress condition and assuming that the crack growth direction is perpendicular to σ_1', we can find the crack deflection angle $\widehat{\alpha}$ at

$$\left.\frac{\partial\sigma_1'}{\partial\alpha}\right|_{\alpha=\widehat{\alpha}} = 0 \qquad \text{and} \qquad \left.\frac{\partial^2\sigma_1'}{\partial\alpha^2}\right|_{\alpha=\widehat{\alpha}} < 0. \tag{2.44}$$

The angle $\widehat{\alpha}$ fulfilling (2.44) can then be found [113] by solving

$$\begin{aligned}0 =& -6K_I\tan\left(\frac{\widehat{\alpha}}{2}\right)-K_{II}\left(6-12\tan^2\left(\frac{\widehat{\alpha}}{2}\right)\right)\\ &+\left\{\left[4K_I-12K_{II}\tan\left(\frac{\widehat{\alpha}}{2}\right)\right]\right.\\ &\quad\cdot\left[-6K_I\tan\left(\frac{\widehat{\alpha}}{2}\right)-K_{II}\left(6-12\tan^2\left(\frac{\widehat{\alpha}}{2}\right)\right)\right]\\ &\quad\left.-32K_{III}^2\tan\left(\frac{\widehat{\alpha}}{2}\right)\left(1+\tan^2\left(\frac{\widehat{\alpha}}{2}\right)\right)^2\right\}\\ &\cdot\left\{\left[4K_I-12K_{II}\tan\left(\frac{\widehat{\alpha}}{2}\right)\right]^2+64K_{III}^2\left(1+\tan^2\left(\frac{\widehat{\alpha}}{2}\right)\right)^2\right\}^{-1/2},\end{aligned} \tag{2.45}$$

which in practice can be done efficiently by a numerical root finding algorithm, such as Brent's method [15], as we know that the angle $\widehat{\alpha}$ is bounded by $\pm\pi/2$, or actually $\pm 70.5°$ for pure mode II loading. The twisting angle $\widehat{\psi}$ can then be calculated as

$$\widehat{\psi} = \frac{1}{2}\arctan\left(\frac{2w_{\alpha z}(\widehat{\alpha})}{\sigma_\alpha(\widehat{\alpha})-\sigma_z(\widehat{\alpha})}\right), \tag{2.46}$$

with

$$\sigma_z(\alpha) = \frac{8\nu}{4\sqrt{2\pi r}}\left(K_I \cos\left(\frac{\alpha}{2}\right) - K_{II}\cos\left(\frac{\alpha}{2}\right)\right) \tag{2.47}$$

and the Poisson's ratio ν. To decide whether a crack grows at all, [113] gives the following formula

$$\begin{aligned} K_\nu = \frac{1}{2}\cos\left(\frac{\widehat{\alpha}}{2}\right)\Bigg[K_I \cos^2\left(\frac{\widehat{\alpha}}{2}\right) - \frac{3}{2}K_{II}\sin(\widehat{\alpha}) \\ + \sqrt{\left(K_I \cos^2\left(\frac{\widehat{\alpha}}{2}\right) - \frac{3}{2}K_{II}\sin(\widehat{\alpha})\right)^2 + 4K_{III}^2}\Bigg] \end{aligned} \tag{2.48}$$

to compute a comparative value K_ν to compare against the critical material parameter K_c.

2.3 PARTITION OF UNITY METHOD

Partition of unity methods are discretizations of the Galerkin method for PDEs, introduced in [83] to overcome limitations in the choice of basis functions in classical FEMs. The key concept of partition of unity methods is a compactly supported PU that covers the computational domain Ω. To each PU function local approximation spaces are attached, the sum over which yields a global, finite-dimensional approximation space. Thus, and in contrast to FEMs, arbitrary basis functions can be incorporated where needed. By using only few problem specific basis functions, partition of unity methods can thereby achieve comparable or better approximation properties with less degrees of freedom, especially in problems that involve singularities or discontinuities.

We can distinguish two classes of partition of unity methods: Meshfree methods, with different ways of defining the underlying PU, and methods based on finite elements. The latter utilize the fact that classical linear finite element hat functions already form a compactly supported PU that covers the mesh and thus the computational domain. They originate from the partition of unity finite element method by Melenk and Babuška [83] and are widely used today as either the GFEM or the XFEM, introduced by Duarte, Babuška and Oden in [22] or in [86] by Moës, Dolbow and Belytschko, respectively. In first meshfree partition of unity methods, the PU was constructed from point sets based on moving least squares approaches, as for example done in hp-clouds [24]. This idea evolved into the particular PUM by Schweitzer introduced in [114, 115], which is used throughout this thesis. Therein an automatic construction of a PU starting from a bounding box of the domain Ω is stated that, in contrast to finite element based partition of unity methods, allows to guarantee global stability of the basis functions.

For context, we assume to solve the abstract weak formulation of a PDE stated in V, which is

$$\text{find } u \in V: \quad a(u,v) = l(v) \quad \forall v \in V. \tag{2.49}$$

In the Galerkin method we reduce that equation to a finite dimensional subspace $V_n \subset V$ spanned by basis functions $\{v_i\}_{i \leqslant n}$. With the expansion

$$\sum_{j \leqslant n} \tilde{u}_j v_j = u \in V_n, \tag{2.50}$$

we arrive at the equation

$$\sum_{j \leqslant n} \tilde{u}_j \, a(v_j, v_i) = l(v_i) \qquad \forall i \leqslant n, \tag{2.51}$$

which can be rewritten as the linear system of equations

$$A\tilde{u} = b \quad \text{with} \quad A_{ij} = a(v_j, v_i) \quad \text{and} \quad b_i = l(v_i), \tag{2.52}$$

where $\tilde{u} \in \mathbb{R}^n$ is the coefficient vector, $A \in \mathbb{R}^{n \times n}$ the stiffness matrix and $b \in \mathbb{R}^n$ the right hand side vector.

A partition of unity method has two key ingredients: a partition of unity $\{\varphi_i \,|\, i \leqslant N\}$ and local approximation spaces $V_i = \operatorname{span}\{\vartheta_i^n \,|\, n \leqslant n_i\}$, with which the finite dimensional approximation space $V_n = V^{PU}$ is defined by

$$V^{PU} \coloneqq \sum_{i=1}^{N} \varphi_i V_i = \operatorname{span}\{\varphi_i \vartheta_i^n \,|\, i \leqslant N,\ n \leqslant n_i\}. \tag{2.53}$$

In the following, we first state properties of an abstract partition of unity method more precisely, introduce the PU and local basis functions chosen in this thesis and present how we can transform them into a stable basis. We further introduce a way of incorporating essential boundary conditions in this meshfree discretization and detail the construction of integration cells suitable for the evaluation of our product basis functions. The latter is necessary for the integrals that need to be evaluated during the assembly of the linear system (2.52). The presentation of the PUM is reduced to the level of detail required for this thesis, with special emphasis on the way it is actually implemented in the software framework PUMA. We refer more interested readers to [115].

2.3.1 *Abstract PUM and Cover*

We start with a specialized definition of a partition of unity for this thesis.

Definition 2.1 (Partition of Unity)**.** *Let $\Omega \subset \mathbb{R}^d$ be an open set, $M < N \in \mathbb{N}$ and C_∇ and C_∞ be positive constants. Let $\{\varphi_i \,|\, i \leqslant N\}$ be Lipschitz functions and a partition of unity in the usual sense*

$$0 \leqslant \varphi_i(x) \leqslant 1 \quad \textit{and} \quad \sum_{i=1}^{N} \varphi_i \equiv 1 \textit{ on } \overline{\Omega}, \tag{2.54}$$

with an upper bound on the covering index λ_{C_Ω}. Specifically, we require that for all $x \in \Omega$

$$\lambda_{C_\Omega}(x) \coloneqq \operatorname{card}\big(\{\varphi_i \,|\, x \in \omega_i\}\big) \leqslant M \ll N, \tag{2.55}$$

where $\omega_i \coloneqq \operatorname{supp}(\varphi_i)^\circ$ *are the so-called patches. Further, assume that the patches* ω_i *are Lipschitz domains and let*

$$\|\varphi_i\|_{L^\infty(\mathbb{R}^d)} \leqslant C_\infty \quad \textit{and} \quad \|\nabla\varphi_i\|_{L^\infty(\mathbb{R}^d)} \leqslant \frac{C_\nabla}{\operatorname{diam}(\omega_i)}. \tag{2.56}$$

For this monograph, PU *refers to a set of functions* $\{\varphi_i \,|\, i \leqslant N\}$ *with these properties. The* PU *is said to be of degree* $k \in \mathbb{N}_0$ *if*

$$\varphi_i \in C^k\left(\mathbb{R}^d\right) \quad \textit{and} \quad \left\|\nabla^k\varphi_i\right\|_{L^\infty(\mathbb{R}^d)} \leqslant \frac{C_{\nabla^k}}{\operatorname{diam}(\omega_i)} \tag{2.57}$$

for all $i \leqslant N$.

Definition 2.2 (Cover). *Let* $\{\varphi_i\}$ *be a* PU *as in Definition 2.1. The set of patches* $\omega_i \coloneqq \operatorname{supp}(\varphi_i)^\circ$ *is referred to as a cover* $C_\Omega \coloneqq \{\omega_i \,|\, i \leqslant N\}$ *of the domain* Ω.

Definition 2.3 (PUM-Space). *Let* $V_i = \operatorname{span}\{\vartheta_i^n \,|\, n \leqslant n_i\} \subset H^1(\omega_i)$ *be function spaces of dimension* n_i *subordinate to a* PU $\{\varphi_i\}$ *as in Definition 2.1. Then we define the* PUM*-space* V^{PU} *by*

$$\begin{aligned} V^{PU} \coloneqq \sum_{i=1}^{N} \varphi_i V_i = \sum_{i=1}^{N} \varphi_i \operatorname{span}\left\{\vartheta_i^n \,\middle|\, n \leqslant n_i\right\} \\ = \operatorname{span}\left\{\varphi_i\vartheta_i^n \,\middle|\, i \leqslant N,\ n \leqslant n_i\right\}. \end{aligned} \tag{2.58}$$

Note that the local spaces V_i *are completely independent of each other, both in dimension* n_i *and incorporated functions* ϑ_i^n. *Also, all basis functions are product functions, since the local basis functions* ϑ_i^n *are multiplied with the* PU *functions* φ_i.

For PUM-spaces as in Definition 2.3 we have have following error estimate [4, 115].

Theorem 2.1 (Approximation Property). *Let* $\Omega \subset \mathbb{R}^d$ *be a Lipschitz domain and* $\{\varphi_i\}$ *be a* PU *as in Definition 2.1. We assume the covering index of the cover* C_Ω *to be uniformly bounded,* $\lambda_{C_\Omega}(x) \leqslant M \in \mathbb{N}$ *for all* $x \in \Omega$. *Let* $u^* \in H^1(\Omega)$ *and assume we have approximation spaces* $V_i = \operatorname{span}\{\vartheta_i^n \,|\, n \leqslant n_i\} \subset H^1(\omega_i)$, *such that we can find functions* $u_i \in V_i$ *satisfying*

$$\|u^* - u_i\|_{L^2(\Omega\cap\omega_i)} \leqslant \hat{\varepsilon}_i \quad \textit{and} \quad \|\nabla(u^* - u_i)\|_{L^2(\Omega\cap\omega_i)} \leqslant \tilde{\varepsilon}_i \tag{2.59}$$

for all $i = 1, \ldots, N$. *That is, on each patch* $\Omega \cap \omega_i$ *we can approximate the target function* u^*. *Then the* PUM *function*

$$u_{PU} \coloneqq \sum_{i=1}^{N} \varphi_i u_i \in V^{PU} \subset H^1(\Omega) \tag{2.60}$$

approximates u^* *in* $H^1(\Omega)$. *Specifically, it satisfies the estimates*

$$\begin{aligned} \|u^* - u_{PU}\|_{L^2(\Omega)} &\leqslant \sqrt{M} C_\infty \left(\sum_{i=1}^{N} \hat{\varepsilon}_i^2 \right)^{\frac{1}{2}}, \\ \|\nabla(u^* - u_{PU})\|_{L^2(\Omega)} &\leqslant \sqrt{2M} \left(\sum_{i=1}^{N} \left(\frac{C_\nabla}{\operatorname{diam}(\omega_i)} \right)^2 \tilde{\varepsilon}_i^2 + C_\infty^2 \hat{\varepsilon}_i^2 \right)^{\frac{1}{2}}, \end{aligned} \tag{2.61}$$

where C_∇ *and* C_∞ *are the constants from Definition 2.1.*

If the covering index λ_{C_Ω} is independent of N, the estimates (2.61) show that the global error is of the same order as the local errors. With the cover construction presented in Section 2.3.5 we always have $\lambda_{C_\Omega} \leqslant 8$ for uniformly refined covers. Note that under the additional assumption of a nonnegative PU, the $\sqrt{M}$ factor can be dropped from the L^2-estimate [118].

If we further assume that $\operatorname{diam}(\omega_i) \approx h \; \forall i \leqslant N$ for all patches and that all local approximation spaces V_i contain polynomials of degree p, then for $u^* \in H^k(\Omega)$ with $1 \leqslant k$ and $p \leqslant k-1$ we have

$$\begin{aligned} \|u^* - u_i\|_{L^2(\Omega \cap \omega_i)} &\leqslant C h^{p+1} \|u^*\|_{H^k(\Omega \cap \omega_i)} \coloneqq \hat{\varepsilon}_i, \\ \|\nabla(u^* - u_i)\|_{L^2(\Omega \cap \omega_i)} &\leqslant C h^{p} \|u^*\|_{H^k(\Omega \cap \omega_i)} \coloneqq \tilde{\varepsilon}_i. \end{aligned} \tag{2.62}$$

In this case, the estimates (2.61) have the form of errors estimates for classical finite elements with uniform h-refinement [4]:

$$\begin{aligned} \|u^* - u_{PU}\|_{L^2(\Omega)} &\leqslant \sqrt{M} C_\infty C h^{p+1} \|u^*\|_{H^k(\Omega)}, \\ \|\nabla(u^* - u_{PU})\|_{L^2(\Omega)} &\leqslant \sqrt{2M(C_\infty + C_\nabla)} C h^{p} \|u^*\|_{H^k(\Omega)}. \end{aligned} \tag{2.63}$$

Note, however, that the estimates do not state that the representation of u_{PU} is unique in the PUM-space. That is, the shape functions $\varphi_i \vartheta_i^n$ could be linearly dependent. For example, in one space dimension, the PU functions

$$\varphi_1(x) \coloneqq 1 - x \quad \text{and} \quad \varphi_2(x) \coloneqq x \tag{2.64}$$

on the domain $\Omega \coloneqq (0,1)$ with the polynomial basis functions

$$\vartheta_i^1(x) \coloneqq 1 \quad \text{and} \quad \vartheta_i^2(x) \coloneqq x \tag{2.65}$$

generates the PUM-space

$$V^{PU} = \operatorname{span}\{(1-x), (1-x)x, x, x^2\} \tag{2.66}$$

containing all quadratic polynomials on $(0,1)$, but with four basis functions instead of three. Note that (2.64) is the coarsest possible GFEM partition of unity on $(0,1)$, where the PU functions fully overlap. To eliminate linear dependencies we thus require an additional property of the cover.

Definition 2.4 (Flat-Top Property). *Let* $\{\varphi_i\}$ *be a* PU *as in Definition 2.1. We define the sub-patches*

$$\omega_{FT,i} \coloneqq \{x \in \Omega \,\big|\, \varphi_i(x) = 1\} \subset \omega_i. \tag{2.67}$$

The PU has the flat-top property, if there exists a positive constant C_{FT} *such that*

$$\mu(\omega_i) \leqslant C_{FT}\mu\big(\omega_{FT,i}\big) \qquad \forall i \leqslant N, \tag{2.68}$$

where μ *is the Lebesgue-measure. We have* $C_\infty = 1$ *for a PU satisfying* (2.68).

Given linearly independent local approximation spaces V_i, the flat-top property is a sufficient condition for linear independence of the global PUM-space V^{PU}, that is, we have

$$\sum_{i=1}^{N} \varphi_i \sum_{n=1}^{\dim(V_i)} u_i^n \vartheta_i^n \equiv 0 \iff \sum_{n=1}^{\dim(V_i)} u_i^n \vartheta_i^n \equiv 0 \qquad \forall i \leqslant N \tag{2.69}$$

for a PU with the flat-top property. The PUM-space V^{PU} then is the direct sum of the local approximation spaces V_i

$$V^{PU} = \bigoplus_{i=1}^{N} \varphi_i V_i. \tag{2.70}$$

Note that on general geometries, i. e. not axis aligned boxes, we actually require the strict flat-top property introduced in [149] for guaranteed global linear independence. There, (2.68) is replaced by

$$\mu(\omega_i) \leqslant C_{FT}\mu\big(\omega_{FT,i} \cap \Omega\big) \qquad \forall i \leqslant N \tag{2.71}$$

to account for the possibly small intersection of a patch with the computational domain. An improved cover construction to account for this issue is also introduced in [149], but not necessary for this thesis, as we include geometric difficulties only through crack surfaces in our numerical examples and compute on otherwise simple domains. For the PUM employed in this thesis, we arrive at the following definition of an admissible cover.

Definition 2.5 (Admissible Cover)**.** *Let* $\Omega \subset \mathbb{R}^d$ *be an open set. Let* $\omega_i \subset \mathbb{R}^d$ *be open sets with* $\omega_i \cap \Omega \neq \emptyset$ *for* $i \leqslant N$*. The collection* $C_\Omega := \{\omega_i \,|\, i \leqslant N\}$ *is called an admissible cover of* Ω *and the sets* ω_i *are denoted admissible cover patches if the following conditions are satisfied:*

- *Global covering:*

$$\overline{\Omega} \subset \bigcup_{i=1}^{N} \omega_i. \tag{2.72}$$

- *Minimal overlap: There exists a constant* $0 < C_{FT}$ *such that*

$$\mu(\omega_i) \leqslant C_{FT}\mu\big(\{x \in \omega_i \,\big|\, \lambda_{C_\Omega}(x) = 1\}\big) \qquad \forall i \leqslant N. \tag{2.73}$$

- *Bounded overlap: There exists* $0 < M \in \mathbb{N}$ *such that for all* $x \in \Omega$ *we have*

$$\lambda_{C_\Omega}(x) = \operatorname{card}\big(\{i \,\big|\, x \in \omega_i\}\big) \leqslant M \ll N, \tag{2.74}$$

- *Sufficient overlap: There exists* $0 < C_S$ *such that for all* $x \in \Omega$ *there exists at least one cover patch* ω_i *such that* $x \in \omega_i$ *and*

$$C_S \operatorname{diam}(\omega_i) \leqslant \operatorname{dist}(x, \partial\omega_i). \tag{2.75}$$

- *Comparability of neighboring patches: A subset*

$$C_i \coloneqq \{\omega_j \in C_\Omega \mid \omega_j \cap \omega_i \neq \emptyset\} \subset C_\Omega \tag{2.76}$$

is called a local neighborhood or local cover of a particular cover patch $\omega_i \in C_\Omega$. *There exists a constant* $0 < C_\mathcal{N} \leqslant 1$ *such that for all intersecting patches* $\omega_i, \omega_j \in C_\Omega$ *with* $\omega_i \cap \omega_j \neq \emptyset$ *we have*

$$\operatorname{diam}(\omega_j) \leqslant \operatorname{diam}(\omega_i) \quad \Longrightarrow \quad \frac{\operatorname{diam}(\omega_i)}{\operatorname{diam}(\omega_j)} \leqslant C_\mathcal{N}. \tag{2.77}$$

For a given cover with the flat-top property and some overlap, we can of course often find such constants, as there are only a finite number of patches. We are thus especially interested in an automatic cover construction that generates an admissible cover for given constants C_{FT} and C_S and for a given domain. Such a construction is detailed in Section 2.3.5.

2.3.2 *PU and Local Basis Functions*

An admissible cover can be used to define the PU for our specific PUM using Shepard functions as follows. Given positive weight functions $0 < W_i(x)$ per patch for all $x \in \omega_i \setminus \partial\omega_i$ we define

$$\varphi_i \coloneqq \frac{W_i(x)}{\sum_{\omega_j \in C_j} W_j(x)}, \tag{2.78}$$

where C_j is the neighborhood of the patch ω_j as defined in (2.76). The regularity of the resulting PU is then determined by the smoothness of the employed weight functions. Note, however, that the construction (2.78) in general yields rational weight functions in the overlap of patches, which can require high integration orders.

From this point onwards, we assume patches to be d-dimensional rectangles

$$\omega_i = \bigotimes_{k=1}^{d} (c_{ik} - r_i, c_{ik} + r_i), \tag{2.79}$$

with c_i the center of the patch and r_i its radius. On those patches, we can define the weight functions as follows. Using affine transformations

$$T_{ik} : [c_{ik} - r_i, c_{ik} + r_i] \mapsto [-1, 1] \tag{2.80}$$

and a spline function $B : [-1, 1] \mapsto \mathbb{R}$, we define the weight function on a patch ω_i by

$$W_i(x) \coloneqq \prod_{k=1}^{d} B \circ T_{ik}(x_k). \tag{2.81}$$

The weight functions are thus tensor products of one dimensional spline functions, whose order can be trivially adjusted to the required regularity. One can use standard B-splines, but by default, we use so-called flat-top splines in PUMA, see Figure 2.8. Those have the benefit that the construction (2.78) yields non-rational weight functions in the overlap of patches on the same discretization level with the construction detailed in Section 2.3.5.

Lemma 2.2. *On an admissible cover as in Definition 2.5 with weight functions* (2.81)*, the set of functions defined by* (2.78) *is a valid* PU *with the flat-top property, i. e. it satisfies Definition 2.1 and Definition 2.4.*

Proof. See [115]. □

Following [115] we choose local approximation spaces V_i as the combination of a polynomials space $\mathcal{P}_i^{p_i} \coloneqq \operatorname{span}\{\psi_i^s\}$ of degree up to p_i and an enrichment space $\mathcal{E}_i \coloneqq \operatorname{span}\{\eta_i^t\}$ consisting of problem dependent functions, thus

$$V_i \coloneqq \mathcal{P}_i^{p_i} + \mathcal{E}_i = \operatorname{span}\{\psi_i^s, \eta_i^t\}. \tag{2.82}$$

As with the weight functions, we choose the polynomial functions to be tensor products of one dimensional polynomials throughout this thesis. Let $\mathcal{L}^q : [-1, 1] \mapsto \mathbb{R}$ be the Legendre polynomial of degree q, then the polynomial part of V_i is given by

$$\mathcal{P}_i^{p_i} \coloneqq \left\{ \prod_{k=1}^{d} \mathcal{L}^{q_k} \circ T_{ik}(x_k) \;\middle|\; \|(q_k)_{k=1}\|^d{}_1 \leqslant p_i \right\}. \tag{2.83}$$

Enrichment functions η_i^t can be added to local spaces individually. Note, however, that the resulting basis function is $\varphi_i \eta_i^t$ on a patch ω_i. This has the following implication. Consider an enrichment function η given globally, that is, defined on e. g. the whole domain Ω. Then its restriction to a patch ω_i is in the PUM-space V^{PU} only if it is applied to all local spaces in the neighborhood C_i of that patch. More precisely,

$$\eta|_{\omega_i} \in V^{PU}\big|_{\omega_i} \qquad \Leftrightarrow \qquad \eta|_{\omega_i} \in V_j(\omega_j)\big|_{\omega_i} \quad \forall \omega_j \in C_i. \tag{2.84}$$

Enriching only a single patch may thus not yield the expected result, depending on the size of the patch relative to the size of the area of influence of the feature the enrichment is trying to model. Appropriate enrichment radii for crack front enrichments are discussed in Section 3.1.3. Besides adding enrichments to a local approximation space, we can further apply an enrichment η_m multiplicatively to the polynomial part of that space, compare Figure 2.8 such that

$$V_i = \operatorname{span}\{\psi_i^s, \eta_m \psi_i^s, \eta_i^t\}, \tag{2.85}$$

which of course adds more degrees of freedom to that local approximation space. Using enrichments in either way may result in linear dependencies between

the polynomial part of the local approximation space and the enrichments, or between multiple enrichments themselves. In the PUM, however, this is easy to handle, as the flat-top property from Definition 2.4 allows us to construct a globally stable basis using only patch local operations.

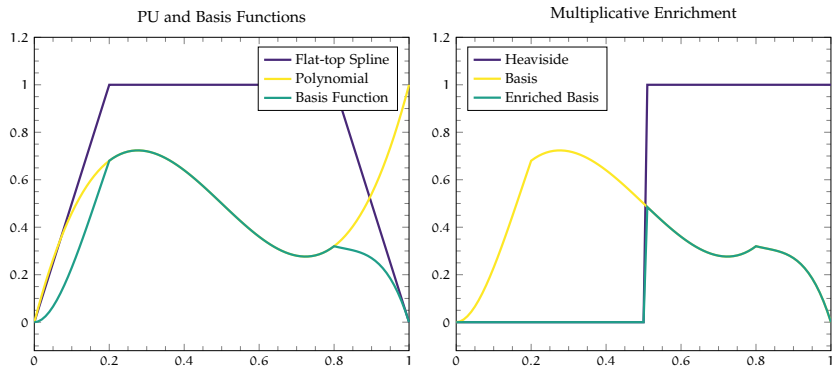


Figure 2.8: Flat-top spline, polynomial function and resulting product basis function on the left. Heaviside enrichment, basis function and their product on the right. The latter for example allows to have polynomial basis functions respecting the jump across a crack.

2.3.3 *Stable Basis*

As we saw in (2.69), the flat-top property guarantees global linear independence of basis functions given local linear independence on each patch ω_i. Here, we face two issues. First, a set of functions that is linear independent on the whole patch ω_i might be linear dependent on its intersection with the domain, especially if $\omega_i \cap \Omega$ is small. Second, enrichments can either be linear dependent to the polynomials employed on a patch, especially if we enrich a larger area around a singularity as suggested by (3.6), or be linear dependent themselves, if multiple enrichments are used on a patch. The latter may especially happen as enrichments are usually defined globally, where they might be linear independent, while still being linear dependent on a single patch. Selecting only linear independent enrichments a priori is impractical and if the enrichments themselves are results of previous simulations, as in the global-local method, even impossible. Furthermore, note that we are not only concerned with strict mathematical linear independence but with numerical independence and conditioning of the resulting linear system as well. To circumvent this limitation Schweitzer proposed [117] a transformation using patch local operations only to transform the original global set of generating functions $\{\vartheta_i^n\}$ into a stable basis $\{\widetilde{\vartheta}_i^n\}$. A similar technique for producing local stability has later been reintroduced [79] for the GFEM, while of course global stability cannot be guaranteed without the flat-top property.

The transformation is built as follows. Using the basis $\operatorname{span}\{\vartheta_i^n\} = V_i$ of the local approximation spaces we assemble the local mass matrices

$$(M_i)_{n,m} \coloneqq \int_{\omega_i \cap \Omega} \vartheta_i^m \vartheta_i^n \, dx \qquad M_i \in \mathbb{R}^{d_i \times d_i} \tag{2.86}$$

where $d_i = \dim(V_i)$ and compute their eigenvalue decomposition

$$M_i = Q_i^T D_i Q_i \qquad Q_i, D_i \in \mathbb{R}^{d_i \times d_i}. \tag{2.87}$$

Here, D_i is diagonal with $\lambda_s \coloneqq (D_i)_{s,s}$ the $s = 1, \dots, d_i$ eigenvalues and Q_i is orthogonal with rows of eigenvectors of M_i. We further assume that the eigenvalues in D_i are ordered with λ_1 being the largest eigenvalue. Again, we create a block-partitioning by

$$Q_i = \begin{pmatrix} \widetilde{Q}_i \\ K_i \end{pmatrix} \quad \text{and} \quad D_i = \begin{pmatrix} \widetilde{D}_i & 0 \\ 0 & \mathfrak{K}_i \end{pmatrix}, \tag{2.88}$$

where $\mathfrak{K}_i$ contains $\{\lambda_s \,|\, \lambda_s \leqslant \epsilon\lambda_1\}$ all small eigenvalues with respect to some user defined $\epsilon \ll 1$. As $0 < \epsilon\lambda_1 < (\widetilde{D}_i)_{n,n}$ we define the projection

$$S_i \coloneqq \widetilde{D}_i^{-1/2} \widetilde{Q}_i \tag{2.89}$$

which removes the near-null space of M_i, as we have

$$S_i M_i S_i^T = \widetilde{D}_i^{-1/2} \widetilde{Q}_i M_i \widetilde{Q}_i^T \widetilde{D}_i^{-1/2} = I_{\widetilde{d}_i} \tag{2.90}$$

with $\widetilde{d}_i \coloneqq \operatorname{card}\big(\{\lambda_s \,\big|\, \lambda_s > \epsilon\lambda_1\}\big)$ the dimension of the transformed space $\widetilde{Q}_i V_i$. The map

$$S_i : V_i = \operatorname{span}\{\vartheta_i^n\} \mapsto \operatorname{span}\{\widetilde{\vartheta}_i^n\} \approx V_i \tag{2.91}$$

thus transforms the generating set $\{\vartheta_i^n\}$ into the stable basis $\{\widetilde{\vartheta}_i^n\}$. From the local transformation we can define the block-diagonal operator S with blocks

$$S_{i,j} \coloneqq \begin{cases} S_i & \text{if } i = j \\ 0 & \text{else.} \end{cases} \tag{2.92}$$

Using this operator we can assemble the stiffness matrix A_ϑ and right hand side $\widehat{f}_\vartheta$ in the original basis $\{\vartheta_i^n\}$ and transform it into the stable basis $\{\widetilde{\vartheta}_i^n\}$ by

$$A_{\widetilde{\vartheta}} \coloneqq S A_\vartheta S^T \quad \text{and} \quad \widehat{f}_{\widetilde{\vartheta}} \coloneqq S \widehat{f}_\vartheta \tag{2.93}$$

and solve there. Should we need the original coefficient vector, for example for visualization, we can recover it by $\widehat{u}_\vartheta = S^T \widehat{u}_{\widetilde{\vartheta}}$.

The proposed algorithm requires local operations only and is thus easy to parallelize. In practice, the bilinear form in (2.86) can be replaced by the H^1 inner product or the bilinear form of the weak form of the PDE to be solved. While theoretically this has no effect in producing a stable basis, numerically it can make a difference. Also the cutoff parameter ϵ has to be chosen, which can affect stability and alter the generated approximation space.

Due to the lack of the flat-top property, the described stable transformation is not available in finite element based partition of unity methods such as the GFEM or XFEM. Consequently, other methods to improve the condition number of the linear system were proposed in that context, such as perturbation of the assembled stiffness matrix [47], application of a specialized preconditioner [5]

or localization of enrichments by means of a cutoff function [17]. A slightly more general technique is subtracting its polynomial approximation from an enrichment function as introduced in the context of the stable GFEM in [3, 47, 48]. This approach is less generic than the above transformation in that it does not account for dependencies between enrichments, especially for computed enrichments [81] where no a priori information about them is available. In contrast to our stable transformation, it should further be difficult to assess if an enrichment is actually spanned by the polynomial basis on some patch in the stable GFEM. There, the enrichment is only evaluated on the mesh nodes during the interpolation and no information about the interior of a cell is taken into account.

2.3.4 *Treatment of Essential Boundary Conditions*

The basis functions employed in the PUM are in general neither interpolatory on the domain boundary $\partial\Omega$ nor do they vanish on it. Therefore, it is more difficult to impose Dirichlet boundary values in the PUM than in e. g. finite element based methods. Nitsche's variational approach [56] is a common way to workaround this by including the boundary treatment in the PDE. From our perspective it has two downsides, though. First, it depends on a parameter, whose optimal value has to be guessed. Too low values yield suboptimal accuracy or even singular systems and too high values deteriorate the condition number of the resulting linear system. Second, the variational formulation has to be derived by hand from a given PDE, impacting the usability of a software framework. To overcome these limitations Schweitzer introduced [116] a conforming treatment of essential boundary conditions in the PUM detailed in the following. Essentially, it reconstructs what finite elements do: disjointly splitting the degrees of freedom into one set to approximate the PDE and one set to enforce the boundary conditions. Here, the key with respect to performance in the PUM is to achieve that by local transformations only. To this end, we construct direct splittings

$$V_i = V_{i,K} \oplus V_{i,I} \tag{2.94}$$

of the local approximation spaces V_i. We approximate the essential boundary data with $V_{i,I}$ on patches $\omega_i \cap \Gamma_D \neq \emptyset$ that intersect the Dirichlet boundary $\Gamma_D \subset \partial\Omega$ and solve the PDE using $V_{i,K}$. This local splitting naturally induces a splitting of the global degrees of freedom

$$V^{PU} = \sum_{i=1}^{N} \varphi_i V_i = \sum_{i=1}^{N} \varphi_i V_{i,K} \oplus \sum_{i=1}^{N} \varphi_i V_{i,I} = V_K^{PU} \oplus V_I^{PU}. \tag{2.95}$$

Assume we apply Dirichlet boundary conditions on parts of the boundary, that is we want to set $u = \bar{u}$ on $\Gamma_D \subset \partial\Omega$. In this case, consider the bilinear b_i and linear g_i forms of the L^2-projection

$$b_i(u_i, v_i) \coloneqq \int_{\omega_i \cap \Gamma_D} u_i v_i \, ds \quad \text{and} \quad g_i(v_i) \coloneqq \int_{\omega_i \cap \Gamma_D} \bar{u} v_i \, ds \tag{2.96}$$

on patches $\omega_i \cap \Gamma_D \neq \emptyset$ intersecting the Dirichlet boundary. Using the basis $\operatorname{span}\{\vartheta_i^n\} = V_i$ of the local approximation space we assemble the local boundary mass matrices

$$(B_i)_{n,m} \coloneqq b_i(\vartheta_i^m, \vartheta_i^n) \qquad B_i \in \mathbb{R}^{d_i \times d_i} \tag{2.97}$$

and the local right hand side vectors

$$(\widehat{g}_i)_n \coloneqq g_i(\vartheta_i^n) \qquad \widehat{g}_i \in \mathbb{R}^{d_i}, \tag{2.98}$$

where $d_i = \dim(V_i)$. We then split the local approximation spaces V_i into $V_{i,K}$ and $V_{i,I}$ based on the eigenvalues of B_i. Let

$$B_i = Q_i^T D_i Q_i \qquad Q_i, D_i \in \mathbb{R}^{d_i \times d_i} \tag{2.99}$$

be an eigenvalue decomposition of B_i, i. e. D_i is diagonal with $\lambda_s \coloneqq (D_i)_{s,s}$ the $s = 1, \dots, d_i$ eigenvalues and Q_i is orthogonal with rows of eigenvectors of B_i. We further assume that the eigenvalues in D_i are ordered with λ_1 being the largest eigenvalue. By choosing $\epsilon \ll 1$ we then create a block-partitioning

$$Q_i = \begin{pmatrix} Q_{i,I} \\ Q_{i,K} \end{pmatrix} \quad \text{and} \quad D_i = \begin{pmatrix} D_{i,I} & 0 \\ 0 & D_{i,K} \end{pmatrix} \tag{2.100}$$

with $D_{i,K}$ containing all small eigenvalues $\lambda_s \leqslant \epsilon\lambda_1$. Therefore the rows of $Q_{i,K}$ represent the numerical kernel of $b_i(\cdot,\cdot)$. The local approximation spaces $V_{i,K}$ and $V_{i,I}$ used to approximate the PDE solution and the boundary conditions, respectively, can now be defined by

$$V_{i,K} \coloneqq Q_{i,K}(V_i) \quad \text{and} \quad V_{i,I} \coloneqq Q_{i,I}(V_i). \tag{2.101}$$

Moreover, we can define a global, block-diagonal basis transformation T by

$$T_{i,j} \coloneqq \begin{cases} I_{d_i} & \text{if } i = j \text{ and } \omega_i \cap \Gamma_D = \emptyset \\ Q_i & \text{if } i = j \text{ and } \omega_i \cap \Gamma_D \neq \emptyset \\ 0 & \text{if } i \neq j \end{cases} \tag{2.102}$$

that transforms assembled global matrices in the original basis $\{\vartheta_i^n\}$ to the separated basis $\{\tilde{\vartheta}_i^n\}$ by

$$A_{\tilde{\vartheta}} = T A_\vartheta T^T. \tag{2.103}$$

Operators T_K and T_I that transform into $V_{i,K}$ and $V_{i,I}$ can be obtained by replacing Q_i with $Q_{i,K}$ or $Q_{i,I}$ in (2.102).

Now, consider the linear system $A\widehat{u} = \widehat{f}$ assembled in the original basis $\{\varphi_i\vartheta_i^n\}$ and probably with the stable transformation described in the last section already applied. In the system, we want to incorporate the boundary conditions $u = \bar{u}$ on Γ_D. To this end, we split the coefficient vector $\widehat{u}$ into

$$\widehat{u}_K \coloneqq T_K\widehat{u} \quad \text{and} \quad \widehat{u}_I \coloneqq T_I\widehat{u} \tag{2.104}$$

while also transforming it into the basis $\{\tilde{\vartheta}_i^n\}$. Here, $\widehat{u}_K$ now contains all interior degrees of freedom for solving the PDE and $\widehat{u}_I$ all boundary degrees of freedom

used to approximate the boundary conditions. The latter is given as the solution of the local L^2-projection

$$Q_{i,I} B_i Q_{i,I}^T \widehat{u}_I = Q_{i,I} \widehat{g}_i \tag{2.105}$$

on patches $\omega_i \cap \Gamma_D \neq \emptyset$ intersecting the Dirichlet boundary, compare (2.97) and (2.98). Note that solving for the boundary conditions requires patch local operations only and thus is easy to parallelize. With $A_{K,K} \coloneqq T_K A T_K^T$, $\widehat{f}_K \coloneqq T_K \widehat{f}$ and $A_{K,I} \coloneqq T_K A T_I^T$ the interior solution $\widehat{u}_K$ is then given by

$$A_{K,K} \widehat{u}_K = \widehat{f}_K - A_{K,I} \widehat{u}_I. \tag{2.106}$$

The solution in the original basis is then given by

$$\widehat{u} = T^T \begin{pmatrix} \widehat{u}_K \\ \widehat{u}_I \end{pmatrix}. \tag{2.107}$$

2.3.5 *Cover and Integration Cell Construction*

We introduced the abstract notion of an admissible cover for our PUM in Definition 2.5. In this section, we are now interested in an automatic construction of such a cover for a given domain and with reasonable constants C_{FT} and C_S. We detail a simplified version of the construction given in [114, 115]. Further discussion, especially of the implications on stability and accuracy, can be found in [149].

Level zero of the cover C_0, the so-called tree root, is the bounding box of the domain Ω. Though not necessary, we usually start with a cubic bounding box, which guarantees uniform scaling of the tensor product polynomials defined on the patches. For level l of the cover C_l, we refine the root box 2^l times uniformly, such that they have radius $\operatorname{diam}(\Omega)/2^l$. Note that we drop all tree leafs from the cover that do not intersect the computational domain Ω. Figure 2.9 shows an example of a cover tree construction.

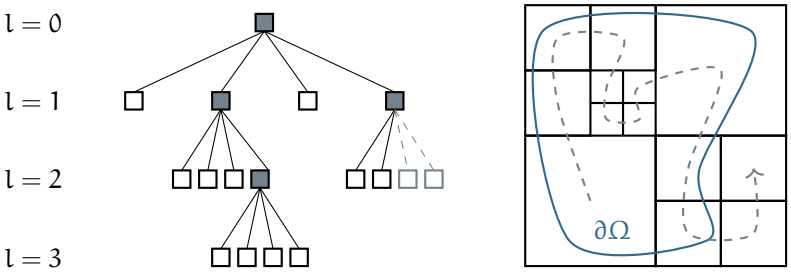
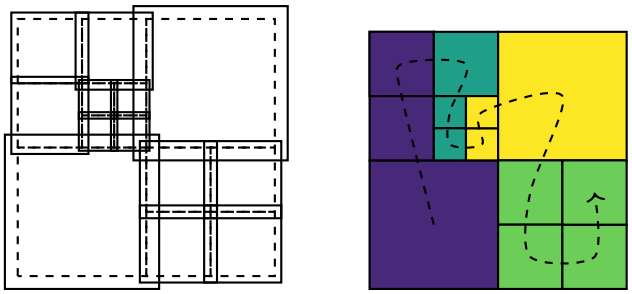


Figure 2.9: Construction of a cover with non-uniform refinements up to level three. Dotted line (right) shows ordering of patches along Hilbert curve. Local patch domains are shown, which are then stretched to define the overlapping patches.

The resulting boxes, which we refer to as the patch local domains, are then stretched by a stretch factor $1 < \alpha < 2$, such that the patches overlap. The cover on level l is thus given by the collection $C_l = \{\omega_{l,i}\}$ of patches

$$\omega_{l,i} = \prod_{k=1}^{d} \left((c_{l,i})_k - \frac{h_l}{2}, (c_{l,i})_k + \frac{h_l}{2} \right), \tag{2.108}$$

where $h_l \coloneqq \frac{\alpha \operatorname{diam}(\Omega)}{2^l}$ and $c_{l,i}$ are the patch centers. In parallel simulations, patches and thus rows of the resulting stiffness matrix are distributed along a space-filling curve through the tree leafs, see Figure 2.10b, where a Hilbert curve [111] is used by default in PUMA.



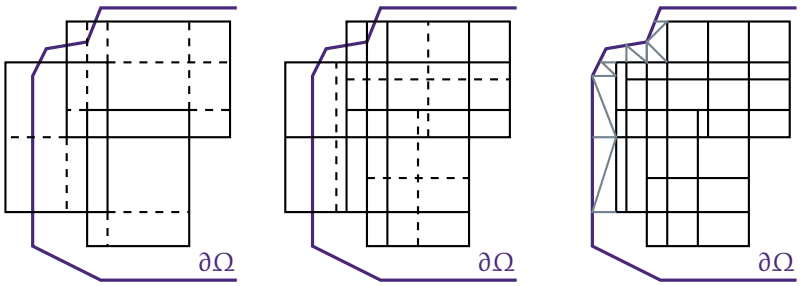
(a) Stretched cover patches. (b) Patch distribution.

Figure 2.10: Patch local domains (dashed) and stretched patches (a). Patch distribution along a Hilbert curve in a parallel simulation with four processes (b).

This construction results in an admissible cover as in Definition 2.5 for uniform refinements, i. e. if we refine all leafs on a given level. If we allow adaptive refinement, the stretch factor may generate patches on lower levels that completely overlap patches on finer levels, as can be deduced from Figure 2.10a. This can be circumvented by capping the maximum level difference between neighboring patches. In PUMA a maximal level difference of one is enforced by default. Note that we further want to drop all patches whose flat-top does not intersect the domain, as discussed in [149]. This prevents linear dependence of the resulting basis. To allow for boundary conditions, neighboring patches do have to be stretched a little more in that case.

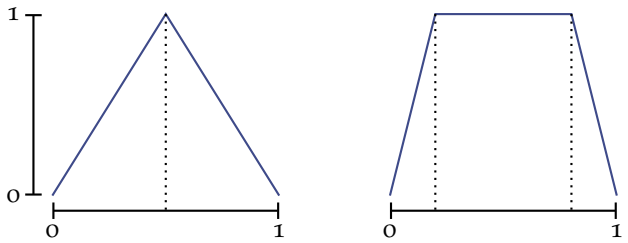
Creating integration cells from the cover C_l then is the three step process sketched in Figure 2.11. First, we compute boxes such that all patch boundaries are resolved. Second, we split those boxes along hyperplanes across which the PU functions or other basis functions are discontinuous in some derivative. For example, a standard linear spline does not have a continuous first derivative in the patch center, whereas the so-called linear flat-top spline has two such points, compare Figure 2.12. Those two points, however, can be aligned with the boundaries of neighboring patches for uniform refinement, in which case the second step can be skipped. As we only employ tensor products of one dimensional splines, finding those split points is trivial. Third, we account for the boundary Γ_Ω of the domain Ω. All boxes intersecting Γ_Ω are discarded and cells respecting the domain boundary are generated inside them instead. For simple geometries this can be achieved using triangles or tetrahedrons. Curved geometries can either be approximated or represented using curved integration

cells, see [149]. The whole process can easily be parallelized on a per patch basis. On the resulting cells numerical integration is performed using tensor product Gauß-Legendre quadrature rules on box cells, quadrature rules by Xiao and Gimbutas on triangles [143] and by Jaśkowiec and Sukumar on tetrahedrons [57].



(a) Tree cells. (b) Basis function splits. (c) Boundary treatment.

Figure 2.11: Three step process to generate integration cells starting from the cover tree cells of three overlapping patches.



(a) Linear spline. (b) Linear flat-top spline.

Figure 2.12: Additional split points required in the generation of integration cells for different types of weight functions.

3 GEOMETRIC CHALLENGES IN FRACTURE SIMULATION

Simulating fracture and crack growth in linear elastic fracture mechanics comes with additional challenges, especially when complicated crack geometries in three space dimensions are considered. We discussed fracture simulation from a modeling perspective in Chapter 2. In this chapter, we are concerned with fracture simulation in the PUM and especially its implementation in PUMA [112, 120], the software framework used for this thesis. To this end, we present how cracks are modeled by enrichment functions in the PUM in Section 3.1 and detail the underlying crack representation allowing their evaluation. In Section 3.2 we discuss the extraction of stress intensity factors from a computed PUM displacement and the prediction and implementation of crack growth. We conclude this chapter with numerical experiments in Section 3.3.

3.1 FRACTURE MODELING IN THE PUM

In fracture simulation, discretizations of linear elasticity must be able to represent the discontinuous displacement across the crack as well as to capture the arising singularity at the crack front. Classical finite elements, for example, duplicate nodes at the crack, thereby change the computational domain, and apply heavy h-refinement at the crack front to achieve this. This in turn couples the crack geometry to the discretization mesh, forcing remeshing in crack growth. In partition of unity based methods, however, it is possible and common to model the crack by special enrichment functions. On patches intersecting the crack but not the crack front, a Heaviside function is used to model the discontinuity across the crack. Around the crack front, we apply so-called Westergaard functions that capture the front singularity, while also modeling the discontinuity. We introduce those functions in Section 3.1.2 and Section 3.1.3 in detail. This allows the discretization to be unchanged during crack growth, except for moving enrichments around, which of course changes the ansatz space.

For those enrichments, we further need a representation of the crack geometry for three purposes. First, we need to know which patches to enrich with the front and Heaviside enrichment functions. To this end, we need to be able to query if a patch intersects the crack and if it intersects the crack front. Second, the enrichments are defined relative to the crack geometry and the representation is thus needed for their evaluation. For the front enrichments, we have to be able to query the distance to the crack front and the relevant crack front basis for a given integration point. For the Heaviside enrichment, we have to know whether a point or cell is above or below the crack. To not introduce integration errors with respect to the latter, it is vital that the final generated integration cells, on which we assemble the linear system, resolve the crack geometry. Otherwise we make an integration error with respect to the sign of the displacement. Thus, third, the crack representation is used to split integration cells.

In the remainder of this section, we introduce the crack representation implemented in PUMA, the specific enrichment functions used to model the crack discontinuity and singularity, and discuss challenges of complex crack geometries on the latter.

3.1.1 *Crack Representation and Integration Cells*

In general, there exist two approaches to crack representation: explicit and implicit. In the former, we explicitly handle the crack geometry, whereas in the latter, the crack is modeled by the level set of one or more functions. For example in [39, 87, 130, 134], the crack geometry is characterized by the level sets of two mesh functions, which also provide the necessary information for the evaluation of the crack enrichments. During crack growth, Hamilton-Jacobi equations have to be solved in a subdomain around the crack, in order to update the crack description [39], which adds computational cost. In [110], explicit and implicit representation is combined. According to the authors, the major issue with an implicit, level-set-based representation is erroneous creation of crack artifacts away from the actual crack surface. The major challenges with an explicit representation are contact of cracks and growth of cracks with non-convex crack fronts. The latter can be solved by applying the face offsetting method [60] to track the evolution of the explicit surface, as was done in [104]. Explicit representations further have advantages in crack geometries with sharp turns and allow for branching cracks, i. e. non-manifold crack surfaces.

Since the domain geometry representation in the PUM is already decoupled from the discretization and we eventually have to resolve the crack geometry in the integration cells anyway, we chose an explicit crack representation in PUMA. Another benefit of this approach is the ability to easily load external crack geometries into PUMA for further simulation. The level set approach can of course also be implemented in the PUM, although differences in runtime performance and accuracy would have to be studied.

IMPLEMENTATION IN PUMA Prior to this thesis only support for cracks in two space dimensions was in place: Here, the crack is modeled by a line string, i. e. consecutive linear segments, where the first and last points on the line are the crack tips. During the generation of integration cells, the segments are resolved in the axis aligned cover cells using a constrained Delaunay triangulation, compare Figure 3.1.

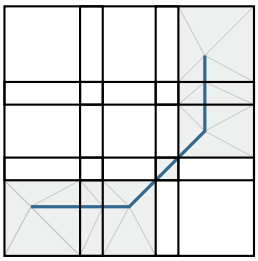
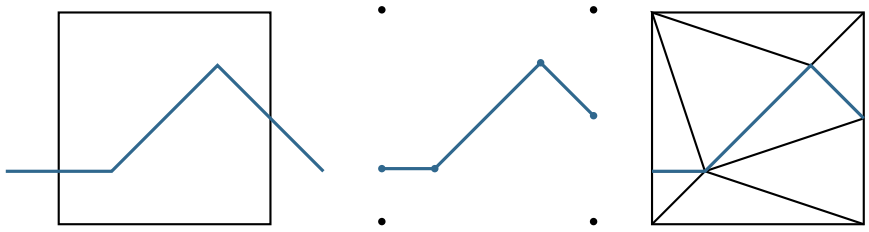


Figure 3.1: Polyline crack through cover patches. Crack does not need to be aligned with the cover, but greyed rectangle cover integration cells require triangulation to resolve the crack during numerical integration.

Similar to [103], we model three-dimensional crack surfaces by a simple triangle mesh. While e. g. CGAL [139] provides mesh data structures that allow traversing neighboring triangles, we do not need that and simply store a list of points and index triples. To ease access to the crack front, however, we additionally store the crack front segments and the indices of corresponding triangles separately. Note that not all boundary segments of the surface mesh have to be part of the crack front, as some might be contained in the domain boundary or even be outside the computational domain.

Resolving the crack surface in the integration cells is then done as follows. Before creating the cells for a patch, we query if the patch intersects the crack. All intersection queries on the crack surface are sped up using axis-aligned bounding box search trees. Assuming that the crack intersects the patch, we next create the axis aligned box integration cells for the patch, compare Figure 2.11. Integration cells resolving the crack surface are then created as sketched in Figure 3.2 in two space dimensions. For each cell we clip the crack surface mesh into the cells geometry. If the resulting mesh is empty, we use the original cell. Else, we add the corners of the cell to the list of the clipped mesh points and pass that to a constrained delaunay tetrahedralizer. A tetrahedralizer then generates delaunay tetrahedrons that mesh the volume of the convex hull of the points we provide. It further guarantees that all provided triangles are present as faces in the resulting tetrahedrons, i. e. that the crack surface is resolved in the resulting integration cells. For all numerical experiments in this thesis, we use TetGen [125] as a tetrahedralizer. On the tetrahedrons we then use the quadrature rules by Jaśkowiec and Sukumar [57] for numerical integration.



(a) Rectangular crack intersecting integration cell.

(b) Clipped crack path and vertices.

(c) Crack resolving integration cells.

Figure 3.2: Generation of integration cells resolving the crack surface sketched in two space dimensions for a polyline crack. This works similarly in three space dimensions, although tetrahedralization (c) is more involved.

Note that multiple cracks inside an integration cell are in principal supported as long as the constraints are manageable by the tetrahedralizer. However, the crack enrichments are of no value in this scenario, aside from the special case of branching cracks. While we assume every crack to be an orientable surface, i. e. every mesh edge has to have two adjacent triangles at most, we support branching cracks through touching meshes. Here, crack meshes must touch in exact matching points and edges, which has to be ensured by the user, though. On patches at the intersection of two cracks, the branch enrichment (3.2) is used to model the arising discontinuities. This was implemented to support the T- or H-shaped cracks in laminated composites, see Figure 6.1.

3.1.2 *Discontinuity*

On all patches intersecting the crack but not the crack front, we use the Heaviside function

$$\mathcal{H}(x) \coloneqq \begin{cases} 1 & \text{if } x \text{ on or above crack,} \\ -1 & \text{if } x \text{ below crack,} \end{cases} \tag{3.1}$$

as multiplicative enrichment (2.85) to introduce a polynomial basis part in which the crack can open. Note that for this to work successfully, it is important that the crack is resolved in the integration cells such that the jump is accurately captured during numerical integration.

Branching cracks require special treatment here. We model those by having two cracks, where the second originates from the first as shown in Figure 3.3. As before, Heaviside functions $\mathcal{H}_1, \mathcal{H}_2$ are applied to the two cracks, but we exclude patches also intersecting the first crack from the second Heaviside. On those patches, i. e. ones that intersect both cracks but no crack front, we use the branch enrichment

$$\mathcal{B}(x) \coloneqq \begin{cases} 0 & \text{if } 0 < \mathcal{H}_1 \\ \mathcal{H}_2 & \text{else} \end{cases} \tag{3.2}$$

multiplicatively, which overall results in local approximation spaces allowing the necessary separation across the cracks.

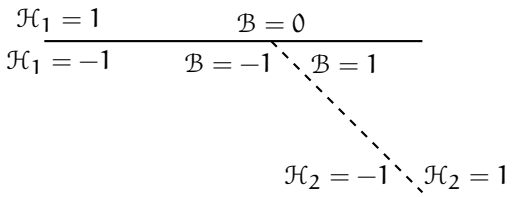


Figure 3.3: Heaviside (3.1) $\mathcal{H}_1$ and $\mathcal{H}_2$ and branch enrichment $\mathcal{B}$ (3.2) functions relative to a branching crack, modeled as a second crack (dashed) originating from the first.

The Heaviside function (3.1) encodes whether we are above or below the crack. As we can assume that only patches intersecting the crack and not the crack front are enriched with it, this is relatively easy to compute. For each integration point, we simply search the closest triangle in the crack surface mesh and determine the points' location relative to the hyperplane it defines. The search again is sped up using an axis aligned bounding box search tree. Note, however, that due to the crack surface being resolved in the integration cells, the Heaviside function is actually constant on every cell. To exploit this, we implemented an observer pattern such that enrichments can be informed every time a new cell is started during integration, and can cache their evaluation accordingly. The branch enrichment (3.2) is implemented as the combination of two Heaviside functions and thus shares the same implementation.

3.1.3 *Singularity*

In this section, we introduce the enrichment functions used to capture the front singularity in two or three space dimensions, but under the assumption of a straight crack front. We cover cracks in a single linear elastic and isotropic material, as well as interface cracks between two such materials. After that, we discuss the area around a crack front in which these functions should be used in order to capture the full singularity and to recover the optimal convergence rates. Finally, we discuss the effects of kinks in the crack front in three space dimensions on accuracy and implementation.

The crack singularity in patches that intersect the crack front or are close to it is modeled by the following set of functions. Around a crack front segment we set up cylindrical coordinates (r, θ, ξ_3), compare Figure 2.6, and define the four tip enrichment functions

$$\begin{aligned} \eta_{\mathrm{tip},1} &\coloneqq \sqrt{r}\sin\left(\frac{\theta}{2}\right), & \eta_{\mathrm{tip},2} &\coloneqq \sqrt{r}\cos\left(\frac{\theta}{2}\right), \\ \eta_{\mathrm{tip},3} &\coloneqq \sqrt{r}\sin\left(\frac{\theta}{2}\right)\sin(\theta), & \eta_{\mathrm{tip},4} &\coloneqq \sqrt{r}\cos\left(\frac{\theta}{2}\right)\sin(\theta). \end{aligned} \tag{3.3}$$

Applied in each space dimension, they span all three tip displacement functions u_{I}, u_{II} and u_{III} and thus successfully capture the crack front singularity [6]. This applies in two and three space dimensions for linear elastic and isotropic materials along straight crack fronts. Directly applying the displacement functions as vector valued enrichments is of course also possible. This trades accuracy for fewer degrees of freedom and, at least in the GFEM, improves the condition number of the linear system [45], making it popular there. While not studied in detail in this thesis, we too found them to be less accurate, but also slower to evaluate during integration, since additional basis transformations of the result vector and its gradient are required. Because of that, we default to the scalar functions in (3.3). Furthermore, the effect of different enrichment formulations for the same singularity on the condition number of the stiffness matrix is less of a concern in the PUM due to the stable transformation outlined in Section 2.3.3.

Cracks at the interface of two linear elastic and isotropic materials, i. e. areas of delamination between material layers, require a different set of crack tip enrichment functions due to the different material properties. Suppose we have materials with Young's modulus and Poisson's ratio E_1, ν_1 and E_2, ν_2 above and below the crack, with corresponding shear moduli G_i and Kolosov constants k_i. The oscillation index or bimaterial constant ε is then given by

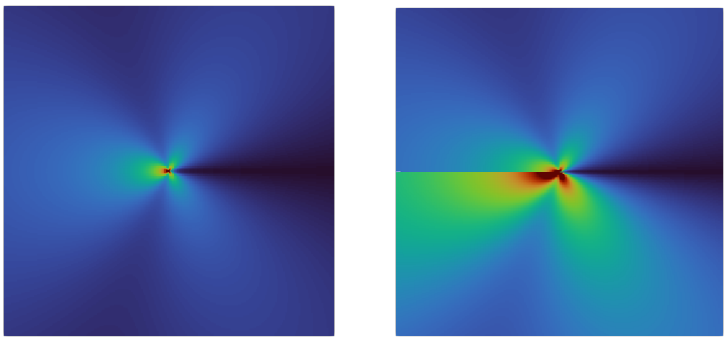
$$\varepsilon \coloneqq \frac{1}{2\pi}\log\left(\frac{G_2 k_1 + G_1}{G_1 k_2 + G_2}\right). \tag{3.4}$$

For bimaterial crack problems, Sukumar *et al.* first proposed [135] a set of twelve enrichment functions. However, these do not take all effects of the different material properties into account. Wang and Waisman then introduced a set of eight crack tip enrichment functions [140], which improve accuracy and

condition number of the resulting stiffness matrix. In cylindrical coordinates (r, θ, ξ_3), they are given by

$$
\begin{aligned}
\eta_{bi,1} &:= \frac{\sqrt{r}}{G}\cos(\varepsilon \log(r))\left[e^{\varepsilon(\Pi-\theta)} + ke^{-\varepsilon(\Pi-\theta)}\right]\sin\left(\frac{\theta}{2}\right),\\
\eta_{bi,2} &:= \frac{\sqrt{r}}{G}\cos(\varepsilon \log(r))\left[e^{\varepsilon(\Pi-\theta)} - ke^{-\varepsilon(\Pi-\theta)}\right]\cos\left(\frac{\theta}{2}\right),\\
\eta_{bi,3} &:= \frac{\sqrt{r}}{G}\cos(\varepsilon \log(r))\left[e^{-\varepsilon(\Pi-\theta)}\right]\sin(\theta)\sin\left(\frac{\theta}{2}\right),\\
\eta_{bi,4} &:= \frac{\sqrt{r}}{G}\cos(\varepsilon \log(r))\left[e^{-\varepsilon(\Pi-\theta)}\right]\sin(\theta)\cos\left(\frac{\theta}{2}\right),\\
\eta_{bi,5} &:= \frac{\sqrt{r}}{G}\sin(\varepsilon \log(r))\left[e^{\varepsilon(\Pi-\theta)} + ke^{-\varepsilon(\Pi-\theta)}\right]\sin\left(\frac{\theta}{2}\right),\\
\eta_{bi,6} &:= \frac{\sqrt{r}}{G}\sin(\varepsilon \log(r))\left[e^{\varepsilon(\Pi-\theta)} - ke^{-\varepsilon(\Pi-\theta)}\right]\cos\left(\frac{\theta}{2}\right),\\
\eta_{bi,7} &:= \frac{\sqrt{r}}{G}\sin(\varepsilon \log(r))\left[e^{-\varepsilon(\Pi-\theta)}\right]\sin(\theta)\sin\left(\frac{\theta}{2}\right),\\
\eta_{bi,8} &:= \frac{\sqrt{r}}{G}\sin(\varepsilon \log(r))\left[e^{-\varepsilon(\Pi-\theta)}\right]\sin(\theta)\cos\left(\frac{\theta}{2}\right),
\end{aligned}
\tag{3.5}
$$

where $\Pi = \pi, G = G_1, k = k_1$ above the crack and $\Pi = -\pi, G = G_2, k = k_2$ below. Figure 3.4 plots the magnitude of the gradients of the second single and bimaterial enrichment functions to visualize the effect of the change in material parameters.



(a) Single material enrichment $\eta_{tip,2}$. (b) Bimaterial enrichment $\eta_{bi,2}$.

Figure 3.4: Gradient magnitude of a second single (a) and a bimaterial (b) front enrichment function. The crack tip is located in the center of the domain. In the bimaterial case, the Young's modulus in the upper half is twice as large as in the lower half.

ENRICHMENT ZONE SIZE In the application of the crack front enrichments, one traditionally chooses between topological or geometrical enrichment [5]. In the former, we enrich all patches intersecting the crack tip, such that the enriched area shrinks when we h-refine our cover. In the latter, we enrich all

patches in a fixed area around the crack tip. Consequently, the enriched area stays constant across discretization levels. Since the singularity dominates the solution in some radius around the crack front, geometrical enrichment has been found to produce more accurate results [5, 46, 117].

In [46], an estimate on the minimally required enrichment radius r around a crack tip in two or three space dimensions has been proven, based on the order $1/\sqrt{r}$ of the singularity. Let h be the mesh size, or in our case, half the size of a cover patch, and p be the order of the polynomials in $\mathcal{P}_i^{p_i}$ around the crack tip. Then

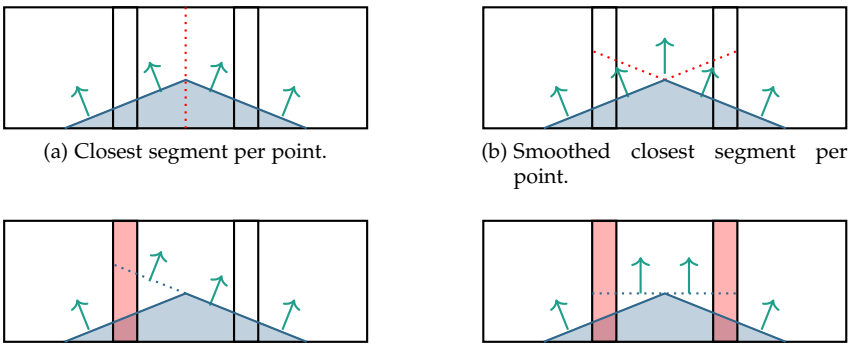
$$Ch^{\frac{1}{2p}} \leqslant r \tag{3.6}$$

is a lower bound on the enrichment radius required to achieve the optimal convergence rates. While the exact size is unfortunately hidden in the constant C, we can observe that higher levels allow for a smaller enrichment radius, while higher polynomial degrees have the opposite effect. Stability issues introduced by increased enrichment radii can be eliminated in the PUM by applying the stable transformation described in Section 2.3.3

CURVED CRACK FRONTS AND IMPLEMENTATION Complex crack front geometries in three space dimensions pose additional challenges. In two space dimensions, we know the exact expansion (2.24) of a solution around a simple straight crack and hence we assume that the corresponding enrichments (3.3) are perfect in an area around the crack tip where the crack is straight. For a simple extruded crack geometry this still works in three space dimensions; however, in complex crack geometries the crack front is not a straight line and it is therefore unclear what its singularity looks like. General crack geometries thus require heavy h-refinement towards the crack front as well as using the front enrichments only close or directly at the crack front. As a consequence, the number and size of patches with kinks in the crack (front) is reduced and more degrees of freedom are present to adapt the front enrichments to the crack geometry. This local increase in degrees of freedom is further the reason why we introduce the global-local method as an approach to separate them onto another discretization in the next chapter.

However, some steps to reduce the required refinement can be taken in the evaluation of the enrichments around kinks in the crack front. There are multiple approaches to implement the evaluation of the crack front enrichments relative to the crack front. Probably the simplest way is to locate the closest point on the crack front for each integration point and evaluate with respect to the local crack coordinate system at that closest point, as sketched in Figure 3.5a. This results in one global enrichment for the whole crack that, however, is discontinuous at kinks in the crack front. A second approach, sketched in Figure 3.5c, is to use separate enrichments per enriched patch along the crack front, each with respect to a single front segment, for example the one closest to the patch. In this case we have a continuously differentiable enrichment on each patch, modulo the jump across the crack, while neighboring patches use different approximations of the crack front. In consequence, approximation spaces of neighboring patches do not model the same singularity, which can lead to the interaction of their enrichments not minimizing the global energy in their overlap. In general, having different and disagreeing enrichment functions for neighboring patches may yield suboptimal results due to (2.84).

Both approaches can potentially be improved by further smoothing the crack front coordinate systems at kinks in the crack front, as shown in Figure 3.5b and Figure 3.5d, which tries to improve the crack front approximation. In [102], the authors tried a version similar to our second approach, compare Figure 3.5d, as well as an approach using a quadratic approximation of the crack front and consequently, smooth, curved enrichment. While the latter approach is harder to implement and computationally more expensive, both performed about the same, with overall accuracy again being solely dependent on the degree of local h-refinement. This was investigated on round cracks, which are perfectly suited for such a smoothing approach, we do not expect much improvements to be possible in the implementation of crack front enrichments for more varying front geometries.



(a) Closest segment per point.

(b) Smoothed closest segment per point.

(c) Closest segment per patch center.

(d) Smoothed closest segment per patch.

Figure 3.5: Four versions of applying the front enrichments relative to the non-straight crack front of a crack surface in three space dimensions. We view from the top and arrows indicate the front normal for each segment, i. e. the coordinate system relative to which we evaluate the enrichment. In the top row the enrichment is discontinuous on patches with kinks in the crack front, but constant across patch overlaps if the kink is in the flat-top region. In the bottom row the enrichment is continuous on each patch, but neighboring patches see different crack fronts in their overlap, and we add another level of approximation to the front representation.

In PUMA, we implemented all four versions of Figure 3.5 and present a numerical experiment to assess their respective accuracy in Section 3.3.3. If not indicated otherwise, we use the global enrichment without smoothing, i. e. the version shown in Figure 3.5a. Regarding the implementation, we have the following remarks.

With a global enrichment we locate the closest point on the crack front for each integration point, which again is sped up using a search tree. If the closest front point is in the interior of a front segment, we evaluate relative to the coordinate system of this segment. If it is a front node, i. e. end or start point of segments, we can interpolate between the coordinate systems of the neighboring segments. With individual enrichments per patch, we initially find the closest front segment to the center of each enriched patch. Evaluation on each patch is then performed relative to this segment's coordinate system, without needing to search segments again.

3.2 FRACTURE PROPAGATION

In this section we provide practical and implementation details on extracting stress intensity factors and updating the crack representation during crack growth, while the underlying math is presented in Section 2.2. We are only concerned with three space dimensions here, since extraction via the CIM and the J-integral method were already implemented in PUMA in two space dimensions prior to this thesis. Additional extraction methods, such as the interaction integral method [1, 144], can easily be implemented using the generic expression compiler for PUMA (GECO) form language, since integration cells for volume based extraction are already available in two space dimensions.

3.2.1 *Extraction of Stress Intensity Factors*

For this thesis, we implemented the CIM and the DCM in PUMA, which we introduced in Section 2.2.2. While the latter is rather easy to implement, since it requires only point evaluations, for the former automatic generation of line integration cells in three space dimensions that also work in parallel had to be added. Moreover, support for the necessary coordinate transformations in the extraction integrals was improved. Another option for the extraction of stress intensity factors in a partition of unity method is the cutoff function method. However, it is the computationally most expensive and hardest to implement of the three methods due to its reliance on volume integrals, while providing only little, if any, benefit in accuracy [44]. It further requires relatively large extraction radii, making it less suitable for non-planar, arbitrary crack surfaces. We therefore refrained from implementing it for this thesis. Regardless of the specific extraction method, we need to discuss the extraction locations and limitations on the extraction radii first.

EXTRACTION LOCATION In two space dimensions, we extract close to the crack tip, whereas in three space dimensions, we have the discrete crack front given by a one dimensional polyline around the crack surface and extract relative to user defined points on that polyline. While having several extraction locations can be useful for long front segments, we usually extract either at the midpoints of the front segments or at the front nodes, i. e. the endpoints of the front segments. The latter is useful if the discrete crack representation approximates a smooth surface and the front nodes are placed exactly on the true crack front, resulting in minimal geometry approximation error. The former is a sensible default in all other situations, since the crack front singularity should be best captured the farthest away from kinks in the crack font, i. e. in the center of each segment. We denote the extraction location for a segment k by $\bar{s}_k$. At each location $\bar{s}_k$, all methods then require a coordinate system (ξ_1, ξ_2, ξ_3) local to the crack front position, compare Figure 3.6. For extraction at the front segment midpoints, we naturally choose ξ_3 as the direction of the front segment, ξ_2 as the normal of the triangle at the segment and ξ_1 perpendicular to the former two and pointing away from the crack. At the front nodes we use the mean of the coordinate systems of the adjacent segments.

1: Compute mid points $\bar{s}_k$ for all n front segments.
2: h-refine such that $\forall i : \mathrm{card}\left(\left\{k \,\middle|\, \bar{s}_k \in \omega_i\right\}\right) \leqslant 1$.
3: Find mid points on local processor domain.
4: Compute minimal distances r_k to patch boundary for all patches $\left\{\omega_i \,\middle|\, \bar{s}_k \in \omega_i\right\}$.
5: Communicate global minimum $r_{ex} \coloneqq \min_k r_k$ of distances r_k.
6: Use r_{ex} as global extraction radius.

Algorithm 1: Computation of a safe radius r_{ex} for the extraction of stress intensity factors at front segment midpoints in parallel and for arbitrary crack geometries especially in crack growth.

EXTRACTION RADIUS In the simulation of three-dimensional fracture, we can have upper limits on the possible extraction radii. First of all, the front enrichments, as well as all extraction methods, assume the crack surface to be flat inside the enrichment and extraction regions. In the PUM, extraction of the stress intensity factors is also most accurate close to the crack front, inside the region enriched with the crack front enrichments [117]. Non-flat crack surfaces thus limit the enrichment and the extraction radius, and their evolution in crack growth is unknown a priori. In parallel simulations, we further have the choice of distributing extraction around each location individually or only distributing extraction locations. For this thesis, we implemented the latter approach, where we have to guarantee, however, that we extract inside the region local to each process. In the following, we thus describe how to arrive at safe extraction radii for each location based on the above requirements, compare Algorithm 1 for a summary. All required functionality is of course available in PUMA.

The approach is based on assumptions on the h-refinement towards the crack front: We have to refine the discretization such that at least front intersecting patches contain as few kinks of the crack surface as possible. Of course, this is impossible to prevent fully in patches that intersect front nodes with kinks between adjacent triangles. Here, a reasonable approach is to refine the discretization until each patch contains at most one node of the current and last crack front. In consequence, enrichments can be applied on all patches intersecting the crack front and extraction can be performed inside them. We thus need a function to determine radii such that extraction happens inside the patch containing each extraction location $\bar{s}_k$. This also guarantees that we extract inside the local process region owning that patch. To this end, we determine the patch which contains an extraction location by testing if the location is contained in the patch local domain. This yields a unique patch for all but the points on the local domain's boundary, in which case we simply gather all the intersecting patches owned by the current process. In a second step, we compute the distance r_k of the extraction location $\bar{s}_k$ to the boundary of that patch or patch set. This distance r_k at that extraction location, or their r_{ex} minimum across all of them, can then be used as a safe extraction radius. In numerical experiments investigating the influence of the extraction radius we further multiply the computed radius by a factor f_{ex}. Of course, other user defined radii are possible, too, in case one knows that larger radii are applicable based on the crack geometry and parallel distribution.

CONTOUR INTEGRAL METHOD For the CIM, we have to numerically evaluate the integrals in (2.30). For example, the mode I factor is given by

$$K_I = \sqrt{2\pi} \int_{\Gamma_*} \mathcal{T}(u_{PU}) \cdot \frac{1}{C_I} v_I \, ds - \int_{\Gamma_*} \frac{1}{C_I} \mathcal{T}(v_I) \cdot u_{PU} \, ds. \tag{3.7}$$

Hence, we have to evaluate the computed displacement u_{PU} and the analytically given function v_I on the closed path Γ_* around the crack front. The extraction path Γ_* has to be contained in the $(\xi_1 \xi_2)$-plane through the extraction location $\bar{s}$, which is perpendicular to the front tangent ξ_3, compare Figure 3.6.

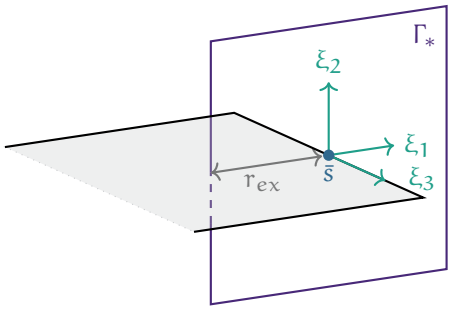


Figure 3.6: Rectangular extraction path Γ_* with radius r_{ex} around extraction location $\bar{s}$. Further, the front local coordinate system ξ_1, ξ_2, ξ_3 at that point.

Therefore, we have to create integration cells for such a path. If we extract from an analytical solution around a crack front or tip, the value of the integral is independent of the specific path, as we validate numerically in Section 3.3.1. In consequence, we can choose a simple rectangular path around the crack front, which is easy to implement. Extracted stress intensity factors from a numerical solution, however, will likely vary with the local accuracy of the computed displacement. To this end, extracting inside the front enriched patches is beneficial, as was also confirmed numerically in [44].

In the following, we detail the generation of integration cells for extraction via the CIM in the case of extraction at the crack front mid points. Extraction at front nodes, i. e. segment end points works analogously. As a preparatory step, we apply sufficient refinement towards the crack front, as discussed before, and determine an extraction radius r_{ex} based on Algorithm 1. Around the mid point $\bar{s}_k$ of each crack front segment, we create a square shaped polyline path of radius r_{ex} in the plane perpendicular to the crack front tangent, compare Figure 3.6. We further split the parts of the generated polyline based on a user provided factor, in a way that all resulting parts are of equal length. This way, we can adjust integration quality of the extraction integrals independent of the function space resolution. These segments then are the integration cells.

Note that we ideally would also resolve the function spaces patch boundaries in the generated integration cells to account for discontinuities in the basis functions, but this has not been implemented yet. Tests with increasingly refined extraction cells indicate that this is not a practical issue, though, as the generated integration cells are rather small compared to the function space patches. On the resulting segments, intersecting patches are precomputed to speed up later evaluation of basis functions. In a parallel simulation, each process only generates paths and extracts around segment mid points $\bar{s}_k$ that are contained

in the local domains of the patches it owns. Local results are then communicated after extraction.

Note that unlike in FEM based methods, our extraction paths do not line up with patches in any way. The dependence of the extracted stress intensity factors' accuracy on the extraction radius is studied in Section 3.3. While not further described here, we found that for accurate results we have to assign rather high polynomial degrees to the extraction functions (2.31), i. e. we have to apply quadrature rules of relatively high degree on the generated extraction cells.

DISPLACEMENT CORRELATION METHOD The DCM introduced in Section 2.2.2 has the advantage of only requiring point evaluations of the numerical solution. Consequently, it is both easier to implement than the CIM and more runtime efficient. As the results in Section 3.3 show, it is more robust with respect to the extraction radius and provides better results in most cases.

Again, we describe the algorithm for extraction at the crack front segment mid points, yet extraction at the crack front nodes works analogously. For each mid point and a pair of extraction radii, we compute $\widehat{K}$ from (2.40) for each crack opening mode as follows. On each process, we loop over all extraction points that are contained in the local domains of patches owned by the current process. For these, we compute the extraction radius r_{ex} as detailed in Algorithm 1. We then compute $\widehat{K}$ at equidistant points slightly above and below the crack, compare Figure 3.7.

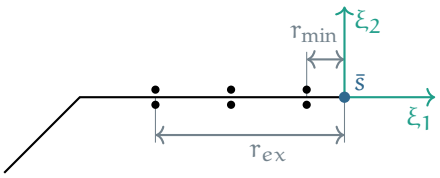


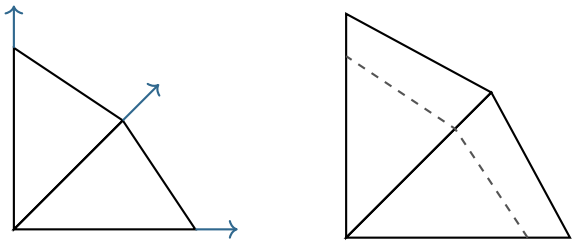
Figure 3.7: Sketch of extraction points of the DCM in (ξ_1, ξ_2)-plane at an extraction location $\bar{s}$.

These points are positioned in negative ξ_1 direction behind the extraction location on the crack front, at a distance between r_{ex} and $r_{min} := r_{ex} c_m$, where c_m is a user defined constant. In this thesis, unless stated otherwise, we use $c_m = 1/10$ as the factor for the minimal extraction radius. Pairs of results of neighboring radii are combined via Richardson extrapolation, as discussed in Section 2.2.2 and the final stress intensity factors are taken as the average of those values. Finally, we gather all remote results such that each process has the full and identical list of extracted values. Experiments suggest that a wider extraction range yields better results, while around 10 extraction points suffice.

3.2.2 *Crack Geometry Update*

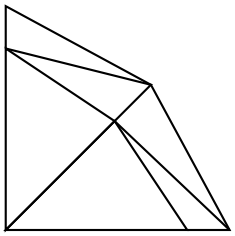
Based on the extracted stress intensity factors, the direction of crack growth can be estimated. For this purpose, several models are available in the literature, and we have chosen Schöllmann's criterion in Section 2.2.3, which estimates two crack growth angles in front local coordinates for each extraction point. Based on these angles, we can compute an update direction for each extraction location

in global coordinates, which we use to define update vectors for each crack front node. Note that the crack growth increment, i. e. the length of each update vector, is user defined in a simulation. Based on the relative comparative stress values (2.48), relative update lengths for each node can be determined, though. There are currently three algorithms implemented in PUMA for applying the estimated growth direction to a given crack geometry, see Figure 3.8, originally presented in [104]. For in-plane crack growth, the current front nodes are shifted in the update direction and no new triangles are added, compare Figure 3.8b. If the crack grows out of the plane containing the triangles connected to the current crack front, we add new front nodes and triangles accordingly, compare Figure 3.8c. If in that case the length of the new front segments exceeds some user defined threshold, we further refine some of the new triangles by adding points along the crack front, compare Figure 3.8d.

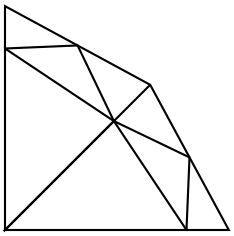


(a) Initial geometry with update vectors.

(b) Propagate and smooth (PAS).



(c) Propagate and extrude (PAE).



(d) Propagate and extrude with refinement (PAER).

Figure 3.8: Schemes to propagate the crack front in fracture simulations as introduced in [104] and now implemented in PUMA.

While this works for appropriate update vectors, one further issue needs to be addressed in future work: the surface description should never be self-intersecting. One simple example which illustrates this issue is a flat crack with concave crack front that grows inside its plane, as sketched in Figure 3.9. Here, new triangles of the opposing faces in the intruding part could easily overlap. An algorithm to address this based on the face offsetting method is presented in e. g. [104] but still needs to be implemented in PUMA. Therefore, our current implementation is only valid for crack surfaces with convex crack front.

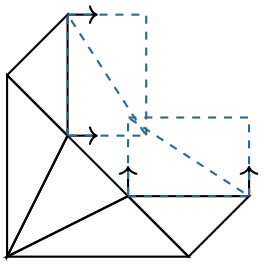


Figure 3.9: Possible self intersections in the update of concave meshes, depending on the update directions and increment. Algorithms to circumvent this issue are available in the literature and have to be implemented.

3.3 NUMERICAL RESULTS

We conclude this chapter with numerical experiments testing the adaptation of fracture simulation in three space dimensions in the PUM. First, we test whether our implementation of the CIM is indeed path independent when extracting from an analytic function, and if that property also holds in the extraction from a computed displacement. Second, we validate our whole implementation on an inclined elliptical crack example, for which the reference stress intensity factors around the crack front are known analytically. This example is reported in other publications, hence our results can be compared to that of other methods. Third, we investigate the effect of different techniques to improve the front enrichment around kinks in the crack front on the same example. Lastly, we run a crack growth simulation of an initially penny shaped crack and check the obtained crack path.

3.3.1 *Path Independence*

As mentioned in Section 2.2, extraction of the stress intensity factors via the CIM is independent of the chosen extraction path as long it is closed around the crack tip. This of course only applies when extracting from an analytic near tip function. When extracting from a numerical approximation, we expect spatial differences in approximation quality to hinder path independence. To test this, we simulate an extruded square domain with a through the thickness crack from the left side to the center of the domain, compare Figure 3.10. On the boundary, we apply Dirichlet conditions such that the solution of the problem is given by

$$u^* := u_I + u_{III} \tag{3.8}$$

the superposition of the first terms of pure mode I and mode III crack opening given in (2.28). To test path independence, we extract on several square shaped paths around the mid point of the domain and crack front with varying extraction radius, as described in Section 3.2.1. We further compare extraction from the analytic function u^* and a numerical approximation u_{PU} of it. To this end, we solve the problem on a level five discretization with the front enrichments (3.3) added to all patches on the domain to further improve accuracy.

Results are shown in Figure 3.11. While we present relative errors for mode I and III, we show the absolute error for mode II, as the reference is zero here.

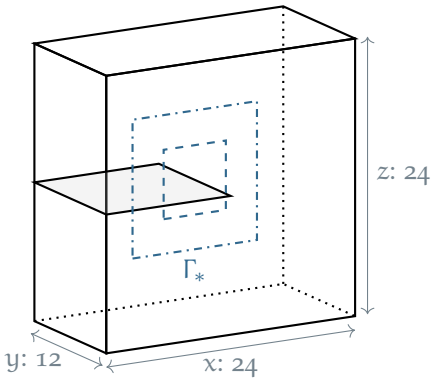


Figure 3.10: Cracked domain to study path independence of extraction of stress intensity factors via the CIM. Extraction paths Γ_* with different radii around the center of the crack front are indicated. The crack front itself is along the x, z center of the domain. Boundary conditions are applied such that we know the analytic solution of this problem.

We can clearly observe that extraction from the analytic function is indeed path independent with an error in the order of machine precision, validating our implementation of the three-dimensional CIM in PUMA. While the error in extracted stress intensity factors from the numerical approximation is overall low, the values vary with the extraction radius, probably due to varying approximation quality in the domain or relative position of the extraction path to flat-top or overlap regions. Regarding the latter, note that a patch in this example has diameter $24/2^5 = 0.75$ and the crack front is right in the middle of the central patch overlap. Extraction further away from the crack front seems to perform better in this example, but is unfortunately not possible in more complex crack geometries.

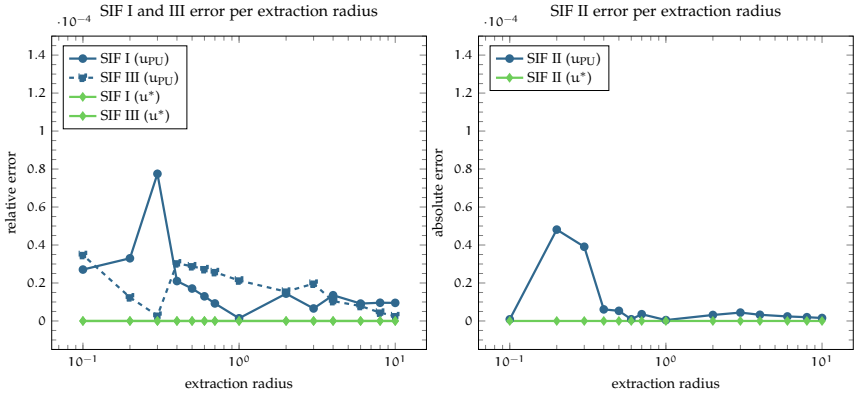


Figure 3.11: Errors in extracted stress intensity factors (SIF) from the analytic u^* and from a numerical solution u_{PU} on varying extraction radii. Extraction from the analytic solution via the CIM is path independent, extracting from a numerical solution clearly is not, due to the approximation error.

3.3.2 *Inclined Elliptical Crack*

With this example, we validate our adoption of the CIM and DCM in the PUM on a mixed mode problem with non trivial crack geometry. Thereby, we further validate our whole implementation of fracture simulation, from geometry representation over integration cells to the front enrichments. To this end, we reproduce example 4.2 in [44], an inclined elliptical crack in cubic domain. Analytic reference stress intensity factors for this crack geometry and loading are available in [51], although the analytic crack problem is studied in an infinite domain there. As we run this example in parallel due to number of degrees of freedom required for the resolution of the crack front, we further test that aspect of the implementation, too.

The crack ellipse is centered in the coordinate system and has semi-major and semi-minor axes $a = 0.10$ and $b = 0.05$ respectively. It is further inclined by $\zeta = \pi/4$ with respect to the global x-axis, compare Figure 3.12. To reduce effects of the finite computational domain, the cube is set up with a radius of 1.0. We again simulate an isotropic and linear elastic material with Young's modulus $E = 1000$ and Poisson's ratio $\nu = 0.3$ and apply unite tensile traction $\sigma = 1.0$ in normal direction on the top and bottom in y-direction of the domain.

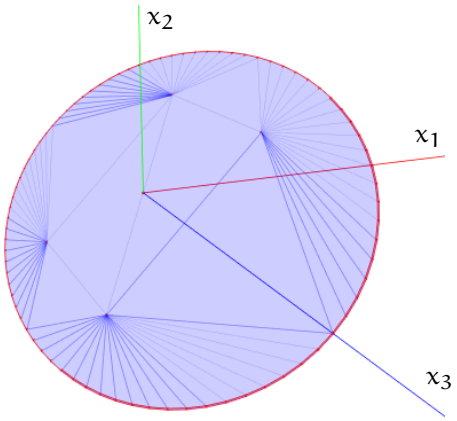


Figure 3.12: Mesh of the inclined ellipse crack for the validation of stress intensity factor extraction.

With $k' := b/a$ and $k := \sqrt{1 - k'^2}$, the reference stress intensity factors for this problem in an angle $\gamma \in [0, 2\pi]$ around the elliptic crack are given by [51]:

$$
\begin{aligned}
K_I^* &= \frac{\sigma \sin^2(\zeta)\sqrt{\pi b}}{E(k)} \left(\sin^2(\gamma) + \left(\frac{b}{a}\right)^2 \cos^2(\gamma) \right)^{1/4}, \\
K_{II}^* &= \frac{-\sigma \sin(\zeta)\cos(\zeta) k^2 \sqrt{\pi b}}{\left(\sin^2(\gamma) + \left(\frac{b}{a}\right)^2 \cos^2(\gamma) \right)^{1/4}} \left(\frac{k'}{B} \cos(\frac{\pi}{2}) \cos(\gamma) + \frac{1}{C} \sin(\frac{\pi}{2}) \sin(\gamma) \right), \\
K_{III}^* &= \frac{\sigma \sin(\zeta)\cos(\zeta) k^2 (1-\nu) \sqrt{\pi b}}{\left(\sin^2(\gamma) + \left(\frac{b}{a}\right)^2 \cos^2(\gamma) \right)^{1/4}} \left(\frac{1}{B} \cos(\frac{\pi}{2}) \sin(\gamma) - \frac{k'}{C} \sin(\frac{\pi}{2}) \cos(\gamma) \right).
\end{aligned}
\tag{3.9}
$$

Here, the constants B and C are given by

$$\begin{aligned} B &:= (k^2 - \nu)E(k) + \nu k' K(k), \\ C &:= (k^2 + \nu k'^2)E(k) - \nu k'^2 K(k), \end{aligned} \tag{3.10}$$

where $K(k)$ and $E(k)$ are the elliptic integrals of first and second kind:

$$\begin{aligned} K(k) &:= \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2(\phi)}} \, d\phi, \\ E(k) &:= \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2(\phi)} \, d\phi. \end{aligned} \tag{3.11}$$

Figure 3.13 shows the raw extracted values at the crack front vertices around the ellipse together with the reference values.

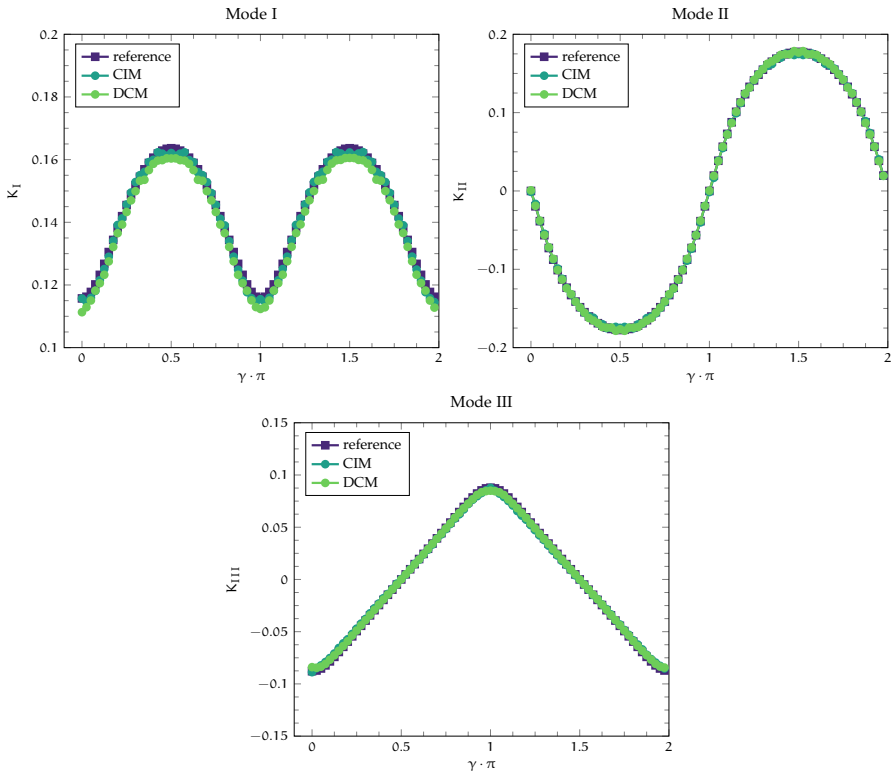


Figure 3.13: Stress intensity factors around (γ) inclined ellipse, analytic reference and raw extracted values in PUMA. We use the computed extraction radius r_{ex} multiplied by a factor of $f_{ex} = 1$ in the CIM and $f_{ex} = 3$ in the DCM.

Regarding the experimental setup, we discretize the cubic domain on level five and then apply local h-refinement around the crack and towards the crack front such that we end up with the finest patches on level ten. That is, at the crack front we have $h = \text{diam}(\Omega)/2^{10+1} \approx 1 \times 10^{-3}$. We employ an iterative solver, a simple CG with block Jacobi preconditioner. We approximate the crack front by eighty segments, compare Figure 3.12 and extract at the front vertices, as

these are located exactly on the ellipse. The minimal extraction radius computed by Algorithm 1 is $r_{ex} = 2.96\times10^{-4}$, so approximately a third of the patch radius at the crack front. In [44], a moving least squares interpolation of the extracted stress intensity factors is adopted to smooth oscillations, since in this example the factors form a smooth curve around the ellipse. Following their example, we smooth the raw extracted values with a simple centered moving average of length three for comparison. However, in future work more sophisticated algorithms might be tried as well.

To quantify the accuracy of the extracted stress intensity factors, we compute the relative error $e(K_i)$ between the extracted values K_i and the references K_i^* by

$$e(K_i) := \frac{\|e_i\|_{l^2}}{\|K_i^*\|_{l^2}} = \frac{\sqrt{\sum_{j=1}^{M}\left(K_i^{(j)} - K_i^{*(j)}\right)^2}}{\sqrt{\sum_{j=1}^{M}\left(K_i^{*(j)}\right)^2}} \tag{3.12}$$

per crack opening mode $i \in [I, II, III]$ over all M extraction locations around the crack front. Table 3.1 shows the errors in smoothed extracted stress intensity factors of the inclined ellipse crack problem simulated by PUMA in comparison to results obtained in the GFEM [44]. Concerning discretization parameters, [44] reports h to be between 1.60×10^{-3} and 2.90×10^{-3} along the crack front, and the application of quadratic polynomials as basis functions. We use linear polynomials but slightly smaller $h \approx 1\times10^{-3}$ at the crack front. Results are quite similar, which validates our implementation.

Table 3.1: Error in extracted and smoothed stress intensity factors in comparison to results obtained in [44] with the GFEM. Extraction in PUMA with the computed radius r_{ex} multiplied by a factor of $f_{ex} = 1$ in the CIM and $f_{ex} = 3$ in the DCM.

		$e(K_I)$	$e(K_{II})$	$e(K_{III})$
PUMA	CIM	0.007	0.009	0.036
	DCM	0.017	0.005	0.023
GFEM	CIM	0.020	0.016	0.066
	DCM	0.020	0.010	0.026

For further comparison, Figure 3.14 shows the errors for a range of extraction radii, as done in [44], yet only for the CIM and DCM. Overall shape of the curves matches with their reported results, i. e. both too small and too large extraction radii yield suboptimal results. While the optimal range of radii is problem-specific and unknown a priori, we can observe that extraction quality deteriorates once the factor f_{ex} is bigger than four, which is approximately when we start evaluating outside the patches that intersect the crack front, in this example. Besides being much faster, the main benefit of the DCM is that it is less dependent on the extraction radius than the CIM. Essentially, underestimation of the required extraction radius produces robust results. Interestingly, the extraction accuracy does not differ that much between opening modes in our implementation, except that mode III is a little more radius dependent. Also, the DCM works way better with small extraction radii in our implementation.

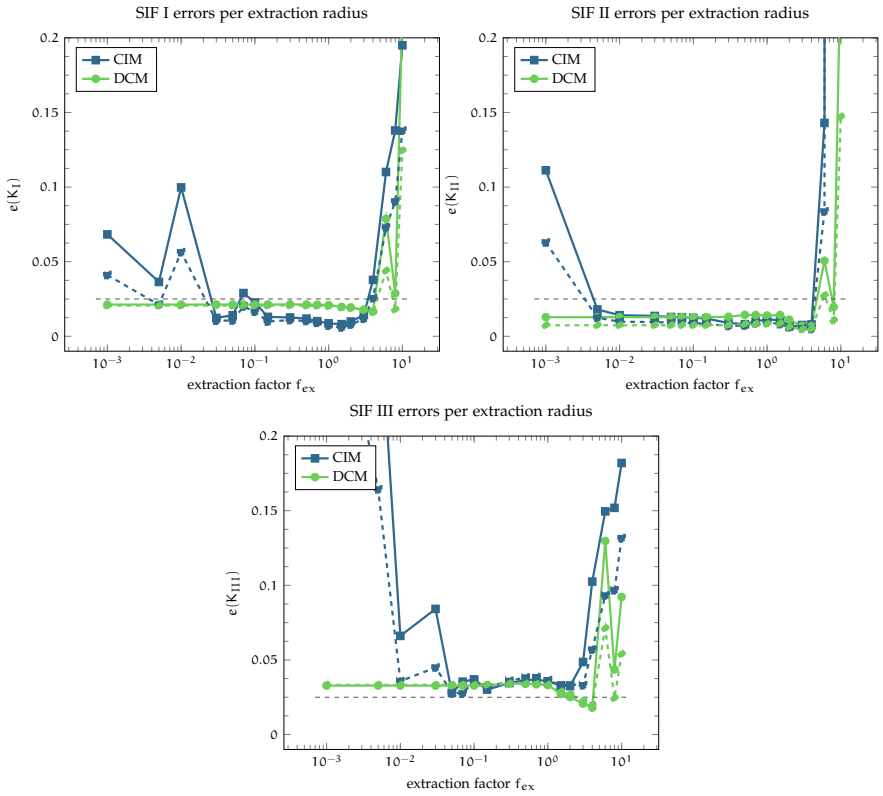


Figure 3.14: Errors in stress intensity factors (SIF) for different extraction radii $f_{ex} \cdot r_{ex}$ with a computed extraction radius of $r_{ex} = 2.96 \times 10^{-4}$. Dashed lines indicate a centered moving average of length three of the extracted values. Dashed horizontal line indicates error of 2.50 percent.

While further techniques to improve the quality of the extracted stress intensity factors, such as MLS based smoothing, could be implemented, we conclude that our adaption of fracture simulation algorithms in the PUM works as intended and thus opens several new research possibilities with PUMA. To arrive at this accuracy in this simple example with a single crack, we had to invest over a million degrees of freedom and about 8000 patches were enriched with the crack front enrichment. We believe, that this further highlights the need for scale separation in linear elastic fracture mechanics, which we introduce by means of the global-local method in the next chapter.

3.3.3 *Front Enrichment Study*

In Section 3.1, we presented four options to implement the evaluation of the crack front enrichments around kinks in the crack front. As shown in Figure 3.5, we either have a global (for the whole front) enrichment and evaluate it with respect to the closest segment (a) and (b) or have individual enrichments per patch with respect to the closest front segment (c) and (d). In the first case (a/b), the enrichment is discontinuous between segments, in the second (c/d), enrichments of neighboring patches are different. The two further options (b/d)

per case are created by optionally applying some smoothing of the crack front between segments for the enrichment evaluation.

To study the effects of the different enrichments, we revisit the problem from the previous Section 3.3.2, but now with a fixed extraction radius of 3.93×10^{-3} for all extraction locations, which is equal to half the radius in which we use the front enrichments around the crack front. We also only extract via the DCM and again report the errors (3.12). Compared to the previous setup, we additionally report results with one patch refinement less around the crack, to further study the effect of reduced accuracy in the numerical solution we extract from.

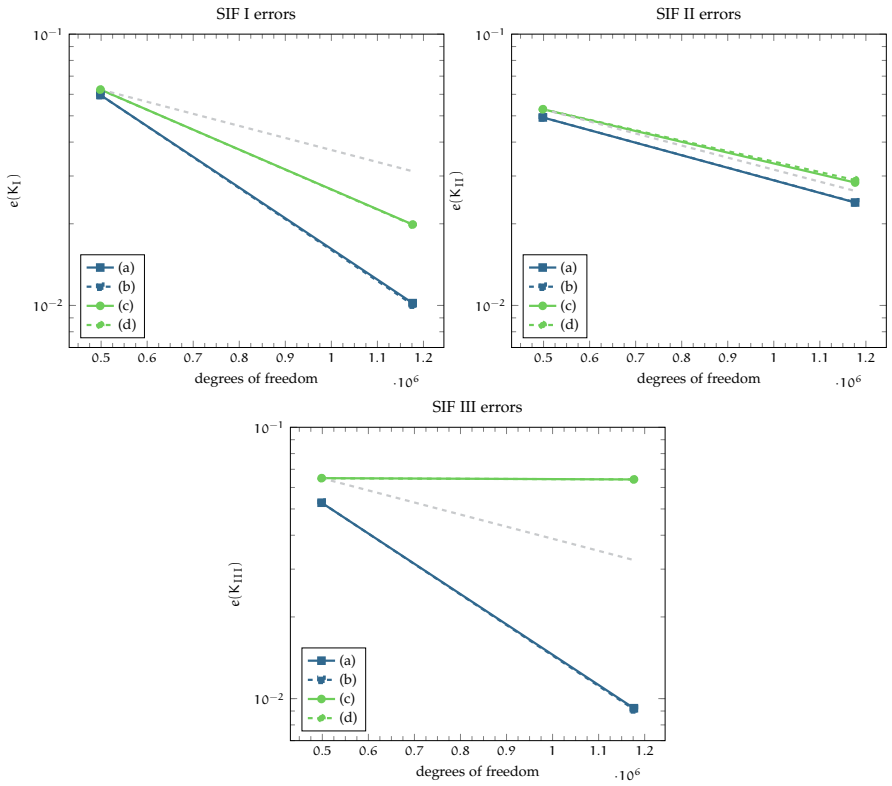


Figure 3.15: Errors of stress intensity factors (SIF) around an inclined penny shaped crack, extracted from numerical solutions computed with different versions of evaluating the front enrichment functions. Versions are defined in the text and visualized in Figure 3.5. Linear convergence rate is indicated by the grey dashed line.

Figure 3.15 shows the results for all three crack opening modes. First of all, trying to smooth the evaluation of the front enrichment functions between front segments only has a negligible effect. This might be due to the already fine resolution around the crack that could be hiding the issue, but we did not see effects with coarser discretizations either. More likely, the difference in the local coordinate systems between adjacent front segments is too small to be an issue here and we would require less smooth crack geometries to observe effects. In contrast, having a global enrichment with discontinuous evaluation for the whole crack versus individual enrichments per patch makes a difference, especially with a finer discretization. As we expected in Section 3.1.3, having different

enrichments in neighboring patches that model different crack fronts, and thus singularities, in their overlap yields worse results than a global discontinuous enrichment. One might further argue that the discontinuity in the enrichment of the first approach is hidden by inexact numerical integration anyway. Hence, in this example the naive approach of simply evaluating relative to the closest front segment for each integration point produces the most accurate results and was thus employed in all other experiments in this thesis. Note that our findings are in line with [102], where the authors tried even more elaborate approaches to smooth the enrichment functions around the crack front, but found that essentially only the overall discretization level around the crack mattered. Regarding the runtime of the individual enrichment schemes, we found the naive version, where we search the closest point on the crack front for each integration point, to actually perform consistently better than the seemingly cheaper approach of constant enrichments per patch. Most likely, this version is faster, since only one enrichment has to be evaluated versus around a thousand in the other version.

3.3.4 *Crack Growth*

In this example, we use the extracted stress intensity factors to guide the growth of an inclined penny shaped crack under uniaxial load, reproducing experiment 5.4 in [104]. We expect two qualities of that growth process: First, the symmetric crack together with symmetric loading should result in symmetric growth. Second, the crack should grow such that mode I is the dominant opening mode, i. e. the result should be a flat crack for these boundary conditions.

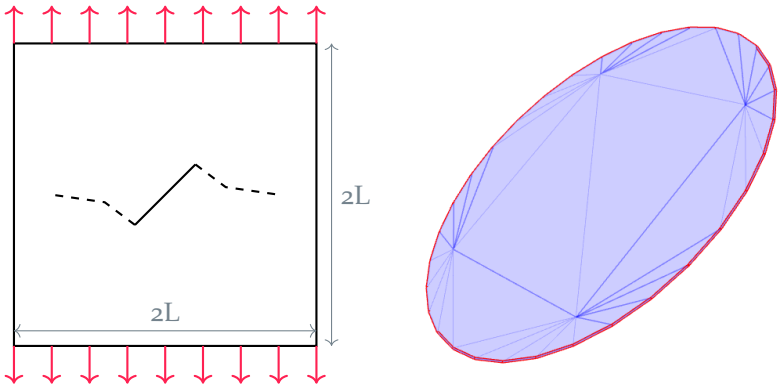


Figure 3.16: Front view (xy) of computational domain Ω with inclined penny shaped crack (left). Approximate crack growth direction is indicated by dashed lines. Discretization of penny shaped crack in initial configuration (right).

We simulate a cracked, cubic domain with radius $L = 1$, see Figure 3.16, which is expanded after some crack growth steps to keep enough distance between the crack and the domain boundary. The round crack with radius $a = 0.1$ centered at the origin is inclined by $\zeta = \pi/4$ with respect to the global x-axis, compare Figure 3.16. The round crack front is discretized by 40 segments. Unit load $\sigma = 1$ is applied at the boundaries above and below the crack in normal direction. We discretize the domain on level four and apply three further

adaptive h-refinements towards the crack front. Additional h-refinements are applied on top such that no two vertices of the current and last crack front are contained in a single patch during crack growth. This ensures that the crack is planar inside the patches at the crack front, which is required for the crack front enrichments. For extraction, we employ the DCM with a radius factor of 1×10^{-1}. For simplicity, we apply a fixed increment of 6×10^{-1} times the current front segment length to update the crack at each step. The kink and twisting angles $\widehat{\alpha}$ and $\widehat{\psi}$ for the crack update are then computed using Schöllmanns criterion by (2.45) and (2.46).

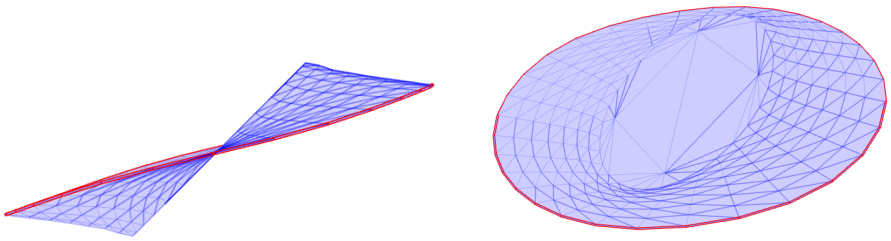


Figure 3.17: Penny shaped crack after eight growth steps.

Figure 3.17 shows the crack configuration after eight growth steps. As expected, crack growth is smooth and symmetric. The reference stress intensity factors for the initial penny shaped crack are given by [51, 104]

$$\begin{aligned} K_I^* &= \frac{2\sigma\sqrt{\pi a}}{\pi}\sin^2(\zeta), \\ K_{II}^* &= \frac{4\sigma\sqrt{\pi a}}{\pi(2-\nu)}\sin(\zeta)\cos(\zeta)\cos(\gamma), \\ K_{III}^* &= \frac{4\sigma\sqrt{\pi a}}{\pi(2-\nu)}\sin(\zeta)\cos(\zeta)\sin(\gamma), \end{aligned} \tag{3.13}$$

for an angle γ around the crack. In Figure 3.18 we plot the extracted values in the initial and in the final configuration and add the reference values (3.13) in the former. While we slightly underestimate the initial stress intensity factors due to the coarse discretization, mode II and III contributions clearly vanish as the crack grows, as expected. Figure 3.19 further illustrates that the crack is almost planar in the final growth step.

We conclude that the simulation of mixed-mode crack growth in isotropic, linear media works as expected in the PUM, opening up new research possibilities with PUMA. We present three potential topics in Chapter 6.

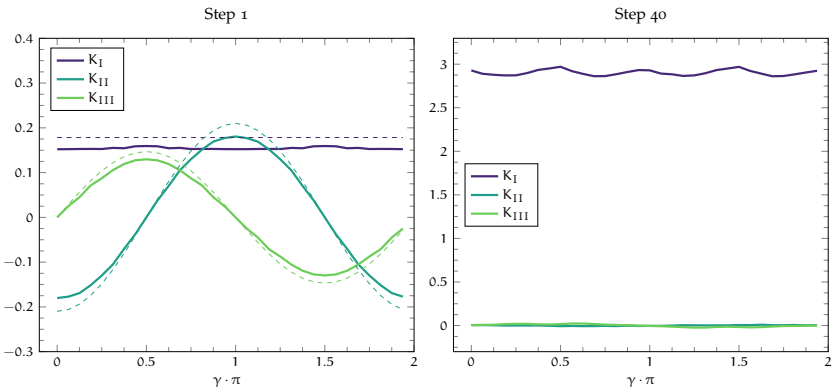


Figure 3.18: Stress intensity factors around (γ) inclined round crack. Dashed line in left plot indicate the analytic reference values for this problem. Crack grows to be mode I dominant.

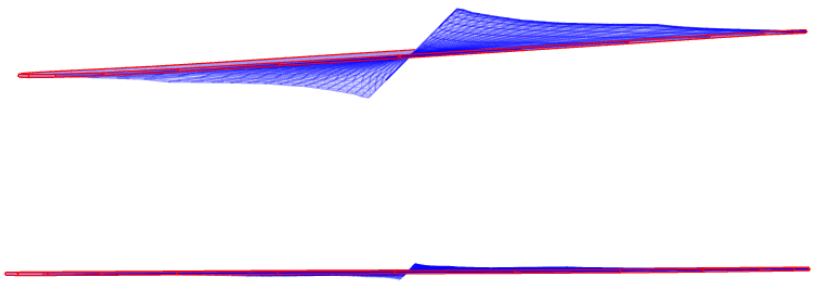


Figure 3.19: Crack growth steps 32 (top) and 41 (bottom) of the inclined penny shaped crack.

4 GLOBAL-LOCAL ENRICHMENTS

Let Ω_G be the global domain with a very localized feature of interest, like a crack, a hole, an inclusion or a sharp laser heat source, for which we do not have an analytic or precomputed enrichment available. Using heavy h-refinement around the feature increases the number of degrees of freedom, such that solving the global system may become unreasonably expensive. A two-scale approach to this problem is to separate the involved scales and solve for the smooth global problem and the localized feature separately. This approach has been researched and used since the 1970s as the so-called global-local finite element analysis (FEMGL) [88, 94], the zooming technique [27, 136] or the micro-macro strategy [70]. In the FEMGL, the global problem is solved disregarding the localized feature around which a local problem is set up. The two problems are combined using some interface condition and the resulting method is similar to non-overlapping Schwarz domain decomposition method [37]. This splits the number of degrees of freedom into two separate problems while retaining local resolution.

Resolving the localized feature can significantly change the global energy minimum we are solving for. In the global-local enrichments method, the local solution is thus used as an enrichment in the global problem, which is then solved again to incorporate the new information, compare Figure 4.1. This concludes one global-local cycle, which may be iterated to further improve the computed enrichment.

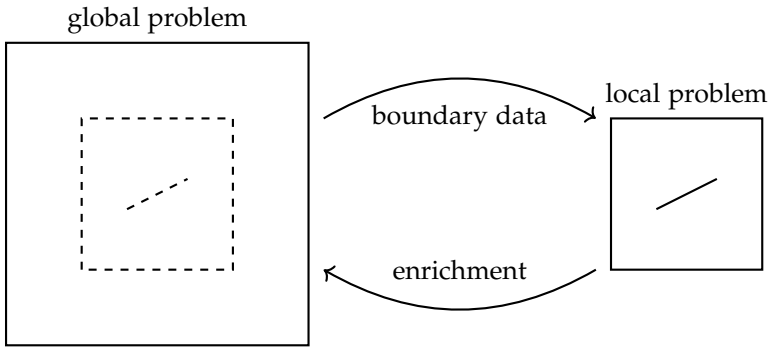


Figure 4.1: The original global-local enrichments cycle: The solution of the local problem computed with boundary data from the global problem is used to compute another enriched global solution. Updating the global solution is beneficial since the resolution of the local feature can change the global energy minimum.

Using multiple iterations to account for the global effect of the improved resolution of the local feature has already been successfully tried in the initial global-local version without enrichments [141], where a Newton-Raphson iteration was employed to equilibrate the solutions on the global and local meshes. The global-local enrichments method was introduced as the GFEM^{gl} in 2007 by

Duarte and Babuška [23] for the GFEM, which is a finite element based partition of unity method. Since then, it has been applied to a variety of problems with different local features: sharp thermal gradients [96, 97], linear [63, 68] and cohesive [62] fracture, localized [64] and thermo-plasticity [107], simulation of spot welds [73], as well as local material heterogenities [106]. From now on, we refer to the global-local enrichments method depicted in Figure 4.1 as just the global-local method or global-local enrichments.

4.1 PROPERTIES OF THE GLOBAL-LOCAL METHOD

In this section, we first discuss accuracy and runtime performance of the global-local method. We then detail the local equation and boundary conditions for the case of a global linear elasticity problem and end with a comparison of two ideas on what constitutes a local problem in a simulation.

ACCURACY In [49], Gupta *et al.* derived an error estimate for the global-local method for smooth and simple PDEs based on Caccioppoli's inequality. The setting, sketched in Figure 4.2, is a local domain Ω_L with a smooth boundary and a smaller domain $\Omega_L^\tau \subset \Omega_L$ inside it, also with a smooth boundary. In the global-local method, we apply the boundary data u_{PU}^G from the global problem on $\partial\Omega_L$ but enrich only global patches inside Ω_L^τ with the computed local solution. We refer to the area between $\partial\Omega_L$ and $\partial\Omega_L^\tau$ as a bufferzone with size $\tau := \operatorname{dist}(\partial\Omega_L, \partial\Omega_L^\tau)$.

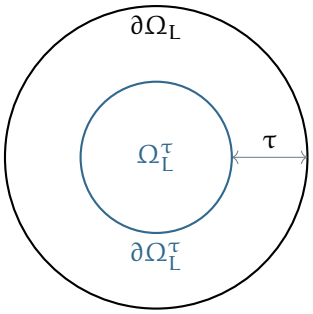


Figure 4.2: Domains of the local error estimate with the bufferzone f_τ.

For the analysis we consider the model problem

$$\begin{aligned} \operatorname{div} A\nabla u &= 0 \qquad \text{on } \Omega_L \\ u &= g \qquad \text{on } \partial\Omega_L \end{aligned} \tag{4.1}$$

on Ω_L, with smooth coefficient $A(x)$ and Dirichlet boundary conditions with varying boundary data. Let u^* be the solution of (4.1) with the true global solution as the boundary data, $g = u_G^*$. Let u^{gl} be the solution of (4.1) with the inaccurate boundary data $g = u_{PU}^G$, for example from the global numerical

solution, and let u_{PU}^{gl} be its discretization. Then the error of the computed enrichment u_{PU}^{gl} in the energy norm is bounded by

$$\begin{aligned}\left\|u^* - u_{PU}^{gl}\right\|_{E(\Omega_L^\tau)} \leqslant C_1 \inf_{v \in V^{PU}(\Omega_L)} \left\|u^{gl} - v\right\|_{E(\Omega_L)} \\ + \frac{C_2}{\tau}\left\|u^* - u^{gl}\right\|_{L^2(\Omega_L)},\end{aligned} \tag{4.2}$$

where we have the discretization error of the local problem and the L^2-error induced by the inexact boundary data divided by the bufferzone size τ on the right hand side. To improve accuracy of the global-local method, we thus have to balance local approximation power with the error in the boundary data. The latter heavily depends on the global solution, hence overall accuracy of the global-local method can actually be limited by the global discretization. Up to that point, accuracy can be improved by two techniques, though. First, one can apply multiple global-local iterations [49, 96, 141], thereby recomputing better enriched global solutions and thus improved local boundary data, but at a high runtime cost. Second, the term dependent on the boundary data can be reduced by increasing τ and thus the size of the bufferzone.

The use of a bufferzone was introduced in [49] for the global-local method, but similar approaches have been widely used in other methods, such as the Multiscale FEM [52] and prior GFEM research on computed basis functions [131]. Observe that in the setting of estimate (4.2) the position of the boundary of the local domain is fixed and we reduce the area where the computed enrichment is applied. In the global-local method, however, we would usually fix the area of application and increase the local domains size, thereby moving the position of its boundary, which changes the boundary data. In practice, we expect this to have even more benefits, since by applying a bufferzone we evaluate the global solution further away from e. g. a singularity, where it should be more accurate. Increasing the bufferzone, or reducing the area in which the computed enrichment is applied, has of course limits. The minimum size of the enriched area is one patch and may further be bounded from below by the enrichment radius required to resolve the local feature, as is the case for crack singularities (3.6). Conversely, heavily increasing the size of the local domain defeats the purpose of the global-local method.

A third technique to improve the quality of the computed enrichment based on the same inaccurate boundary data is the application of a different kind of boundary condition. In [9, 68], Robin boundary conditions were found to be an improvement over Dirichlet boundary conditions under some conditions. Specifically, Robin conditions depend on a parameter κ which has to be estimated correctly for optimal results. In Section 4.3.2, we investigate the effects of a bufferzone, multiple global-local iterations and different types of boundary conditions on accuracy in numerical experiments.

PERFORMANCE CONSIDERATIONS Performance-wise, global-local enrichments are only preferable if no analytic or precomputed enrichments are available for a problem. Given such a problem, let us first consider the case of a sequential simulation or a single local problem and thus ignore the parallelization opportunity in the global-local method for the moment. Compared to using

h-refinement on the global domain, the global-local method trades a reduced number of global degrees of freedom for additional solution and numerical integration steps at the same or comparable global accuracy. For the global-local method to have a performance advantage, the reduced number of global degrees of freedom should therefore result in enough speedup of the employed global solver, to make up for the additional assembly and solution steps.

After the initial, potentially cheap global solution step, every iteration has four runtime dominating tasks: the global and local assembly of the linear system and their corresponding solution steps. When running multiple global-local iterations, some work may be saved, though. As the local approximation space does not change, only the boundary conditions have to be re-assembled on the local problem. If we use a direct solver, or an incomplete LU decomposition as preconditioner, on the local problem, we can cache also the factorized matrix. Consequently, we obtain multiple local solutions for virtually no additional cost. Unfortunately, we cannot avoid traversing the local integration cells once every global-local iteration when assembling the newly enriched global system, since the global-local enrichment changed. However, all entries not affected by the enrichment can be kept, which, depending on the relative sizes of global and local problems, may save some time. When using iterative solvers, we can at least use the last global solution as an initial guess. In this case, and if the last solution is the initial, unenriched solution, we have to pad the coefficient vector with zeroes for the enrichment coefficients. In [67], it was further proposed to build a preconditioner that combines the factorized initial unenriched stiffness matrix and the factorized matrix of enriched patches, while ignoring the interaction part.

Runtime measurements of the global-local method and a comparison with direct h-refinement can be found in Section 4.3. However, we provide a rough runtime estimate in the following. Let dof_g be the number of initial global degrees of freedom and let us assume there are $1 \leqslant n$ equally sized local problems with dof_l degrees of freedom each. We further assume that $1 \leqslant i$ global-local iterations are run and, for simplicity, that the enriched global problem again has dof_g degrees of freedom. For convenience, we neglect linear contributions like the assembly of the linear systems; however, the resulting estimate is almost the same if they were indeed modeled. Assuming a quadratically scaling solver, we can model the order of the runtime $\mathcal{O}_{\mathrm{gl}}$ of the global-local method as

$$\mathcal{O}_{\mathrm{gl}} = (1 + i)\,\mathrm{dof}_g^2 + i \cdot n \cdot \mathrm{dof}_l^2\,. \tag{4.3}$$

For simplicity, we assume that the directly h-refined global problem has the same number of degrees of freedom as the combined global and local problems and thus has runtime complexity

$$\mathcal{O}_{\mathrm{direct}} = (\mathrm{dof}_g + n \cdot \mathrm{dof}_l)^2. \tag{4.4}$$

In this simplified model, and by further assuming that we have less global-local iterations than local problems, i. e. $i \leqslant n$, we arrive at the implication

$$\mathrm{dof}_g \leqslant \frac{2n}{i}\,\mathrm{dof}_l \quad \Longrightarrow \quad \mathcal{O}_{\mathrm{gl}} \leqslant \mathcal{O}_{\mathrm{direct}}. \tag{4.5}$$

Besides reminding us that multiple global-local iterations are expensive, this estimate states that the number of degrees of freedom required to resolve the local features must be comparable to the number of remaining global degrees of freedom, in order for the global-local method to pay off. Changing the underlying complexity assumptions of (4.5) of course changes the estimate. For example, more expensive solvers, such as a Newton solver for nonlinear problems, benefit more from the reduced amount of global degrees of freedom in the global-local method [64, 65, 95]. In consequence, the global-local method could be preferable for less relative local degrees of freedom. Another upside of the global-local method, however, is enabling simulations on constrained hardware in the first place by reducing the size of the largest occurring linear system. In this case, we may not even be interested in the asymptotical complexity of the method.

This said, the most important feature of the global-local method with respect to runtime is the parallelization opportunity it provides when multiple local problems are used in a simulation. Here, multiple local problems can be solved completely independent of each other and communication is only necessary in the evaluation of the global solution for the boundary data and in the assembly of the global enriched linear system. As a result, it should be possible to introduce equally sized local features to a simulation, together with a constant number of processes for each, without increasing the total runtime. Put differently, the global-local method should show excellent weak scaling behavior with local features, which we verify numerically later in this chapter.

4.1.1 *Local Boundary Conditions*

On parts $\partial\Omega_L \cap \partial\Omega_G$ of the boundary of the local domain Ω_L that intersect the global boundary $\partial\Omega_G = \Gamma^D \dot\cup \Gamma^N$, we use the global boundary conditions on the local problem. On the remaining parts of the local boundary $\Gamma^L := \partial\Omega_L \setminus \partial\Omega$, we use the global solution u_G as boundary data to apply either Neumann, Dirichlet or Robin boundary conditions, the latter interpolating between the first two.

Assume we solve the weak formulation (2.19) of linear elasticity on the global domain, with Neumann boundary data $\bar t$ and Dirichlet boundary data $\bar u$. With Dirichlet boundary conditions on Γ^L the local weak form is then given by:

Find $u_L \in V_L^{PU}$ such that

$$\int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L)\,dx = \int_{\Gamma^N \cap \partial\Omega_L} \bar t v_L\,ds + \int_{\Omega_L} b\,v_L\,dx \tag{4.6}$$

for all test functions $v_L \in V_L^{PU}$. The Dirichlet boundary conditions on Γ^L and Γ^D with data u_G and $\bar u$ respectively are treated as described in Section 2.3.4.

Based on a parameter $0 \leqslant \kappa$, Robin boundary conditions interpolate between Neumann and Dirichlet boundary conditions. For $\kappa = 0$ we obtain Neumann boundary conditions, while $\kappa \to \infty$ yields Dirichlet boundary conditions, similar to a penalty method. Robin conditions are given by

$$\mathcal{T}(u_L) + \kappa u_L = \mathcal{T}(u_G) + \kappa u_G \qquad \text{on } \Gamma_L. \tag{4.7}$$

where $\mathcal{T}(u) = \sigma(u)\cdot n$ is the prescribed traction. The corresponding weak form then reads:

Find $u_L \in V_L^{PU}$ such that

$$\begin{aligned} &\int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L)\,dx + \kappa \int_{\Gamma_L} u_L v_L\,ds \\ &= \int_{\Gamma^N \cap \partial\Omega_L} \bar{t} v_L\,ds + \int_{\Gamma_L} \Big(\mathcal{T}(u_G) + \kappa u_G\Big) v_L\,ds + \int_{\Omega_L} b\, v_L\,dx. \end{aligned} \tag{4.8}$$

for all test functions $v_L \in V_L^{PU}$.

Research on global-local enrichments in the GFEM context [68] finds that Robin boundary conditions perform the best for a reasonably wide range of the parameter κ. Furthermore, studies on non-overlapping Schwarz methods, e. g. [29], suggest that Robin conditions can improve accuracy, especially with no or only small buffer zones. All this depends on finding the right parameter κ, though. To this end, Kim *et al.* [68] propose estimating

$$\kappa \approx \frac{E}{h} \tag{4.9}$$

by Young's modulus E divided by the characteristic length of a global finite element h along the local boundary. In this case, κ scales with the material parameters and the global-level, yet ignores other features of the problem at hand, such as the local stress.

In [9], we argued instead that in the linear system assembled from (4.8), the right hand side vector contributions from $\mathcal{T}(u_G)v_L$ and $\kappa u_G v_L$ should be of comparable size. This way we aim to be around halfway between Neumann and Dirichlet boundary conditions. Hence, we assemble the vectors

$$\widehat{T} = \left(\int_{\Gamma_L} \mathcal{T}(u_G)\varphi_i \vartheta_i^n\,ds \right)_{i,n} \quad \text{and} \quad \widehat{K} = \left(\int_{\Gamma_L} u_G \varphi_i \vartheta_i^n\,ds \right)_{i,n} \tag{4.10}$$

for all basis functions that span V_L^{PU} and set κ to be the fraction of their discrete l^2-norms

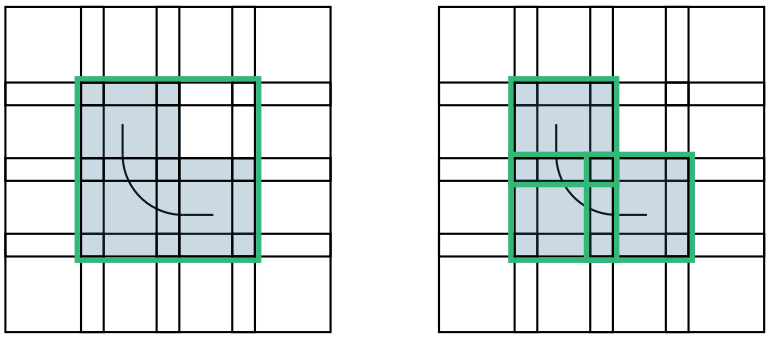
$$\kappa := \frac{\|\widehat{T}\|_{l^2}}{\|\widehat{K}\|_{l^2}}. \tag{4.11}$$

In this scheme, κ scales with the material parameters and with features of the gradient of the solution we are seeking, but not with the global discretization level. In Section 4.3.2 we study the effect of different kinds of boundary conditions and evaluate the above scheme (4.11) to compute κ numerically, now in three space dimensions. While not tested yet, the above scheme could also be used to compute a separate value of κ for each patch at the local boundary, which would further adapt the boundary condition to the local stress distribution.

4.1.2 *Local Problem Domain*

The smallest entity we can enrich in a partition of unity method is a single patch. Any local problem thus has to cover all global patches on which its solution is used as an enrichment. For a given feature on the global domain, we can

thus either create a single local domain covering the whole feature or multiple overlapping local problems, with a maximum of one per global patch affected by the feature, compare Figure 4.3.



(a) Local problem per feature.

(b) Local problem per enriched patch (LPP).

Figure 4.3: Three crack-intersecting global patches with either a single local problem covering all patches (a) or one local problem for each (b). In the latter case, local boundaries overlap with the crack, but more local problems are generated, which are also better adapted to the geometry of the local feature.

Both approaches have their tradeoffs, especially with respect to accuracy and performance, as well as implications for the parallelization of their implementation. Also note that the difference between the approaches depends on the global discretization, i. e. the number of global patches at a feature: The coarser the global discretization with respect to the size of the feature of interest, the less difference there is between the two approaches. Now, large simulations can easily be limited in their global patch width while still benefitting from the resolution of small scale features. This is demonstrated for example in [97], where a localized heat source smaller than a global patch constitutes a local problem. The approach of one local problem per enriched global patch (LPP) illustrated in Figure 4.3b has been implemented and studied in the GFEM^{gl} [23, 49, 66]. However, in this thesis, we decided to implement the single local problem per feature version shown in Figure 4.3a due to the reasons outlined in the following.

Estimate (4.2) shows that the accuracy of the global-local method is determined by the local discretization and the quality of the boundary data on the local problem. If we spend the same number of degrees of freedom on a local problem as we would using adaptive h-refinement directly, we expect a comparable resolution of the local feature at least after some number of global-local iterations. Multiple smaller local problems suffer from the fact that their boundaries are closer to the initially unresolved feature or might even overlap with it, as can be observed in Figure 4.3b. For example, the singularity at a crack front has an effect within some radius around the front (3.6), as proven in [46], inside which a coarse global solution and thus the local boundary data cannot be accurate. Therefore, for multiple overlapping local problems, the boundary data provided by the global solution is worse, and we expect more global-local iterations to be required until we reach the same accuracy as with a single, larger local problem. In Section 4.3.1, we confirm this in a numerical experiment.

More global-local iterations then also imply higher runtime for the same approximation quality. On the other hand, solving multiple small problems might be cheaper than solving one large local problem due to the smaller sizes of the involved linear systems. Note, however, that due to the overlap of local problems in LPP, this approach uses more degrees of freedom overall to resolve the local feature than does a single local problem. Decisions in that regard likely have to be made on a per simulation basis, based on numerical experience, especially since the two approaches further have different parallelization characteristics.

Multiple local problems, for one or several features in the domain, provide an excellent parallelization opportunity, as they can be solved completely independent of each other. Synchronization is only required in the evaluation of the global boundary data and in the assembly of the global enriched linear system. In the GFEM^{gl}, a parallelization strategy of the global-local method is presented in [66] where local problems are solved parallel to each other, but each on a single thread. Since one local problem for every global, enriched patch is created there, they are small and thus cheap to solve even sequentially. Furthermore, in most examples presented in the GFEM^{gl} context, the global discretization is fine enough to generate enough parallel work, while coarser global discretizations would severely limit the parallelization opportunity. In contrast, having a single local problem per feature lends itself to the parallel solution of each problem, while of course problems for different features should still be solved simultaneously. Another difference between our and the implementation presented in [66] is that we chose to implement the global-local method in distributed memory using the message passing interface (MPI). While the GFEM^{gl} implementation can of course be extended to support local problems on different threads for each global process region, additional communication will be needed for overlapping local problems on different global processes. In Section 4.2 we present a distributed memory parallel implementation of the global-local method with non-overlapping local problems that also exploits the inherent parallelism of the method. This implementation shows excellent scaling characteristics, as we demonstrate in numerical experiments in Section 4.3.3.

Another difficulty in the implementation of overlapping local problems in our specific meshfree PUM arises from the numerical integration of the computed enrichment for the enriched global linear system. Here, integration cells have to be used that at least resolve the patches of the local spaces in order to integrate their basis functions accurately. To this end, and to exploit the parallelism of the local problems, it is useful to directly use the integration cells of the local problems, as was also proposed in [66]. The interaction of which integration cells are used and the parallelism of the resulting method is explained in more detail in Section 4.2. To be able to use the local problems' cells efficiently, it is beneficial to have the local cells nested in the global ones. Nested cells automatically resolve the global patches and simplify the generation of exactly matching cells in the overlap of local problems. In the GFEM [66], the latter is achieved by a third intermediate mesh layer common to all local problems which defines their elements. However, while constructing nested integration cells is very easy in FEM and mesh based methods, compare Figure 4.8a, it is very difficult in our meshfree, flat-top PUM, at least with the cover construction detailed in Section 2.3.5, compare Figure 4.8b. Finding a different construction of a flat-top cover with the property of nested refinements might be possible of course, but

it is likely to have other downsides. Hence, solving overlapping local problems is currently not possible in an efficient manner in a parallel setting in PUMA.

IMPLEMENTATION IN PUMA The smallest entity we can enrich in the PUM is a single patch. We therefore have to guarantee that a constructed local domain in the global-local method contains all global patches we want to enrich with the computed enrichment. In PUMA we base the construction on the *Feature* class, whose derived classes represent features of the geometry, such as a crack or only the crack front, or a whole box for e. g. topological enrichment across refinement levels. *Feature* objects are then used to automatically construct local domains and their respective boundary parts, and subsequently to mark all overlapped global patches for enrichment with the computed solution. Specifically, for the setup of the local domain, we query the function space for all patches whose stretched domains intersect the feature. The bounding box of those patches then is the local domain. In parallel simulations, we further have to find and communicate the maximal bounding box across all involved processes, since a function space always knows about its local patches only. This construction comes with the disadvantage that the local domain potentially covers global patches that are not going to be enriched, compare Figure 4.4. As a result, the computational costs to compute the enrichment are increased, which could be avoided if the local geometry was given by the union of the enriched, global patches. Effects on runtime and accuracy of such a domain construction would have to be studied.

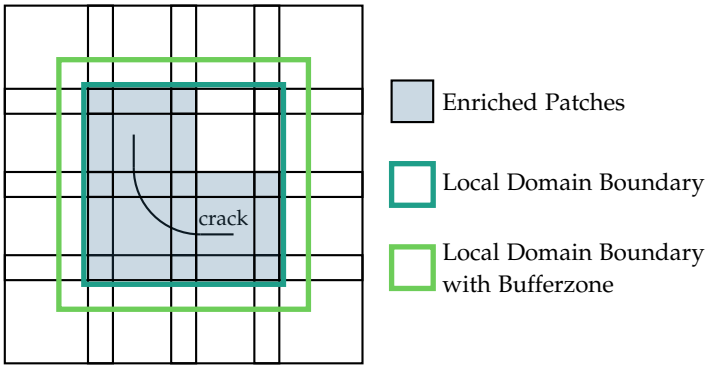


Figure 4.4: Local domain with and without bufferzone around crack. Only global patches intersecting the crack are enriched, i. e. we use geometric enrichment in this example.

One technique to improve accuracy of the global-local method is applying a bufferzone to the local domain, which means enlarging the domain we are computing on while keeping the enriched area constant. To this end, the user can provide a buffer factor $0 < \tau$ to increase the size of the local domain by $100 \cdot f_\tau$ percent. Here, a stretch based on the smallest dimension of the local domain's bounding box is applied, thereby making a rectangular domain more square shaped, for the following reason. As described in Section 2.3.2, our standard polynomial basis functions are tensor products of one dimensional polynomials. If patches are not quadratic, the derivative of those polynomials is therefore of different magnitude depending on the direction, which deteriorates the condition of the stiffness matrix. Thus, by default, we base cover construction

in PUMA on a quadratic version of the domain's bounding box, leading to potentially fewer patches in thinner directions of the domain, though, which might not be desirable either. Applying a buffer factor based on the smallest domain dimension mitigates these issues. It further ensures that we move the local domain's boundary away uniformly from, for example, the crack singularity. This is also more similar to the approach used in the FEM based GFEM^{g1}, where layers of elements are simply added to the local domain [49]. However, applying a bufferzone as described above in PUMA comes with the side effect of coarsening the local discretization if the local domain level is kept constant, because we increase the size of the local domain. To keep the local resolution constant, we therefore had to set the local domain's bounding box size manually in experiments comparing the effects of different bufferzone sizes, see Section 4.3.2. Moreover, we actually always have to apply at least a small bufferzone in the PUM. Otherwise, would evaluate our global solution on patch boundaries, where the derivative of our weight functions and thus of our basis functions is ill-defined. By default we apply a factor of $f_\tau = 0.01$ in PUMA.

As geometry and domain features in the PUM are not encoded in a mesh and thus have to be stored separately, effort was made to not duplicate the information while implementing the global-local method in PUMA in order to improve usability and reduce resource usage. This was the case for the *Domain* class, which stores subdomains, such as cracks, and makes references to parts of the geometries boundary, as well as for the *Geometry* class, which represents the geometry. In both instances wrappers were implemented for the local domain and geometry that simply pass most queries to the global domain and geometry. The *DomainWrapper* stores the local geometry wrapper, for bounding box queries or access to the geometry and holds a reference to the global domain to which it passes all functions related to subdomains. Subdomains, such as cracks, added to the global domain are therefore visible on the local domain both instantly and under the same name, which makes applying boundary conditions and writing GECO forms easier. The *GeometryLocalDomain* stores the local geometry, which is always an axis aligned box, and a reference to the global geometry. All queries on the local geometry are then first restricted to the local domain's box and further passed to the global geometry.

4.2 PARALLELIZATION AND INTEGRATION

The global-local method is inherently parallel, since local problems can be solved completely independent of each other. As explained in the beginning of this chapter, we chose to solve each local problem in parallel using MPI. Hence, we have to restrict local problems to subsets of all available processes in order to exploit the parallelism of the global-local method. In MPI language, we use *MPI_Comm_split* to split the global world communicator *MPI_COMM_WORLD* into several communicators, so that each then contains a subset of the total processes. Such a split communicator is identified by a so-called color. For the implementation this poses the additional challenge, that we have to communicate between different subsets of the overall available processes. In the following, we first explain the necessary parts of the parallelization in PUMA as implemented prior to this thesis and continue with the changes we made to support local

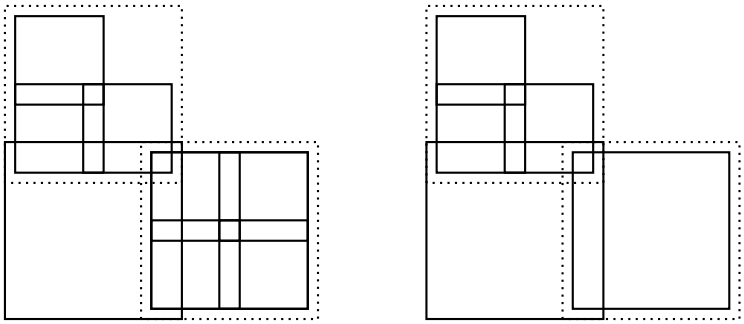
problems on split communicators. Though intertwined with the parallelization strategy, we present details concerning the integration cells thereafter.

In PUMA, parallelization is achieved by distributing patches. All n patches on a given level are ordered along a space-filling curve, see Section 2.3.5, and each of the p processes receives its batch of n/p patches. This way, the stiffness matrix is automatically distributed by rows, with each matrix entry being a block corresponding to the interaction of the local basis functions ϑ_i^n and ϑ_j^m of two patches ω_i and ω_j. Note that the computational load associated with each patch during assembly of the linear system depends on the number of integration cells and points associated with it. That is, the load of each patch depends on the number, complexity and required integration resolutions of the polynomial basis functions and enrichments applied on it and its neighbors. Moreover, adaptive h-refinement can increase the computational cost of coarser neighboring patches, as more integration cells are generated in the overlap. In such a case, higher integration resolutions might also be required due to the shape of the PU functions in the overlap of differently sized patches Enrichments, unsurprisingly, can significantly increase the relative computational cost of individual patches and also that of their neighbors. Currently, these effects are not accounted for in PUMA, which affects certain decisions and also results of our numerical experiments in Section 4.3. However, space-filling curves are well suited for this kind of load balancing problem [40] by adapting the count of each patch along the curve based on a load estimate. Achieving good load balancing should thus be possible in this approach, but is left for future work, as it was not the focus of this thesis.

As patches overlap and the stiffness matrix contains blocks corresponding to the interaction of neighboring patches, a process needs to be aware of neighboring patches managed by other processes, in order to evaluate their basis functions during assembly of the linear system. We call those remote patches, a term always relative to the current process. Note that the neighbor relations of patches needs to be computed only once per given set of function spaces on a constant discretization level per simulation and is negligible in runtime. Regarding the integration cells, we have two options during assembly of the stiffness matrix: either every process has all integration cells required for its patches, the default in PUMA, or integration cells of different processes are disjoint, called disjoint integration in PUMA. In the former case, we communicate integration cells before, in the latter, we communicate matrix entries after assembly.

When using enrichments computed by PUMA, the global function space has to evaluate the basis of the local function space during the assembly of the global enriched linear system. The local function space, in turn, has to evaluate the basis of the global space when applying the global boundary data. In parallel, both directions are complicated by the fact that overlapping regions of the global and local domain must not belong to the same process in parallel. We hence need to inform the incorporated function spaces about their new process as well as remote patch neighbors. This operation was already implemented in PUMA prior to this thesis, but it had to be adapted to allow local problems on split communicators. We outline the respective algorithm in the following.

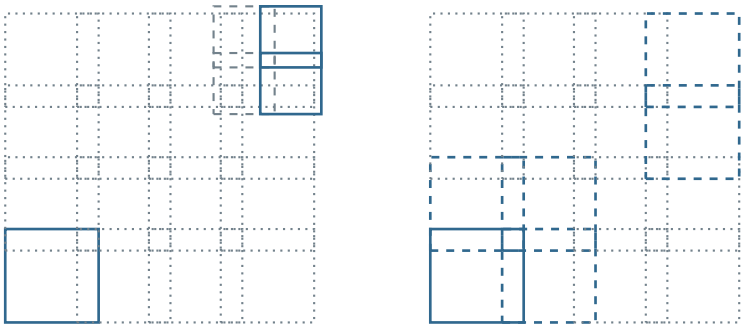
Computing neighboring processes between two covers is based on the notion of a minimal cover. Essentially, we keep the stretched domains of the tree leaves, but merge those where all siblings belong to the same process, compare Fig-



(a) Patches on some process. (b) Corresponding miminal cover.

Figure 4.5: Sketch of minimal cover for local patches on one MPI process. Dotted lines indicate the size of the parent patches for the refined ones.

ure 4.5. In effect, for each process we save the smallest possible set of cover boxes such that all children belong to the process, but account for the diminishing effect of the applied stretch factor on higher discretization levels. Intersections of the minimal covers then determine the neighbor relations between different covers. To allow evaluation of remote patches during integration without communication, patches marked as remote neighbors are then sent to the current process along with the corresponding matrix and vector blocks, see Figure 4.6 for an example. After this setup step, every process can evaluate all required patches of the global and local function space on its integration cells without communication.



(a) Global and local patches on process zero. (b) Remote global patches on process zero (dashed).

Figure 4.6: (a): Global cover with one patch on process zero and local cover with two patches on process zero. (b): Global patches sent to process zero (dashed) due to remote neighborship relations with itself and the local cover.

SPLIT PARALLEL PUMA Since only the communicator changes, only solving the local problem without the global boundary data on a split communicator was essentially possible prior to this thesis. While certain pieces of code had to be adapted to not assume the world communicator, most of PUMA already just passed on the used communicator. Constructing the local domain and having the involved function spaces communicate for the boundary data and for

evaluating the computed enrichment on the global space required some changes, though. To illustrate this, Figure 4.7 sketches one possible scenario with two local problems on two processes each, while the global problem is solved with sixteen processes. For simplicity, we just sketched the process regions, not the cover patches. Note that for example process zero now is a neighbor of process eight, which was not the case on the global problem alone.

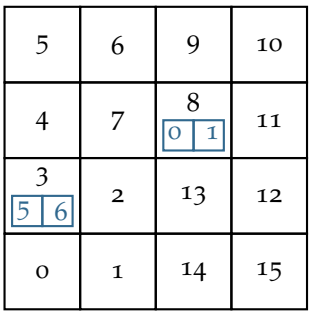


Figure 4.7: Sketch of level two patch distribution across sixteen processes, with two local problems solved on two processes each. Processes numbered with respect to global communicator. Split parallel solution of local problems introduces new process neighborhood relations on both the global and local cover.

Exchanging the remote neighbors between two covers on different communicators requires an additional communication step, which is implemented in PUMA under the assumption that at least one cover is distributed on the world communicator. This assumption implies that the global problem is always solved with all available processes. It further excludes overlapping local problems, a scenario, where assembly of the global linear system is difficult to implement efficiently in the meshfree PUM anyway, as we discuss shortly. Before we are able to compute intersections of minimal covers, the first step is to index the local minimal covers relative to the world communicator. In MPI, processes are index relative to a communicator, where the index is referred to as the rank of the process. The same process can thus have different ranks in the world and in a split communicator. Mapping ranks of the split communicators into the world communicator is done using the MPI *translate_ranks* function, which operates on the process group of a communicator. Hence, after translating the local minimal covers into the world ranks, we perform one global communication, gathering all local contributions into a global minimal cover. From here, everything works as before, that is, remote neighbors are computed and remote neighboring patches of each cover are gathered on the current process together with corresponding matrix and vector blocks. Note that this step has to be performed only once in the setup phase of a simulation.

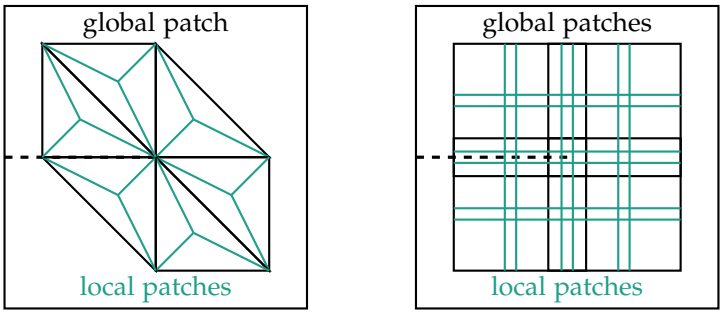
Another implementation difficulty that arises is that a global patch on a particular process may now have an enrichment whose communicator, however, does not contain that process. Figure 4.7 shows an example where process eight is enriched with a solution computed on process zero and one. This has several implications: For one, we can no longer integrate the computed enrichment on the global integration cells. As described in the next subsection, we solve this issue by integrating on the cells of the local function spaces. Moreover, we have to inform the enriched processes at least about the number of degrees

of freedom added by the enrichment, so that blocks of the linear system have the correct size. In addition, the construction of the local domain has to be adapted. For this, we perform the algorithm described in Section 4.1.2 on the world communicator for every feature we want to enrich and then let each color construct only its own local domain. Finally, the implementation of input and output, for instance for visualization, becomes more complicated, as data has to be coordinated between subsets of the overall processes.

Despite the implementation challenges, the proposed parallel implementation of the global-local method should show excellent scaling behavior, which we confirm numerically in Section 4.3.3.

GL INTEGRATION In the following, we are concerned with the numerical integration of the enriched global problem in the global-local method. Specifically, the issue here is the evaluation and integration of the computed enrichment that is defined on another PUM-space with its own parallel distribution. We lay out the challenges during this step, especially when implemented in a meshfree, flat-top partition of unity method, and describe their solution as implemented in PUMA for the thesis at hand.

When several function spaces are involved during the assembly of a linear system, it is crucial to resolve patch boundaries and so-called split points of all function spaces in the integration cells in order to improve integration accuracy. The latter are points along which integration cells are split in all two or three dimensions, to resolve discontinuities of incorporated basis functions, compare Figure 2.11. Achieving this is trivial in FEM based partition of unity methods, because refined cells can easily be nested inside coarser cells. In the PUM, however, patch boundaries on finer discretization levels do not coincide with patch boundaries on coarser levels at all, as can be seen in Figure 4.8b. Thus, patches of different function spaces have to be resolved explicitly during the generation of integration cells.

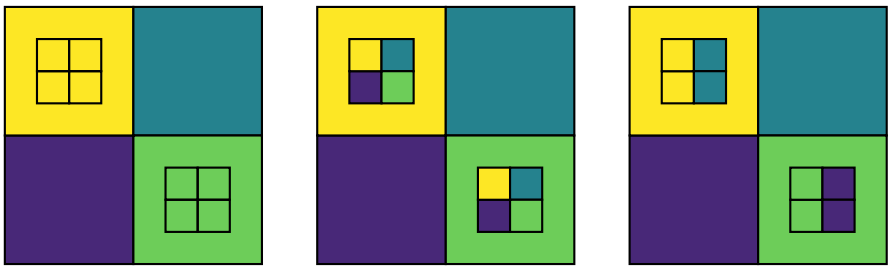


(a) Refined patches in GFEM^{gl}. (b) Refined patches in PUM.

Figure 4.8: Global and local patches around a crack tip in FEM based (a) and our flat-top partition of unity method (b). Nesting refined patches in the latter would be difficult, which has implications for the numerical integration in the global-local method.

By integrating the computed enrichment on the global cells with resolved local patches, we evaluate the enrichment only with the global processes that overlap the local domain. That is, in worst case, all local integration cells are

evaluated by a single process, compare Figure 4.9a. The solution to this issue is using the local integration cells and thus the local process distribution also in the global assembly step, as depicted in Figure 4.9b.



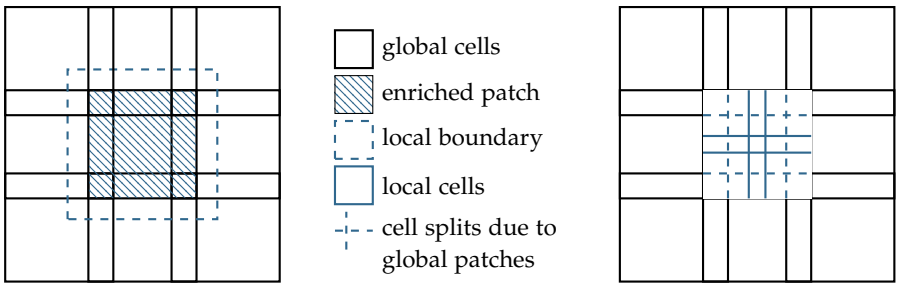
(a) Integration on global cells.

(b) Integration on local cells.

(c) Split integration on local cells.

Figure 4.9: Simplified sketch of global and local level one covers distributed across four processes, with the following cells used as global integration cells. Global cells with resolved local patch domains (a), local cells with resolved global patch domains distributed across all processes (b) and distributed only across subsets of the total number of processes, using split communicators (c). The latter allows parallel solution of local problems.

To this end, we implemented the *IntegrationcellHandlerMultiCover* for this thesis, which takes one global integration cell handler and a list of local integration cell handlers, and automatically manages everything else in all functions that accept an integration cell handler in PUMA. The constructor further takes a list of *Feature* objects specifying the global patches to be handled by each local integration cell handler. We skip all cells for enriched global patches in the global integration cell handler and all local cells outside the enriched patches on the local handlers, such that they are disjoint and the union of all integration cells is the global computational domain, compare Figure 4.10c.



(a) Global patches and local problem.

(c) Integration cells of global problem using local cells.

Figure 4.10: Using local cells during numerical integration of the global linear system. In this sketch, the local problem is discretized on level one and additional cells are introduced to resolve the global patches in the local integration cells.

Earlier we mentioned the two strategies of either sending cells or matrix entries during parallel assembly, which are called default and disjoint integration in PUMA. Note that when using the integration cells of the local spaces, it is

no longer desirable to send cells. When considering a local problem on a split communicator, it is even impossible without complicated communication across different communicators. Hence, disjoint integration has to be used in our implementation of the global-local method. To illustrate this, consider a local problem completely contained inside the region of process zero. If the local problem is parallelized using the world communicator, we would lose the improved patch and cell distribution of the local problem as we send all local patches to process zero. If further the local problem is defined on a split communicator that does not contain process zero, we cannot even send the cells using the local communicator. The split scenario is also visualized in Figure 4.7, but with process eight instead of zero.

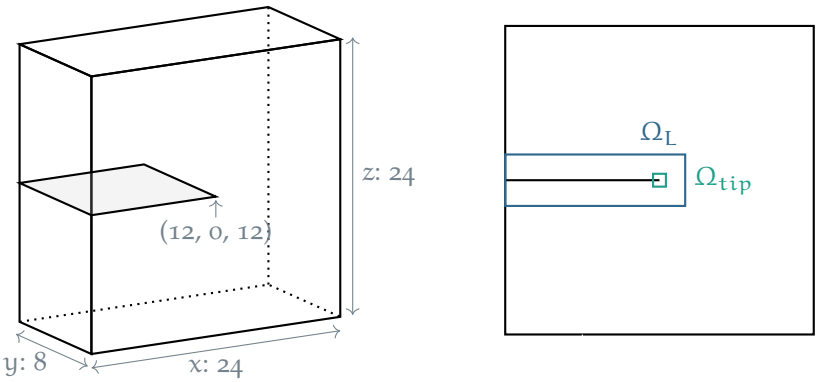
4.3 NUMERICAL RESULTS

In some sense, the global-local method is only a more complicated version of adaptive h-refinement on a separate discretization. Compared to the latter, it potentially trades some accuracy, due to imperfect boundary data, for performance, due to the reduced size of the global problem and the parallelization opportunity. The aim of this section is thus to investigate this tradeoff numerically. Specifically, we conduct the following experiments. First, we examine if the global-local method achieves comparable accuracy to a direct solution. Here, we consider problems where we know the solution analytically. Second, we investigate the effectiveness of multiple iterations, of a bufferzone, and of Robin boundary conditions in improving the boundary data of the local problem and thus the computed enrichment. Third, we turn to the global-local method's inherent parallelization opportunity, and measure parallel scaling with multiple local problems, specifically testing our proposed distribution of problems across split communicators. Lastly, we compare the achieved runtime performance to a directly h-refined simulation and thereby evaluate our runtime estimate (4.5).

We assess the accuracy of the global-local method in two examples in which we have a reference given analytically. For the first example, we use the method of manufactured solutions and define the following problem.

Example 4.1 (Validation Problem)**.** *We study a cracked extruded square domain with a simple flat crack geometry, see Figure 4.11a. The domain is made of a linear elastic and isotropic material with Poisson's ratio* $\nu = 1/3$*, while we vary Young's modulus per experiment and report it there.*

We prescribe Dirichlet boundary conditions on all surfaces except the crack, so that the solution $u^* = u_I + u_{III}$ *is the sum of the first terms of the mode I and mode III displacement fields* (2.28) *with respect to cylindrical coordinates around the crack front, compare Section 2.2. As a result, we can compute errors with respect to the true solution and know the exact stress intensity factors for this problem, which are* $K_I = \sqrt{2\pi}2G$*,* $K_{II} = 0$ *and* $K_{III} = \sqrt{2\pi}G/2$*. We further define a small box* Ω_{tip} *with radius* 0.5 *around the crack front for local error measurement. The global and local problem are discretized on level four and five respectively, resulting in the local problem being two levels finer around the crack. Basis functions consist of linear polynomials and the crack enrichments described in Section 3.1, with the crack front enrichments applied on patches in a radius of one around the crack front.*



(a) Geometry of validation problem.

(b) Side view of local and error extraction domains.

Figure 4.11: Cracked extruded square domain for validation purposes with local domain Ω_L around the whole crack. Furthermore, an error extraction domain Ω_{tip} is used to measure the error locally at the crack front.

We compute the errors on subsets $\Omega^* \subset \Omega$ of the computational domain and use either the L^2-norm

$$\|u\|_{L^2(\Omega^*)} = \left(\int_{\Omega_*} |u|^2 \, dx \right)^{\frac{1}{2}}, \tag{4.12}$$

or the H^1-seminorm, for simplicity referred to as just the H^1-norm,

$$|u|_{H^1(\Omega^*)} = \left(\sum_{|\beta|=1} \int_{\Omega_*} |D^\beta u|^2 \, dx \right)^{\frac{1}{2}}, \tag{4.13}$$

where β is a multi-index, and $D^\beta u$ the β-partial derivative of u. If not stated otherwise, we report the relative error

$$e(\Omega_*) \coloneqq \frac{\|u^* - u_{PU}\|_{\Omega_*}}{\|u^*\|_{\Omega_*}} \tag{4.14}$$

of the computed approximation u_{PU} with respect to the true solution u^*.

As a second example, with more complex crack geometry, we revisit the inclined elliptic crack example introduced in Section 3.3.2. In this example, the stress intensity factors around the crack front are available analytically and we can compute the error (3.12) relative to them.

4.3.1 *Accuracy*

In this section, we validate the global-local method by investigating if it achieves comparable accuracy to a direct, h-refined solution. To this end, we study the validation Example 4.1 with Young's modulus $E = 1$, such that all three stress

intensity factors are of the same order of magnitude. To compare the accuracy of the global-local method against a direct solution, we solve the problem once on the global domain only and once with a local problem set up around the whole crack, compare Figure 4.11b. In the first case, we use the crack enrichments directly on the global problem and in the latter, on the local problem only. Thus, the crack is not modeled in the initial solution step of the global-local method. We apply adaptive h-refinement around the crack in the direct solution, to obtain the same number of degrees of freedom for approximation as in the local problem, compare Table 4.1.

Table 4.1: Degrees of freedom (DOF) and final H^1-errors for the validation problem for a global-local (GL) and a direct solution.

	DOF_g	DOF_l	$e(\Omega)$	$e(\Omega_L)$	$e(\Omega_{tip})$
GL	1.23×10^4	6.24×10^4	8.16×10^{-2}	8.71×10^{-2}	2.56×10^{-2}
Direct	7.72×10^4		8.15×10^{-2}	8.55×10^{-2}	1.93×10^{-2}

To assess the quality of the computed solutions, we compare norms integrated on various subdomains of Ω. In Figure 4.12 we show relative H^1-and L^2-errors on the full domain Ω, on the area of the local domain Ω_L and in the box Ω_{tip} around the crack front. We plot both the error of the global-local method in each iteration as well as the error of the direct simulation, the latter indicated by a dashed, horizontal line. The first thing to note here is that the global-local solution seems to converge to an error level determined by the approximation power of the global and local ansatz spaces. Also, it does not oscillate over iterations. The error of the initial unenriched global solution is rather large as the crack is not yet modeled. After about four iterations, however, the global-local solution reaches about the same accuracy as the direct solution. The direct solution has more global degrees of freedom around the crack front to adjust coefficients for the enrichment, which explains why we see slightly better accuracy around the crack front in this case. The fact that we need four iterations to achieve comparable accuracy motivates the next section, where we study methods to improve the approximation in the first iterations.

It is interesting to note, though, that both approaches have higher relative L^2-error close to the crack front than in the rest of the domain, whereas the opposite is true in the H^1-error. This can be explained as follows: In the application of the enrichment basis function, the linear solver adjusts their coefficients such that the global PDE-energy is minimized. The H^1-seminorm is closer to that energy as it too is derivative based. Small variations in the coefficient of the enrichment to adjust the impact in energy norm can have bigger impacts in L^2-norm, however. Higher L^2-errors then usually occur along the boundary between the enriched and unenriched parts of the domain, as also happens in adaptive h-refinement along jumps of the applied refinement levels. This effect vanishes if we apply the enrichments globally on all patches.

Next, we further investigate the accuracy of the global-local method in extracted stress intensity factors in the validation Example 4.1. Figure 4.13 and Table 4.2 show the relative errors of the extracted stress intensity factors per global-local iteration. Again, we add results from a direct solution for comparison. We extract in a radius of 2.50×10^{-1} around the center of the crack front.

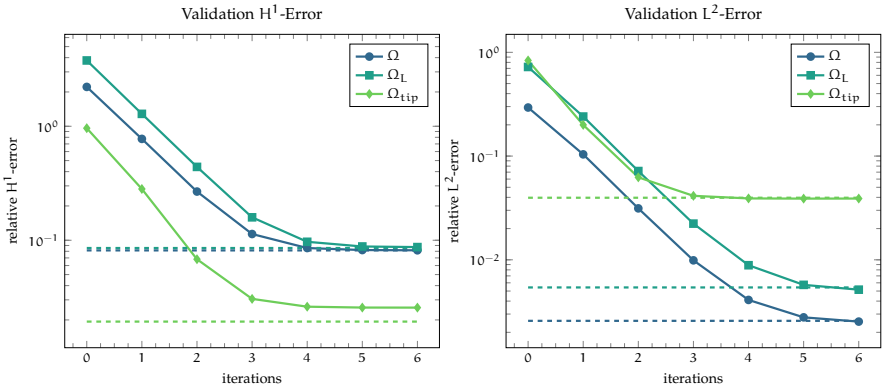


Figure 4.12: Global errors of validation example on different parts of the domain, per global-local iteration. Results of a direct global solution are indicated by dashed lines.

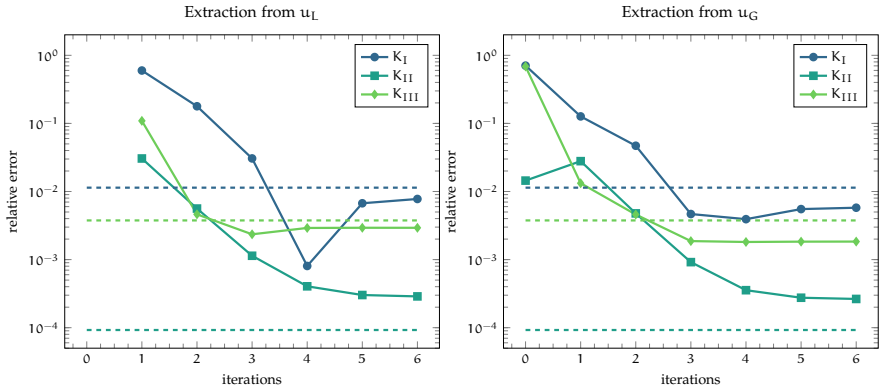


Figure 4.13: Relative error of extracted stress intensity factors per iteration. As the analytic mode II factor is zero, we report the absolute error here. Extraction from either the local solution, i. e. the computed enrichment (left) or the enriched global solution (right) when using the global-local method. Results obtained from a direct solution (dashed) for comparison.

As expected, we get comparable values from the global-local solution and the direct solution, at least after some iterations. We further compare extraction from the global solution as well as from the local solution, both in the global-local method. Here, extracting directly from the local solution, i. e. the computed enrichment, yields the same levels of accuracy. We can also observe that the imperfect boundary data can also erroneously lead to small errors in stress intensity factors, as happens in mode I in the fourth iteration when extracted from the local solution. While we do not expect this in volume measures such as the H^1- and L^2-norm, a single crack opening mode can be predicted well even from wrong boundary data.

We conclude that moving the vast majority of degrees of freedom onto the local problem produces comparable results as using them directly on the global problem, both in the computed displacement as well as in the extracted stress intensity factors. In the validation example, the global problem then has only sixteen percent the degrees of freedom as the direct solution, compare Table 4.1.

Table 4.2: Errors of stress intensity factors in the validation problem in the sixth global-local (GL) iteration, extracted either from the local solution (u_L) or the enriched global solution (u_G). Results of a direct solution for comparison.

	K_I	K_{II}	K_{III}
GL (u_L)	7.77×10^{-3}	2.88×10^{-4}	2.93×10^{-3}
GL (u_G)	5.78×10^{-3}	2.64×10^{-4}	1.84×10^{-3}
Direct	1.14×10^{-2}	9.24×10^{-5}	3.76×10^{-3}

However, due to imperfect boundary data on the local problem, we need several global-local iterations to achieve that accuracy. When interested only in the stress intensity factors, our results further suggest that it is possible to eliminate the final global solution step by extracting directly from the local solution, thus removing the cost of one global-local iteration. Also important to note here is that the global-local solution seems to converge after some iterations and does not oscillate, although no prove of this exists yet.

INCLINED ELLIPTICAL CRACK Next, investigate the accuracy of the global-local method on a problem with a more complex crack geometry. To this end, we revisit the inclined elliptic crack introduced in Section 3.3.2. In this example, the stress intensity factors around the crack front are available analytically. We extract them at each crack front node via the DCM from three different numerical solutions. Again, we compare a direct solution to a global-local solution with one local problem around the entire crack. We apply h-refinement such that both solutions have a comparable number of degrees of freedom and thus a comparable discretization around the crack front, compare Table 4.3.

As discussed in Section 4.1.2, in the GFEM^{gl}-context a local problem is set up for every enriched global problem, which leads to more as well as overlapping local problems. While this emphasizes the parallelization opportunity of the global-local method, it also has implications for the accuracy of the method. The local boundaries overlap the crack discontinuities and singularities, thereby deteriorating the quality of the global boundary data. On that account, we additionally report the results of a second global-local solution, where we define one local problem for each of the eight global, crack-intersecting patches. We refer to this setting as 'local per patch' (LPP) while the single local problem version is referred to as the global-local (GL) solution.

Regarding simulation parameters, we discretize the global problem on level four and start the single local problem (GL) on level three. As we have exactly eight global, enriched patches, we match this resolution by starting with local level two in the LPP case and refine patches in the local region twice in the direct simulation. From there, we first refine all local patches intersecting the crack front thrice and all resulting local patches once more in all three cases. As a result, the finest patches in all three simulations are of the size of a global level ten discretization. Since the crack is rather close to the boundaries of those eight enriched patches, we apply a bufferzone of ten percent, $f_\tau = 0.1$, on all local problems. In consequence, we obtain roughly the same refinement around the crack in all three simulations. Only the direct global discretization has a slightly higher resolution around the crack front due to the stretch of

the local domains caused by the application of the bufferzone. As usual, we enrich all patches intersecting the crack but not the front with the Heaviside enrichment (3.1) and all patches within a radius around the crack front with the front enrichments (3.3). Geometrical front enrichment is possible in this case as the crack is flat and we choose twice the minimal length of the crack front segments as radius, which is 8×10^{-3}. The resulting degrees of freedom and number of enriched patches are presented in Table 4.3. Note, however, that the numbers in the LPP case appear higher as the local problems overlap. We also allowed a level difference of two between neighboring patches in the direct solution to limit the spread refinement.

Table 4.3: Number of degrees of freedom (DOF) and enriched patches with either the front enrichment or the Heaviside enrichment for the inclined elliptical crack example. In the case of one local problem per global enriched patch (LPP), the sum over all eight local problems is presented.

	DOF	Front Patches	Heaviside Patches
Direct	504816	7968	2424
GL	49152 + 409068	5924	2208
LPP	49152 + 670032	9092	3784

All local problems have the global boundary data applied as Dirichlet boundary conditions. Regarding the extraction radius, we choose optimal values for each method and multiply the radius r_{ex} computed by Algorithm 1 by a factor of $f_{ex} = 0.2$ in the global-local cases and $f_{ex} = 4$ in the direct simulation. The actual extraction radii are comparable as the computed radius r_{ex} is based on the global discretization level.

Figure 4.14 shows the error (3.12) in the stress intensity factors for the three numerical solutions over the first six global-local iterations. First of all, we can observe that with enough iterations, both global-local approaches reach the accuracy of the direct solution, even though the latter has more patches at the crack front in this example. Hence, by investing the same number of degrees of freedom, albeit on a different discretization, we get similar solutions even for fine scale features such as the stress intensity factors around the elliptic crack front. Note, however, that the number of degrees of freedom on the global problem for both global-local approaches compared to the direct solution. As expected, the LPP approach requires more iterations to reach the same accuracy as the single local problem due to the poorer boundary data close to the crack. In the GL case, the final approximation quality is already reached in the second iteration and the initial solution is already quite accurate, too. This is all the more impressive given that in the initial global problem the crack is not modeled at all.

Overall, we therefore conclude that the global-local method provides the same accuracy as a direct solution, at least after a few iterations, when the same refinement around the local feature is used. Hence, it is a valid alternative to adaptive h-refinement. For a given simulation, the remaining question is only whether the reduced number of global degrees of freedom outweighs the required additional work in overall runtime performance. In Section 4.3.4 we present a numerical experiment investigating that tradeoff.

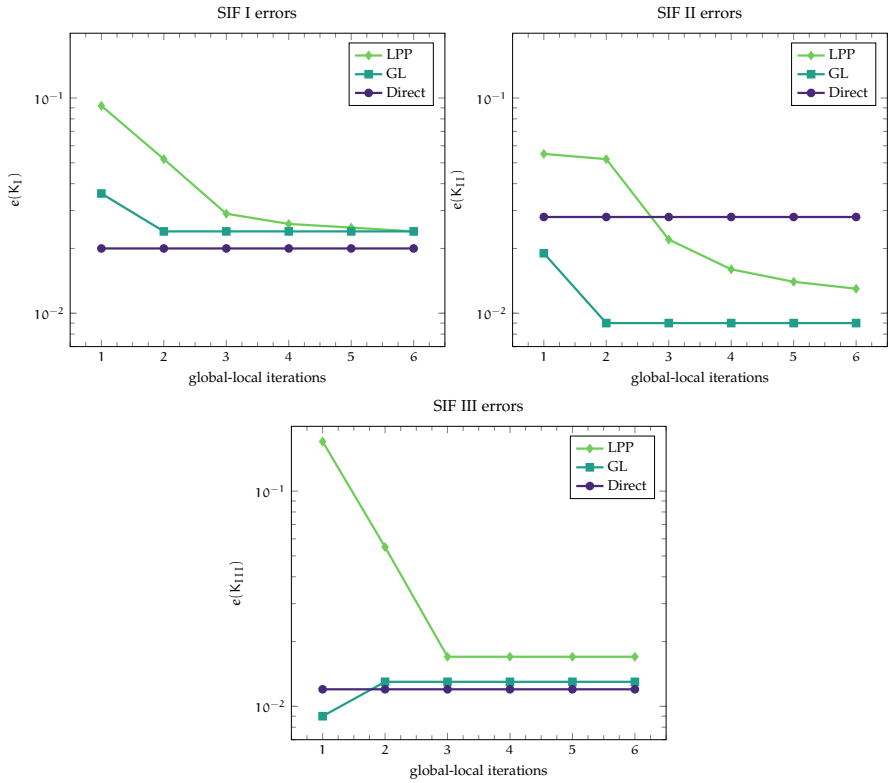


Figure 4.14: Errors in stress intensity factors (SIF) for the inclined elliptic crack extracted with the DCM over several global-local iterations. Comparing errors of a direct solution to having one local problem around whole crack (GL) and having one local problem per crack intersecting global patch (LPP).

4.3.2 *Boundary Effects*

Due to inaccuracy in the boundary data provided by the initial unenriched global solution, the first computed enrichment in the global-local method does not have optimal approximation properties. This improves in each iteration, as we observed in the last examples, but computing the iterations is expensive. We would rather improve the first or first few iterations. For this reason, we test two methods to improve the computed enrichment. First, we measure the effect of applying the global boundary data through Neumann, Dirichlet or Robin boundary conditions on the local problem. Second, we additionally evaluate the global solution further away from the enriched area by applying a bufferzone. To this end, we again study the validation Example 4.1 and the inclined elliptical crack example from Section 3.3.2. In the latter, we set up the local problem on the bounding box of all eight global patches intersecting the crack and discretize it on level four. We further increase the refinement by one level around the crack and additional three levels at the crack front. In both examples, we increase Young's modulus to $E = 1000$ to better illustrate the effect of the Robin parameter κ.

ROBIN PARAMETER Before we compare the three types of boundary conditions, we first study the choice of the parameter κ in Robin boundary conditions. In (4.11), we defined a scheme for estimating the optimal value of κ based on the global solution. Figure 4.15 shows the obtained accuracy in the first global-local iteration for varying parameter κ in the validation example. The dashed vertical line further indicates the value estimated by the scheme (4.11), while the vertical line shows the result obtained from Dirichlet boundary conditions.

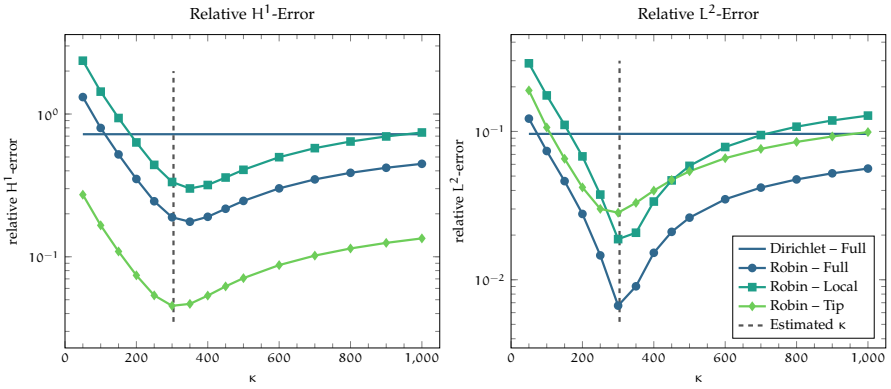


Figure 4.15: Global-local error in first iteration with Robin boundary conditions and varying parameter κ. Kappa computed in first iteration is 304 and 252 respectively, while the safe but suboptimal heuristic (4.9) estimates 1333 in this simulation.

First of all, our estimate for κ is more or less equal to the optimal value in this example, both in H^1- and L^2-norm, especially when concerned with the error close to the crack front. Note that the estimate of κ proposed in [68] is always greater than Young's modulus E for reasonable meshes and thus yields suboptimal results in this example. Worst and best results are separated by almost an order of magnitude in H^1- and even more in L^2-norm. Furthermore, we can observe that Robin boundary conditions indeed converge to Dirichlet boundary conditions in the $\kappa \to \infty$ limit. As Robin boundary conditions interpolate between Neumann boundary conditions, $\kappa = 0$, and Dirichlet boundary conditions, $\kappa \to \infty$, we can already conclude that Robin boundary conditions with approximately optimal κ should perform better than the other two in this example, at least in the first iteration. Figure 4.17 confirms this.

However, looking at the error in extracted stress intensity factors around an elliptic inclined crack as in Section 3.3.2, the picture changes somewhat. Figure 4.16 shows the error (3.12) for all three crack opening modes and varying values of the parameter κ in Robin boundary conditions in the first global-local iteration. We show results of extraction with the CIM and with the DCM for comparison. Dashed lines indicate the result using Dirichlet boundary conditions. Here, the effect varies across the crack opening modes and also between extraction methods. In mode I and II, we more or less see the expected results with both extraction methods: Values of κ that are too small yield worse results than Dirichlet boundary conditions, to which Robin boundary conditions converge for large κ. In between, we have a small window where Robin boundary conditions outperform Dirichlet boundary conditions. In mode III, however, both extraction methods approach Dirichlet results from above.

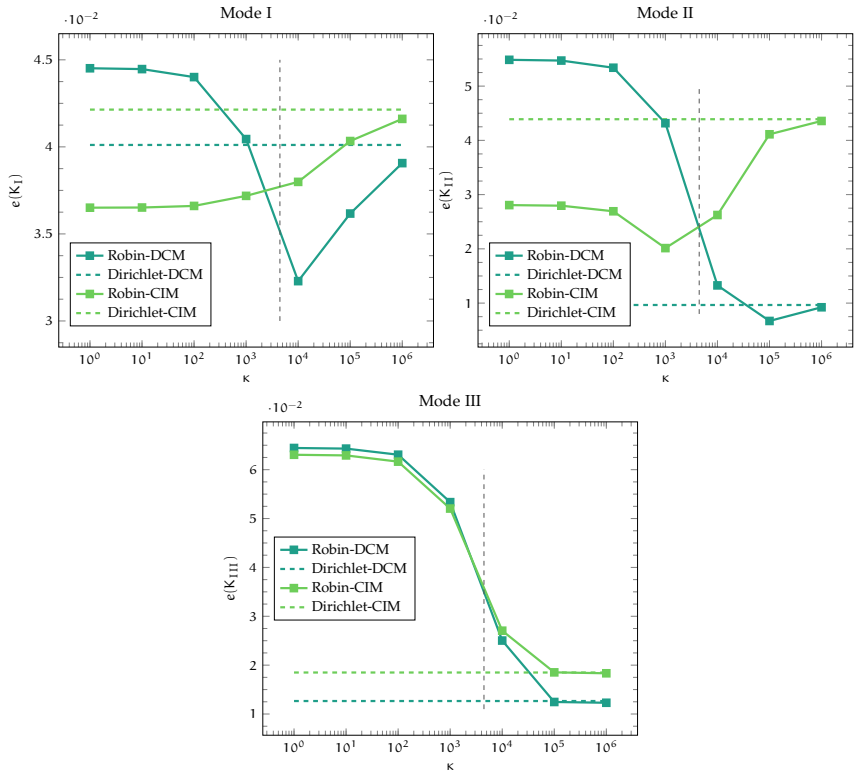


Figure 4.16: Errors of extracted stress intensity factors around an inclined elliptical crack in the first global-local iteration. We applied Robin boundary conditions with varying parameter κ. Dashed lines indicate the computed parameter κ at 4.46×10^3, whereas the heuristic (4.9) estimates 8.50×10^3.

Interestingly, extraction with the CIM seems to work better with near Neumann boundary conditions than extraction with the DCM, likely because the latter considers function values only and ignores the derivative of the displacement, whereas the CIM considers both. Figure 4.16 also shows that our scheme (4.11) for estimating the optimal value of κ produces decent results in configurations with an actual optimal parameter, but cannot account for the variability in results between crack opening modes. The heuristic (4.9) estimates a parameter of comparable magnitude in this example.

When interested in the stress intensity factors, the choice of boundary conditions and parameter κ seems to depend on the considered crack opening mode and the extraction method. Especially in extraction with the DCM, Dirichlet boundary conditions clearly seem to be preferable overall. We conclude that estimating an optimal value of κ may be difficult or impossible in some examples, at least for such detailed features of the solution.

BOUNDARY CONDITIONS AND BUFFERZONE Next, we compare Robin boundary conditions with the estimated optimal parameter κ to Neumann and Dirichlet boundary conditions over multiple global-local iterations. We additionally include the effect of a bufferzone by increasing the local domain size by 10, 30 and 60 percent with the stretch factors $f_\tau \in \{0.1, 0.3, 0.6\}$. Note that for this

comparison to work, we had to slightly adjust the setup of the local domain. By default, applying a bufferzone results in bigger local patches, i. e. larger local h, as the domain size and consequently the size of the initial bounding box for the cover construction increases. For the experiment at hand, we thus fix the size of that initial bounding box across all bufferzone sizes and thereby keep the resolution around the crack front constant. In the validation example, we report the error in the H^1-seminorm only as it dominates the overall error. We also only show errors on the full domain Ω and around the tip Ω_{tip} as the error on the local domain Ω_L behaves as on the full domain. Again, we repeat the same study using the inclined elliptical crack example. For the comparison of bufferzone sizes, this example is especially interesting, since the local domain is smaller here and its boundary is closer to the crack. In both examples, we extract the stress intensity factors with the DCM.

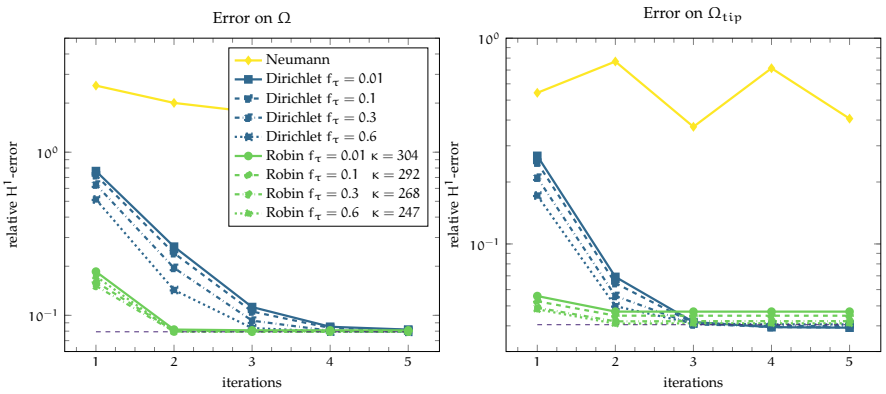


Figure 4.17: Global-local H^1-error over iterations with Dirichlet and Robin boundary conditions for varying bufferzone size. Bufferzone computed with factor f_τ. Error evaluated on full domain (left) and around tip (right). Solution with true solution as boundary data and no bufferzone for comparison. Robin boundary conditions with parameter κ computed in first iteration.

Figure 4.17 shows the results for the validation Example 4.1. The dashed line shows the error obtained from Dirichlet boundary conditions with the true solution u^* as boundary data, indicating what would be possible with perfect boundary data. Neumann boundary conditions clearly perform the poorest, which is why we drop them from additional analysis. As expected, Robin boundary conditions produce best results in the first iteration throughout all cases and almost halve the error of the Dirichlet solution. Both types of boundary conditions eventually reach the accuracy provided by perfect boundary data, which further validates the global-local method. Interestingly, we obtain more accurate results close to the crack front with Dirichlet boundary conditions in and after the third iteration than with Robin conditions. This might be due to not recomputing the Robin parameter κ after the first iteration. Yet, doing so would imply re-assembling the local stiffness matrix, or caching three system matrices in memory, which is not ideal either. Looking at the bufferzone, their addition improves accuracy through all iterations as the error estimate (4.2) suggests, although the effects are negligible with Robin boundary conditions. With Dirichlet boundary conditions, however, a fairly large bufferzone closes the gap to Robin boundary conditions around the tip by the second iteration.

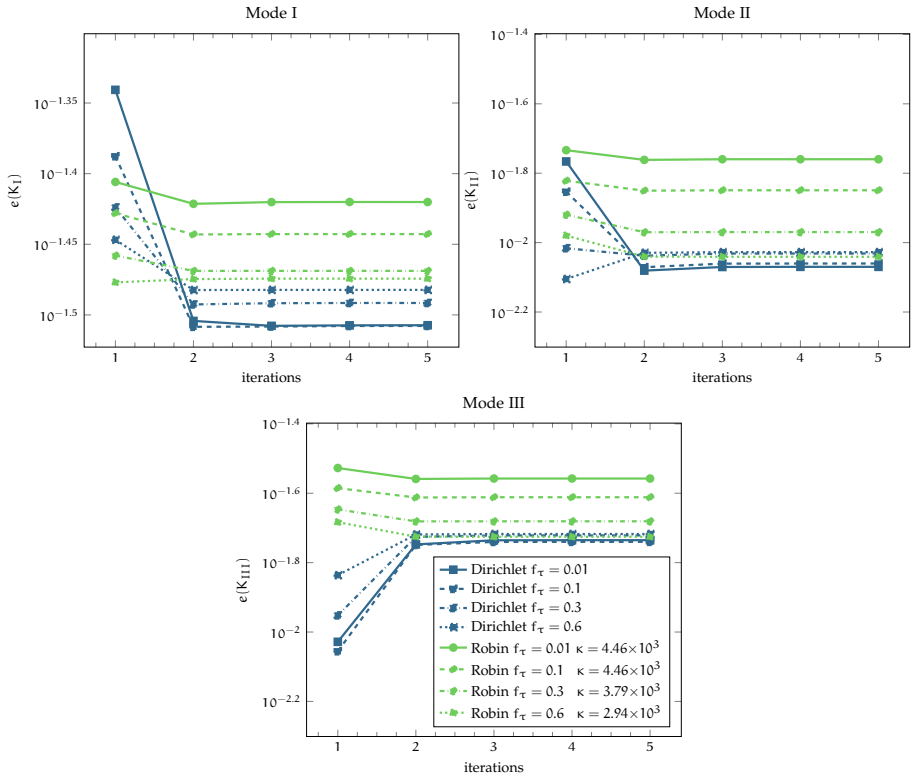


Figure 4.18: Errors of extracted stress intensity factors around an inclined elliptical crack over multiple global-local iterations. Extraction is performed with the DCM. We compare Dirichlet and Robin boundary conditions in combination with increasing bufferzone sizes. Robin conditions use the computed parameter κ by scheme (4.11).

Figure 4.18 shows that we see more effects from the boundary condition types and the bufferzone size in extracted stress intensity factors around the inclined elliptical crack. For one, applying a bufferzone reduces the error in the first iteration in almost all cases. The mode III error with Dirichlet boundary conditions is the only exception here. As we already have seen in Figure 4.16, Robin boundary conditions improve accuracy in the first global-local iteration only in mode I compared to Dirichlet conditions. The bufferzone size has different effects on the boundary condition types after the first iteration: With Dirichlet conditions we observe a slight decrease in accuracy, with Robin conditions a significant increase. We expect the latter because of the error estimate (4.2). However, we also mentioned in the discussion of the estimate that it might not be directly applicable in the global-local context, since we move the local boundary with the application of a bufferzone. Therefore, the slight reduction in accuracy in Dirichlet boundary conditions with a bufferzone might be explained by a slight spatial variation in global accuracy. In this regard, note that Dirichlet conditions we evaluate the displacement only, whereas Robin conditions additionally take its derivative into account. As a result, we achieve the same accuracy with both boundary conditions types after the first iteration with a bufferzone factor of $f_\tau = 0.6$.

CONCLUSION Based on the experiments described above, we conclude that both Robin boundary conditions and a bufferzone show great potential to reduce the error especially in the first iteration. The choice essentially comes down to confidence in the estimate of the Robin parameter κ for a given simulation, where we saw that some examples might not even have one optimal value for all features of the solution. If guessed correctly, though, one single global-local iteration with Robin boundary conditions is probably the best choice if runtime is critical. On the other hand, Dirichlet boundary conditions with some bufferzone and two to three iterations are a robust choice and recommended, especially if accuracy is more important than runtime.

4.3.3 *Parallel Scaling*

In the following example, we investigate the runtime performance of the global-local method. We first assess the parallel scaling of the parallelization approach introduced in Section 4.2 and then continue to compare the runtime of a direct global solution to global-local enrichments. To this end, we study a cube of size 32 with up to a hundred penny shaped cracks, compare Figure 4.19 for a sketch with 24 cracks. The cube is comprised of an isotropic, linear elastic material with Young's modulus $E = 1.00 \times 10^3$ and Poisson's ration $\nu = 0.33$. For boundary conditions, we prescribe a displacement of $0.025 \cdot 32$ in normal direction at the top and bottom (z-direction) of the domain. Cracks are positioned in a way that each crack intersects exactly eight global patches on global level three. These patches define a single local problem for each crack.

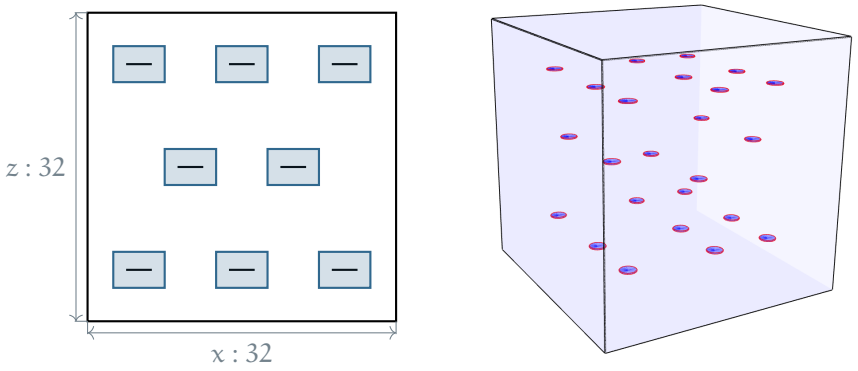


Figure 4.19: A cube with 24 penny shaped cracks in total (right). Slice through domain with sketched local domains per crack (left).

In a first experiment, we study the efficiency of the parallelization scheme for the global-local method described in Section 4.2. Here, we measure the runtime of the solution step of the local problems and of the assembly of the global, enriched linear system, which uses the parallelism of the local integration cells. We pick ten crack locations in our multi-cracked cube example and run the same simulation with up to 40 processes on a single compute node with four sockets of twelve cores each. Consequently, the simulations range from 10 local problems on a single process to solving each local problem on four processes in parallel. Note that for these scaling measurements we do not apply any crack enrichments nor did we resolve the cracks in the integration cells, as this would

create further load balancing issues outside the scope of this study. Both global and local problems are discretized on level four, resulting in 4.94×10^4 and a total of 4.92×10^5 degrees of freedom, respectively.

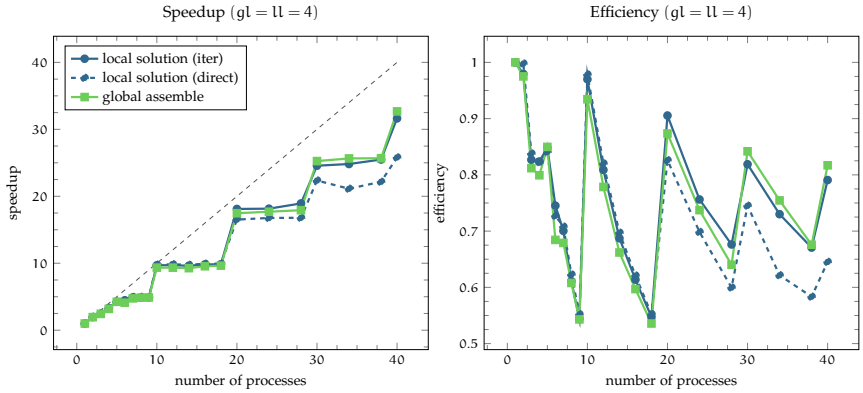


Figure 4.20: Speedup and parallel efficiency of the global-local method on a problem with 10 separate cracks. Local problems are solved in parallel on split communicators. Both global and local problems are discretized on level four, with 4.94×10^4 and 4.92×10^5 degrees of freedom, respectively, where we summed over the local problems. For the local problems, we show results with an iterative (CG with block Jacobi) and a direct linear solver. Optimal speedup is indicated by the dashed line.

Figure 4.20 shows speedup and parallel efficiency of our implementation of the global-local method in PUMA. One aspect immediately noticeable in the plots is a stair-like structure, since the required time is limited by the slowest local problem. That is, while the total number of involved processes increases, the runtime only decreases each time all local problems receive one extra process, hence every ten processes. That is, while increasing the number of total involved processes, the time to solution only drops every time all local problems receive one process extra, i. e. every ten added processes. If we limit the analysis to process counts that are multiples of ten, our parallel implementation shows excellent parallel efficiency, especially for higher process counts: Both the local solution as well as the global assembly step have an efficiency above 0.8. This is reinforced by the fact that we solve each local problem with only up to four processes, which helps to ensure that all local routines have excellent efficiency as well. We can observe the effect of this by comparing the local timing results with a direct solver and an iterative solver. The former does not scale well by design, which is why the global-local algorithm with an iterative local solver shows better overall parallel performance for higher process counts. In contrast to the results obtained in [66], where local problems too are solved in parallel to each other but each only sequentially, our approach seems to utilize more processes more efficiently: Their reported efficiency begins to drop rapidly after 20 processes, although they have 983 local problems for eight crack locations in one example. This issue would be amplified on coarse global discretization, as the number of local problems is tied to the number of enriched global patches.

The highlight of the global-local method in terms of parallelism, however, should be its excellent weak scaling with local features due to their independence. To test this, we increase the number of local problems in the same model problem

up to a hundred, further changing their position and size so that each coincides geometrically with one global patch. In order to have more work available we also increase the local level to five and perform benchmarks with an increasing number of local problems, each distributed on four processes. Figure 4.21 shows the runtimes of the local solution step and the global assembly, measured on up to fifty nodes with two sockets of eight cores each. Compared to the previous example on a single compute node, this increases the cost of communication between processes. We can nonetheless observe that weak scaling is excellent, since the runtime is constant in the assembly step and only a negligible slow down occurs in the local solution step. In this regard, it should be mentioned that the global problem is still only discretized on level four in all runs. Hence, the ratio of local processes interacting with global processes deteriorates with an increasing number of local problems. Especially in the local solution step, all processes on the local problem have to evaluate the global solution for the boundary conditions, while each process only owns around ten global patches with 400 processes .

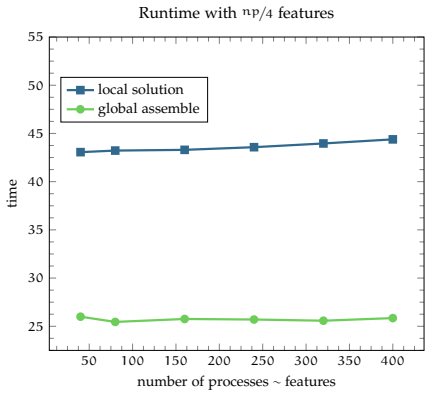


Figure 4.21: Weak scaling of the global-local method with respect to an increasing number of local features (up to a hundred), while the degrees of freedom on the global problem are kept constant.

We conclude that our implementation has the property of embarrassingly parallel local problems and its weak scaling thus is excellent. This suggests that the global-local method can also be used to improve parallel efficiency in the presence of multiple cracks or similar features, in addition to condensing the required global degrees of freedom. For best performance, the number of processes should be a multiple of the number of local problems, though. While not a direct comparison on the same problem and hardware used in [66], our experiments suggest that our decision to create fewer local problems but solve them in parallel does not limit the parallel efficiency of the global-local method. Moreover, it promises better performance on coarser global discretizations. While only an implementation detail, our distributed memory parallel implementation can utilize more processes on common HPC clusters compared to a thread-parallel implementation, even though thread-parallel local problems on a distributed global problem are a straight forward extension of the approach presented in [66]. In strong scaling, the parallel efficiency of the linear solver employed on the local problems affects the overall parallel performance of

the global-local method. Regarding the runtime of the method with multiple iterations, this introduces a tradeoff between solvers that scale well but are not reusable and solvers that scale worse but produce a factorization that can be cached. This tradeoff has to be investigated further.

4.3.4 *Runtime Performance*

The global-local method reduces the number of global degrees of freedom by introducing additional assembly and solution steps. This is even more true when running several iterations, which may be necessary to reach a desirable accuracy if we start with an unenriched global problem. In this section, we try to answer if and under what circumstances the additional work still pays off in terms of computation time. While e. g. nonlinear problems where a Newton solver benefits from the reduced number of global degrees of freedom promise more speedup by the application of the global-local method, we use a simple linear elastic problem in this section. If we see some speedup here, we are confident that this should also translate to more difficult problems.

We consider a problem where the local problems together have at least a comparable size as the global problem, else we would not want to use the global-local method. We compare the global-local method to a direct simulation of the same problem with h-refinement and enrichments applied directly on the global problem, such that both solutions have the same local resolution and thus comparable accuracy after some iterations. The global-local method has the following runtime contributions: The initial unenriched global solution is rather cheap due to the small number of degrees of freedom and cheap assembly. Since the local approximation space does not change over global-local iterations, we have to assemble the local stiffness matrix only once and only need to reassemble the changing boundary data. Depending on the employed local linear solver, we can even keep the factorized local stiffness matrix. In each iteration, we then have to solve the enriched global problem, which is significantly smaller than the global problem in the direct simulation, but still requires to traverse the same number of integration cells in assembly since the local enrichment has to be evaluated. We thus expect the global-local method to improve overall performance if the ratio of global to local degrees of freedom is small.

To verify this numerically, we solve a linear fracture mechanics problem in a three-dimensional cube with up to 24 penny shaped cracks, compare Figure 4.19, using 48 processes. In this experiment, we model the cracks in both the global-local and the direct simulation, since we are not interested in the scaling of the abstract method, but in a performance comparison on a more realistic example. Patches intersecting the crack front are thus enriched with the front enrichments (3.3) and all other patches intersecting the crack use the Heaviside enrichment (3.1). In the direct global solution, adaptive refinement is applied such that the resolution around the crack is identical to that of the global-local solution and we can therefore assume accuracy to be comparable after some iterations. As discussed in Section 4.2, load balancing of adaptive refinement is not optimal in PUMA at the time of writing this thesis, although it should be relatively easy to improve this. We thus try to limit the effect of the load balancing issue on the runtime of the direct simulation in two ways. First, we increase the maximal level difference between neighboring patches to

two. Thereby, we generate less patches in total and especially less patches with neighbors on a different level. Second, we significantly reduce the order of the quadrature rules applied in the overlap of patches on different levels, where the PU functions (2.78) are rational. Specifically, we applied exact quadrature rules for polynomials of only one degree higher than required for the weak formulation in those overlaps. This reduces the computational load associated with integration cells in the overlap of differently sized patches and thus balances the load distribution. Based on the previous results shown in Figure 4.20, we solve all global problems with an iterative solver, namely CG preconditioned with block Jacobi, and employ a direct solver on the local problems, as they are solved by only two processes for many cracks anyway. Note that a multilevel solver showed no improvements for these problem sizes, but we expect this to change for larger problems. The global-local method is parallelized using split communicators for the local problems as discussed in Section 4.2, such that each local problem is distributed across two processes when all 24 cracks are present.

Overall, as the number of cracks increases we expect the global-local method to be faster than the direct solution. The estimate (4.5) proposes that the performance advantage of the global-local method depends on the ratio of overall local to global degrees of freedom. To test this, we run the experiment once with both global and local problems discretized on level four and increased the global level to five for a second run. The cracks are positioned such that exactly four global patches define one local problem with global level four. In the direct simulation, we thus apply three additional levels of refinement on those patches. With global level five, we keep the local resolution and thus the size of the local problems constant. Therefore, eight times more global patches are enriched in the global-local case and we apply only two additional local refinement levels in the direct simulation. The respective results are shown in Figure 4.22. But before we summarize them, we want to explain the probably unexpected effects from increasing the global discretization level observable in the direct solution. Between the area around the cracks and the rest of the discretization, we have a level difference of three with global level four and of two with global level five. Together with the applied maximal level difference of two, we generate additional patches around the refined crack areas with global level four, but not with level five. In consequence, we only have marginally more degrees of freedom in the direct simulation with level five and the solution step does not get significantly more expensive. Additionally, and although we limited its effects in this example, the current load balancing of adaptive refinement in PUMA leads to the assembly step being actually faster with level five than with level four. Also, with level five, we can observe less fluctuation in the runtime of this step with different numbers of local cracks in the simulation. Hence, comparing the two direct solutions in the columns of Figure 4.22 might be perplexing, but the comparison with the respective global-local solution in two different distributions of global and local degrees of freedom is what we wanted to investigate here. To summarize, providing simulation parameters for a fair performance comparison of a direct and a global-local simulation in parallel is not trivial. Yet, we believe that our choices favor the direct one in runtime and that the global-local solution is slightly more accurate here.

Returning to the results, Figure 4.22 clearly shows that the direct solution scales worse with the introduction of additional cracks than the global-local

method. It also shows that the cost is driven by the linear solver as we modeled in our runtime estimate (4.5). Concerning the global-local simulation, the time spent in the global linear solver is almost constant with respect to the number of cracks as we only add one degree of freedom per space dimension to each enriched global patch. Furthermore, the overall time, the global assembly time and the time required for the local solutions show the characteristic kinks introduced by the parallelization strategy: These times are determined by the smallest number of processes available for a local problem. We can observe that for global level four, the global operations in the global-local method are cheap and thus global-local iterations are cheap. Therefore, starting from around ten cracks even the third global-local iteration finishes before the direct solution is completed. This changes when we increase the global level to five, where global operations suddenly become noticeably more expensive. Here, the first global-local iteration is still cheaper than the direct simulation by a wide margin, but the second iteration only if we have twenty cracks or more. For the third iteration, it seems that we would have to add a few more cracks to the simulation for the global-local method to be faster.

Our performance estimate (4.5), however, seems to be off by about a factor of two, i. e. replacing it by

$$\mathrm{dof}_g \leqslant \frac{n \cdot \mathrm{dof}_l}{i} \quad \Longrightarrow \quad \mathcal{O}_{\mathrm{gl}} \leqslant \mathcal{O}_{\mathrm{direct}} \tag{4.15}$$

would fit the results of this experiment better. This estimate would predict the global-local method to have a performance advantage starting from one, two and three cracks with global level four and eight, 16 and 24 cracks with global level five in the first, second and third iteration, respectively. While still slightly optimistic, it better catches the trend in this experiment.

We conclude that the global-local method can improve the performance of even simple simulations of linear elasticity, conditional on the relative number of degrees of freedom required for the global problem and the resolution of local features. Compared to directly applying adaptive h-refinement, it especially provides an excellent parallelization opportunity, allowing perfect scaling of simulations with local features by adding only a constant number of processes per feature.

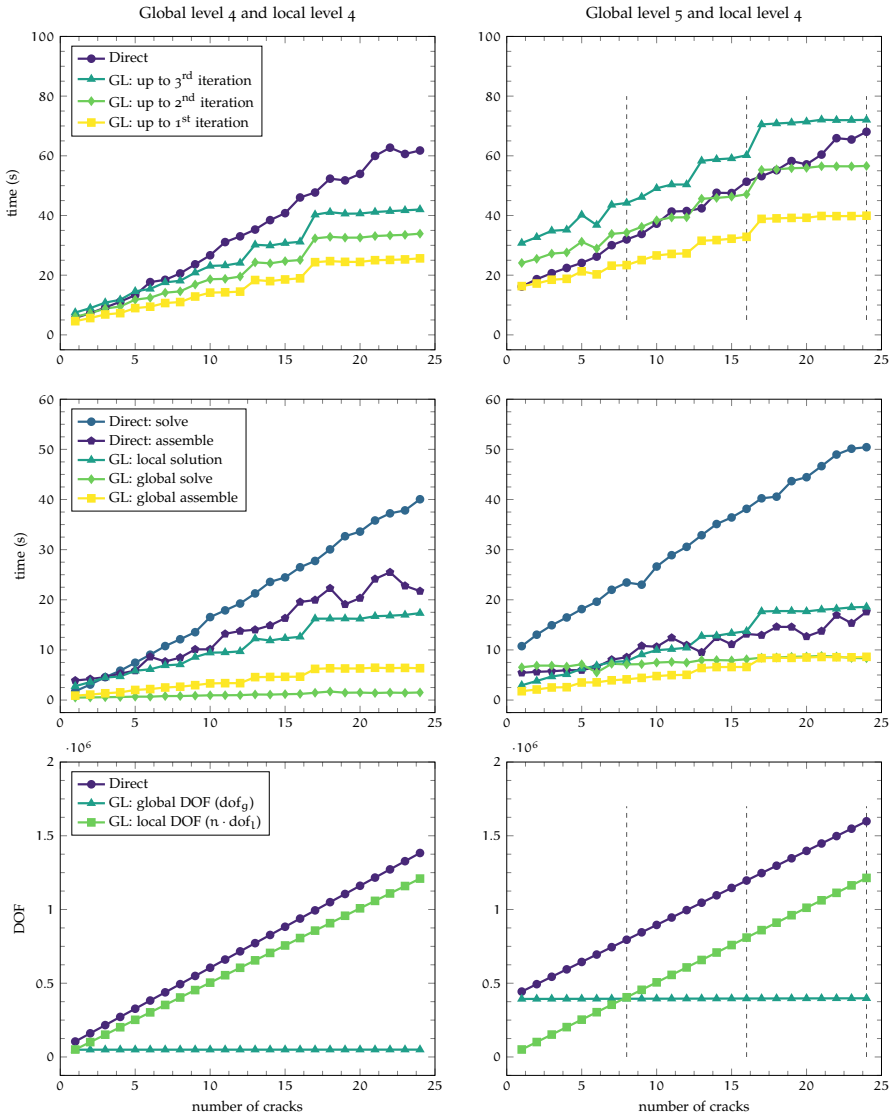


Figure 4.22: Time measurements and degrees of freedom (DOF) for a direct and for global-local (GL) solution on a problem with increasing number of cracks. All simulations are run in parallel with 48 processes in total, resulting in a minimum of two processes for each local problem.

5 GLOBAL-LOCAL COUPLING METHOD

In the global-local method, the local problem does not have to simulate the same material model or use the same discretization as the global problem. In principle, any method that accepts boundary data from the global problem and generates some form of enrichment in return is applicable. This presents an excellent coupling opportunity, which to our knowledge has not been exploited in the literature yet. In the context of fracture simulation with a global computationally cheap simulation of linear elasticity, two methods in particular appear to be complementary, as we discussed in Section 2.1: peridynamics [126] and the phase-field method [28]. Opposed to linear elastic fracture mechanics, both methods handle fracture initiation and propagation naturally, but are computationally very expensive in return [21].

Applying the local solution directly as an enrichment on the global PUM problem may face one challenge, though: The solution of the different model on the local problem does not necessarily minimize the global linear elastic energy. This can be circumvented by not using the local solution directly, but instead extracting the crack path from it. The updated crack path can then be modeled by the usual crack enrichments in the global PUM problem. This way, the local model is still used to predict the crack propagation and the global problem is also cheaper in consequence. Crack path extraction is further necessary in any method that localizes peridynamics or the phase-field method once the local region moves with the propagating crack, at least if we want to keep the local region small. We therefore generalize the global-local method to a coupling method by interpreting the extracted crack path as the computed enrichment, compare Figure 5.1. We already proposed this approach in the context of coupling with peridynamics in [8].

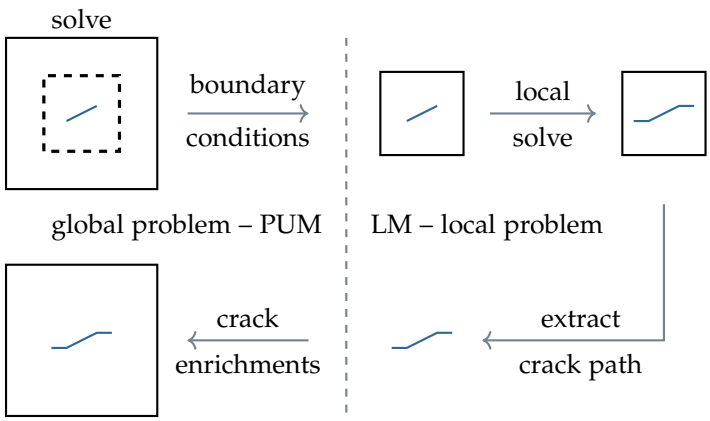


Figure 5.1: Generalized global-local coupling method. Crack growth happens in the local model (LM) and the updated crack path is incorporated back into the global PUM problem using standard crack enrichments. Of course, this procedure can be iterated if necessary.

We want to highlight one particularly notable feature of coupling via global-local enrichments: The two involved models and discretizations do not interact directly at any point. The local problem is independent and only happens to receive boundary data from another method. On the global problem, we only update the crack geometry. This has the following advantages compared to tight coupling of methods. Usually, in the coupling of two methods an artificial interface is introduced between them, where the exact interaction of the two models or discretizations has to be defined. There is no clear answer on how to do this and the interface may cause problems such as wave reflections as we discuss shortly. Second, local regions can be added, relocated and discarded trivially during a running simulation to adapt the resolution locally. Lastly, it can be implemented non-intrusively, for example for fast prototyping or connecting to commercial codes. This is possible because the global and local problem can be solved independently and information between the two involved codes can be easily transferred using standard file formats.

Due to their complementing nature, multiple methods have been proposed to combine peridynamics or the phase-field method with, most of the time, finite element based discretizations of classical continuum mechanics models and are still actively researched. With regards to phase-field methods, we want to highlight two publications here. In [31], the global-local method is applied to a phase-field formulation; however, the proposed approach simply computes two separate enrichments for the global elasticity and phase-field problems and thus no actual coupling via the global-local method is performed. In [36], the XFIELD is proposed as a means to couple the XFEM with the phase-field method, which is similar to the approach presented in this thesis. In contrast to what we present here, though, the authors apply additional boundary conditions on the phase-field problem. We believe that the latter is not necessary, which is also confirmed by our numerical experiments.

Regarding peridynamics, the review article [18] classifies and compares nine different approaches of local to non-local coupling of continuum models. These are either energy-based or force-based and can further be categorized by how they treat the transition zone between the two models. Moreover, at least six further methods are mentioned in [18] that tackle the coupling problem at the discrete level. The global-local coupling presented in this thesis falls into this latter category. In this category, the publication also lists the coupling approach between peridynamics and the FEM based partition of unity method XFEM presented in [35], where the peridynamics region is restricted to the area around the crack tips, while the rest of the crack is modeled using a Heaviside enrichment. In this method, however, the solution around the crack tip is given by the peridynamic solution exclusively and no enrichment is used to capture its response. Common to all coupling approaches mentioned in [18] is the issue of wave reflections at the interface in dynamic simulations, which is explained and investigated in more detail in [34]. Here, waves reflect at the interface instead of transmitting to the other model, leading to simulation artifacts in the PDE model and overheating in the peridynamics model as energy cannot leave the peridynamic region. To the best of our knowledge, a definitive solution to this problem does not exist yet, and circumventing it requires fine-tuning of discretization parameters and, in particular, prohibits strong differences in refinement between the coupled methods. Yet, having a much finer discretization

of peridynamics than of the PDE is partly the aim of their coupling. The approach envisioned in this thesis, however, should not exhibit wave reflections due to the loose coupling via the enrichment function from the separate local problem: We do not transfer energy or couple function values between the models, but rather compute a consistent global solution based on the extraction of geometric features from the independent local solution.

Regarding the actual extraction of the crack path, we found three approaches in the literature. The probably simplest to implement is recovering the crack path based on intersections of the damage field with mesh cells, as presented in [90]. However, extending this approach to three space dimensions seems to be more involved since more corner cases are possible. Another approach detailed in [33] but also used in the XFIELD is computing a discrete gradient of the damage field, whose minimum should be the crack tip. The most promising approach, also according to [36], seems to be the θ-simplified medial axis algorithm [10], though, which was applied to crack path extraction in [138]. Essentially, it recovers the centerline of extracted iso-lines from the damage or phase-field and does so robustly [105]. It is further independent of the discretization width and can easily be extended to three space dimensions. In the PUM we can efficiently extract iso-lines of the damage field by using the marching cubes algorithm [80] on the integration cells.

The remainder of this chapter is structured as follows: In the next section, we present the PUM-peridynamics coupling using the global-local method developed in joint work with P. Diehl and R. Lipton and published in [8]. We then present selected numerical results of [8] together with so far unpublished comparisons against real world experimental data. We conclude the numerical examples with initial results of coupling the PUM with a local phase-field model, using the same approach.

5.1 COUPLING WITH PERIDYNAMICS

In [8], we already used the global-local coupling algorithm presented in Figure 5.1 to combine the PUM with peridynamics. Coupling with peridynamics presents three additional challenges in the global-local method, which we discuss in the following.

To begin with, we have so far sketched the coupling algorithm without taking into account time. That is, we assumed that we solve the equilibrium PDE in the PUM and the local problem quasi-statically. However, as the name suggests, the peridynamic model is usually stated in time, and quasi-static solutions of it are computationally even more expensive. Yet, solving both methods dynamically introduces new challenges both on the PUM side and in the interaction of the methods. In the PUM, crack growth changes the set of basis functions as the crack enrichments move with the growing crack. As a result, solutions of different time steps exist on different function spaces, which has to be handled during time integration. Stable time integration schemes that account for this issue are available in the literature [25, 38]. Large crack updates can further introduce shock waves in the linear elasticity solution. Regarding the interaction in time of the two methods, the questions arise about how often we exchange information and whether peridynamics requires a smaller time step size than the PUM. For the proposed method, this is left for future work. However, different time

step sizes and integrators in an uncoupled global-local setting have already successfully been applied in [50]. Instead, we focus on the compatibility of the two methods in this thesis and if successful transmission of information between them is possible. To this end, we solve the equilibrium equation (2.6) in the PUM problems of our numerical experiments, disregarding time. In peridynamics, we approximate an equilibrium solution by slowly increasing the prescribed displacement up to the value provided by the PUM solution, since a quasi-static peridynamic solution is too expensive. While not presented in this thesis, we further verified that for most model problems discussed in Section 5.3, dynamic and quasi-static PUM solutions actually coincide, due to the undynamic nature of the examples.

The second challenge in coupling non-local peridynamics with the linear elastic PDE model side is the difference in the underlying material models. We expect good agreement in the material responses as long as we are in the linear regime of the peridynamic material model (2.16). This assumption is tested numerically in Section 5.3. In the presence of softening bonds on the other hand, we expect the material responses to be quite different. In consequence, we have to make sure that the local problems' boundary is far enough away from any arising discontinuities in the PDE model when coupling with peridynamics.

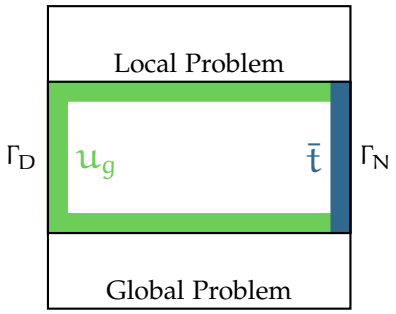


Figure 5.2: Boundary condition layers on local peridynamics problem. While global Dirichlet boundary data $\bar{u}$ on Γ_D is best applied through the global solution u_g, global Neumann data $\bar{t}$ on Γ_N can be applied as in (5.2).

The third challenge is that we do not have the same notion of a boundary in peridynamics as in the PDE model. Rather, boundary conditions are usually applied in a layer of nodes of at least horizon size. On a local peridynamics problem in the global-local method, we have three scenarios, see Figure 5.2 for a visualization of the setting. First, for parts of the local boundary layer away from the global one, we use the global solution u_g as Dirichlet boundary data. Here, the global solution can easily be applied on a layer of peridynamic nodes by simply evaluating it in the respective points. Second, on parts that overlap the global Dirichlet boundary Γ_D, it is probably best to use the global solution u_g too, instead of the the actual data $\bar{u}$. The global solution should already conform to the global boundary conditions and can be applied to a layer of nodes more easily. Third, on parts overlapping the global Neumann boundary Γ_N, the applied force can be distributed across a layer of peridynamic nodes as described in (5.2). In this layer of nodes, however, we should not additionally apply the global solution as Dirichlet boundary conditions, compare Figure 5.2.

5.2 PERIDYNAMICS DISCRETIZATION

We discretize the equation of motion (2.10) in space using a collocation approach, the EMU nodal discretization [19, 127]. Here, discrete peridynamic nodes x_i are distributed along a grid of width h in the reference configuration Ω. To each node we associate a surrounding volume V_i, such that the volumes are non-overlapping $V_i \cap V_j = \emptyset$ for $i \neq j$ and the sum of the volumes approximates the size of the domain $\sum_{i=1}^{n} |V_i| \approx |\Omega|$. On each volume, the peridynamic solution is constant, i. e. we only associate values with nodes. Discretization in time is done via a simple central difference scheme. The discrete version of (2.10) thus reads

$$u(x_i, t^{k+1}) = 2u(x_i, t^k) - u(x_i, t^{k-1}) + \frac{\Delta t^2}{\rho(x_i)} \left(b(x, t^k) + \sum_{x_j \in B_\delta(x_i)} f_{PD}\Big(x_j - x_i,\, u(x_j, t^k) - u(x_i, t^k)\Big) \right). \tag{5.1}$$

For details on the convergence order of this scheme, we refer to [59]. Nevertheless, note that convergence requires the discretization width h to decrease faster than the horizon size δ, which further increases the computational cost of the peridynamics discretization. The density of the resulting system matrix, due to the non-local interactions across the horizon, makes direct solutions of the peridynamic problem impractical.

BOUNDARY CONDITIONS In the non-local peridynamics model, we do not have a sharp boundary as in the PDE model. For all experiments in this thesis, differences in boundary condition treatment between the two methods are addressed as follows. When applying Dirichlet boundary conditions in the PDE model, we prescribe a fixed displacement on all peridynamics nodes in a layer of one horizon size at the actual geometry boundary. For Neumann boundary conditions, we transform the applied force to an external force density b by

$$b(x_i, t) = \frac{F}{V}, \tag{5.2}$$

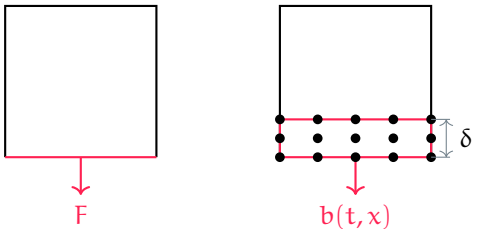
where F is the applied traction force in Newton and V is the sum of all volumes V_i associated to nodes x_i in the layer of horizon-size δ. The force density b is then applied on a layer of peridynamic nodes, as illustrated in Figure 5.3.

DAMAGE FIELD No bonds are broken in the above peridynamic model and we assume fracture based on the damage function

$$dmg(x_i) \coloneqq \frac{\max_{x_j \in B_\delta(x_i)}(r_{i,j})}{r_c}, \tag{5.3}$$

evaluated at discrete points x_i. Here,

$$r_c = \sqrt{1/\beta} \tag{5.4}$$



(a) PDE model.

(b) Non-local peridynamics.

Figure 5.3: (a) Traction condition in the PDE model on the lower boundary. (b) Application of the non-local traction conditions in peridynamics in a layer of horizon size δ. In the $\delta \to 0$ limit, both boundary conditions coincide [77].

is the critical value where the material model transitions from linear behavior to softening, i. e. the inflection point of g_{PD}, and

$$r_{i,j} := |x_i - x_j| \, S^2\Big(x_i - x_j, f_{PD}\Big(x_j - x_i, u(x_j, t^k) - u(x_i, t^k)\Big)\Big) \tag{5.5}$$

is the stretch between two discrete peridynamic points as passed to the derivative of the double well potential (2.16) in the material model (2.11).

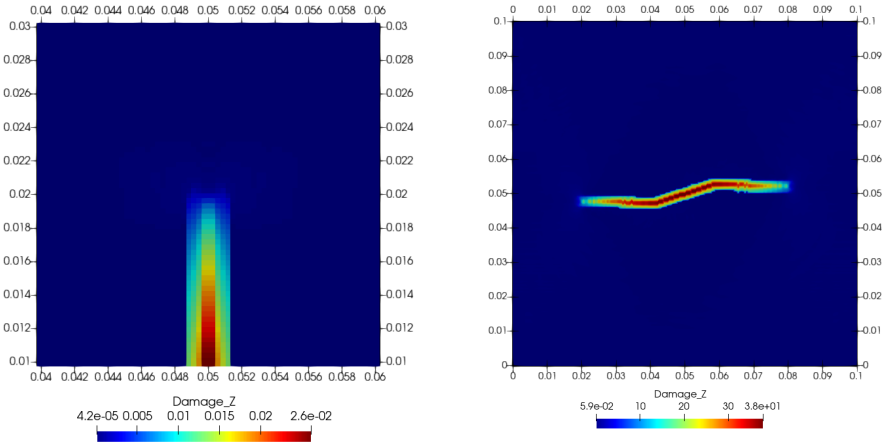


Figure 5.4: Peridynamic damage function `dmg` of two fracture simulations on coarse meshes to highlight fuzziness of crack location.

From the perspective of our coupling method this implicit crack representation has the disadvantage that the crack location in peridynamics is only determined up to the discretization width h and might be even fuzzier if multiple neighboring nodes experience high damage, as for example shown in Figure 5.4. In the PUM, however, we expect a sharp and precise crack and thus inevitably introduce some geometry approximation error when extracting the crack path from the peridynamic solution. Conversely, if we want to apply an initial crack configuration in a peridynamics simulation, we have to arrange nodes symmetrically around the crack geometry for best results, if possible.

Regarding the actual extraction of the crack path we have not yet implemented any automatic algorithm. Instead, crack paths are currently extracted by hand from visualizations of the damage field, which was enough for the purposes of the first publication and this thesis.

5.3 NUMERICAL RESULTS

In this section, we first present selected results of the coupling of the PUM with peridynamics via the global-local method already published in [8]. The goal of these experiments is to verify the soundness of the proposed coupling approach numerically, while not yet tackling real world simulations. We conclude the section with results of so far unpublished experiments on the peridynamics coupling that are part of the ongoing collaboration. In these examples, we try to reproduce physical experiments with the global-local coupling method for the first time. Since we neglect time and extract the crack path by hand in rather coarse intervals, we do not expect perfect results at this point. Rather, we hope to obtain reasonable results that demonstrate the feasibility of the proposed approach and motivate more studies of it in the future.

For the combination of the two different material models to be feasible, it is essential that they predict comparable material responses from the same load, at least in regions of linear elastic deformation away from any discontinuities. In the first experiment, we thus apply increasing loads to an uncracked bar and compare the computed displacements. In the second experiment we introduce a crack, but apply a load that produces an equilibrium solution without crack growth in both methods. Hence, we still expect similar material responses. In this experiment, we further solve a local peridynamic problem around the crack tip with boundary data from the global PUM solution and check that the solution is interchangeable with a local PUM solution. In other words, we test the transfer of information to the peridynamic model. In the third example, we solve a simple but coupled crack growth example by passing the updated peridynamics crack path back to the PUM and iterating. We continue by comparing results of the coupled method to reference data of a physical three point bending test, which concludes the experiments with a local peridynamic problem. Since the presented coupling approach is not limited to peridynamics, we additionally present initial results with a local phase-field method. In the respective experiment, we also test automatic crack path extraction for a first time. We finish with a phase-field simulation in three space dimensions as an outlook.

Unless stated otherwise, we use the following material parameters in all experiments involving peridynamics: material density $\rho = 1.20\times10^{3}\,\mathrm{kg\,m^{-3}}$, Young's modulus $E = 3.25\mathrm{GPa}$, bulk modulus $K = 2.16\mathrm{GPa}$, a Poisson ratio $\nu = 1/3$ and a critical energy release rate $G_c = 500\,\mathrm{J\,m^{-2}}$. Note that we run all coupled simulations in a quasi-static manner by increasing the applied load over multiple steps. We thus solve the equilibrium equation (2.6) in the PUM. We also scale all mentioned forces or displacements applied on peridynamics problems in time: We start the dynamic simulation with a load or prescribed displacement of zero and increase it linearly until the last time step. By this, we approximate a quasi-static peridynamic solution on the local problems and ensure that we do not lose nodes at the boundary due to excessive load. Discretization parameters are stated for each experiment, though we discretize all PUM simulations on

level six and use linear polynomials. Heaviside and crack front enrichments are applied whenever cracks are present and we use a direct linear solver in the PUM. All peridynamic simulations were done with the NLMech/PeriHPX code [20], all PUM simulations with PUMA [112, 120]. For comparison of the computed displacements, we define the maximal magnitude of the computed displacement by

$$U_{max} = \max_{i=1,\dots,n} (|u(t, x_i)|) \tag{5.6}$$

evaluated on all VTK nodes used for visualization.

5.3.1 *Uncracked Bar – Material Models*

In this first example, we investigate the compatibility of the peridynamics and the linear elastic PDE model for low loads and test the softening that is built into the peridynamics model (2.16). To this end, we stretch a two-dimensional bar by applying increasing levels of force on both ends, compare Figure 5.5.

Figure 5.5: Two-dimensional bar for comparing the peridynamics material model to linear elasticity. The bar is stretched by a force F applied on both sides, going from a linear response to softening in the peridynamics model for increased load.

We discretize the problem on level six in the PUM which results in a patch radius of $h_{PUM} = 7.81\times10^{-3}$m. The peridynamics discretization, on the other hand, has a node spacing of $h_{PD} = 5\times10^{-4}$m and a horizon size of $\delta = 4h_{PD}$. The peridynamics simulation is run up to time $T = 1\times10^{-3}$s with a time step size of $t_s = 2\times10^{-8}$s. While we only show results for the quasi-static PUM solution, an explicit dynamic simulation computes the exact same material response for this slow and simple load scenario.

We expect the two material models to coincide as long as the peridynamics model is in the linear regime of the double well potential (2.16), after which it softens and thus should predict larger displacements. To test this, we apply a base load of 9×10^5 Newton and increase it by up to a factor of twelve. Note that in the peridynamics model we scale the applied load up to that level over time and apply it as an external force density computed by (5.2), which adds to the differences between the two simulation models. Table 5.1 lists all resulting maximal displacements and the corresponding maximal damage.

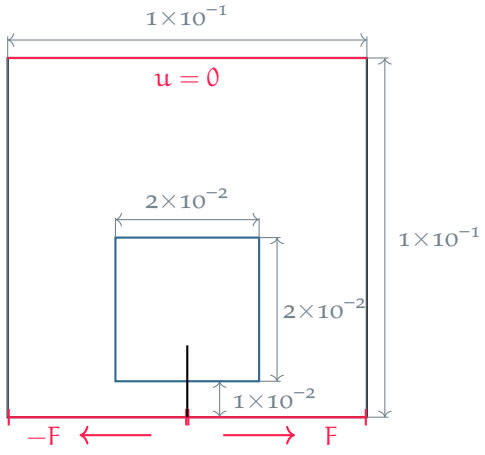
We can observe that the material responses indeed seem to correspond for low loads. Moreover, the peridynamics material model softens for increased load, as is also indicated by the increasing damage, with values above one implying nodes with softening bonds. From the perspective of the global-local method, we are thus confident that a local peridynamic problem computes displacements of comparable magnitude when loaded with boundary data from the global PUM problem.

Table 5.1: The maximal displacement magnitude U_{max} obtained by the PUM and by the peridynamic (PD) simulation of the two-dimensional bar example sketched in Figure 5.5. While the material responses agree initially, we can observe the softening of the peridynamic model for increased load.

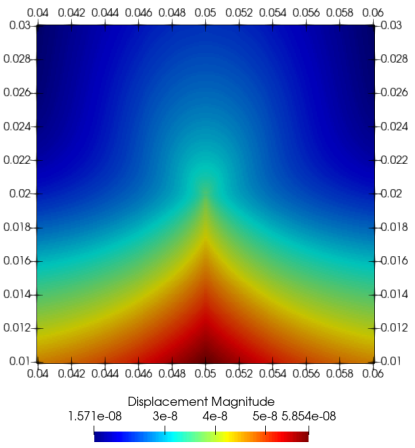
Load [N]	U_{max}–PUM	U_{max}–PD	Damage_{max}
9×10^5	1.24×10^{-4}	1.20×10^{-4}	7×10^{-2}
$4\cdot9\times10^5$	4.94×10^{-4}	4.89×10^{-4}	3.00×10^{-1}
$8\cdot9\times10^5$	9.88×10^{-4}	1.02×10^{-3}	6.30×10^{-1}
$12\cdot9\times10^5$	1.48×10^{-3}	1.68×10^{-3}	1.06

5.3.2 *Mode I Boundary Data Transfer*

In our next example, we examine the transfer of information from PUM to peridynamics on a pure mode I crack opening problem. This problem further studies the compatibility of the material models in the presence of a crack. Specifically, we run the first half of the global-local method by applying a global PUM solution as boundary data on a local peridynamics problem. The respective computational domains are sketched in Figure 5.6a.



(a) Mode I geometry in meters.



(b) Mode I local peridynamic solution.

Figure 5.6: Global and local problem domain for investigating the PUM to peridynamics boundary data transfer. The local domain contains half of the crack.

Compared to the last example, the compatibility of the underlying material models is further tested by the presence of a crack. We apply a tangential force of $F = 1\times10^3$N on the left and right of the crack, which is low enough to not produce crack growth so that the PUM and peridynamics simulations can agree on an equilibrium solution. The top of the global domain is fixed via Dirichlet boundary conditions. On the local domain, we apply the global boundary data as a prescribed displacement on a layer of peridynamic nodes of one horizon size. We again discretize the PUM problem on level six, which yields $h_{PUM} = 7.81\times10^{-4}$m on this geometry, and have a node spacing of

$h_{PD} = 5\times10^{-4}$m and a horizon of $\delta = 4h_{PD} = 2\times10^{-3}$m in the peridynamics discretization. The latter is run until time $T = 1\times10^{-3}$s with a time step size of $t_s = 2\times10^{-8}$s.

For the global-local method to work, local solutions by the PUM and peridynamics should be more or less interchangeable. To confirm this, we solve the global PUM problem and prescribe the computed displacement on both a local PUM and a local peridynamic problem and compare the results. Figure 5.6b shows the local peridynamics solution, whereas Figure 5.7 shows the global and local solution as computed by the PUM. Despite the presence of the crack, both solutions look similar qualitatively and agree in their maximal displacement.

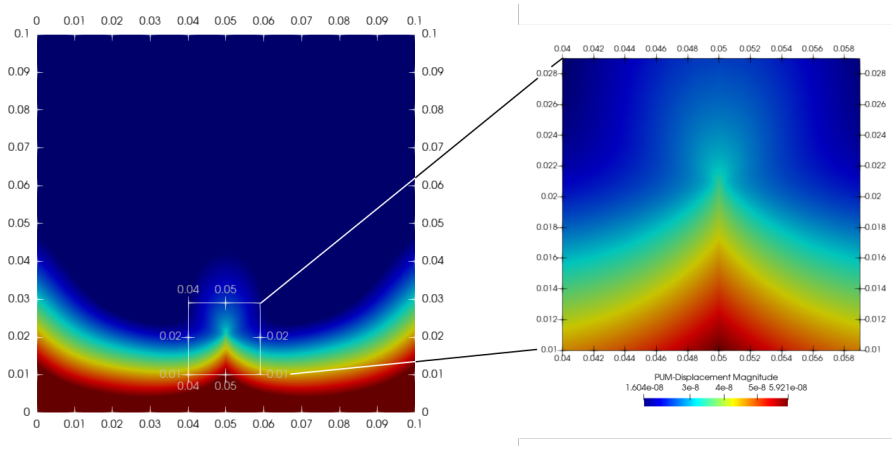


Figure 5.7: Global and local PUM solution of the mode I crack problem.

Moreover, we can use this example to highlight the differences in runtime between the two methods. To this end, we solve the global problem dynamically and quasi-statically with the PUM and with peridynamics for comparison. We further report the peridynamics runtime of the local problem. The results in Table 5.2 clearly show the expected runtime improvements when applying peridynamics only locally as it reduces the computational cost by a factor of eight. In this example, the dynamic global PUM simulation is even faster than the local peridynamic simulation, while having almost five times as much degrees of freedom. When factoring in that the PUM simulations were run on four cores whereas 40 where used for the peridynamic solutions, it is also evident why there is so much interest in combining peridynamics with computational more efficient methods.

5.3.3 *Coupled Crack Growth*

In this example, we apply the coupled global-local method to a simple crack growth problem. On a square with an initial crack through half of the domain, compare Figure 5.8, we pull on the top and bottom boundaries so that the crack grows through the domain. The local peridynamics domain contains the crack tip throughout the simulation. Moreover, we prescribe the global displacement on the top and bottom of the local problem only.

The material parameters are Young's modulus $E = 2\times10^{11}$ and Poisson's ratio $\nu = 1/3$. We apply a maximal displacement of $\bar{u} = \pm(0, 6.50\times10^{-6})$ at the

Table 5.2: Runtime and degrees of freedom (DOF) for simulations of the mode I problem. On the global problem, we measure the runtime of quasi-static and dynamic PUM, and peridynamic (PD) simulations. On the local problem, we report the runtime of the peridynamic simulation only to indicate the potential speed-up of the coupled method. Note that peridynamics solutions were computed using 40 cores, whereas the PUM simulations were run with only four.

Method	Time [s]	DOF
PUM quasi-static	1.00	2.51×10^4
PUM dynamic	1.38×10^2	2.51×10^4
Global PD	4.39×10^3	6.42×10^5
Local PD	5.22×10^2	4.36×10^3

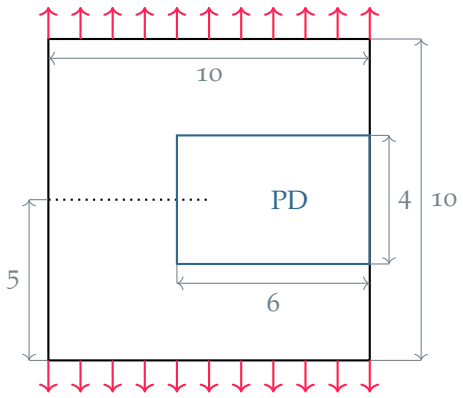
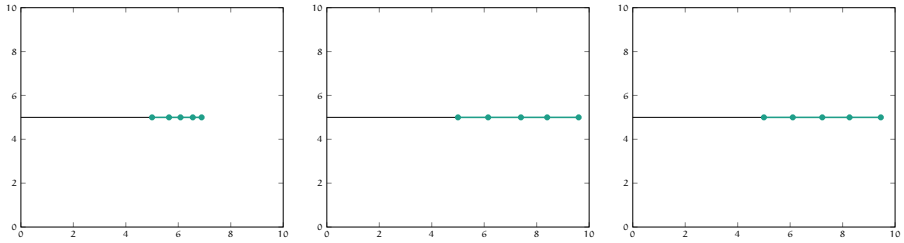


Figure 5.8: Geometry of the simple PUM with peridynamics (PD) coupling example in meters.

top and bottom of the global PUM problem, but scale it linearly over several load steps. In the local peridynamics problem, the crack starts growing with 86 percent of the final displacement applied. From the peridynamic solution, we extract the new crack path and pass it to the global PUM problem, which is rerun with two percentage points higher prescribed displacement at the boundaries. We iterate this and thus simulate with full data exchange between the two models. In fast dynamic processes, we expect to have more frequent interactions between the methods, but we can still conclude that the proposed global-local coupling method works as intended. The analysis of the interaction frequency between the two models and codes is a topic for future research.

We further use this example to measure the effect of different discretization widths on the local peridynamics problem. To this end, we choose three different node spacings h_{PD}^i with corresponding horizons δ_i on the local peridynamic problem while keeping the global discretization level constant. Figure 5.9 shows the respective node spacings and horizon sizes as well as the extracted crack paths from the three coupled simulations.



(a) $\delta_1 = 1/2$ and $h^1_{PD} = \delta_1/2$ (b) $\delta_2 = 1/4$ and $h^2_{PD} = \delta_2/4$ (c) $\delta_3 = 1/8$ and $h^3_{PD} = \delta_3/8$

Figure 5.9: Extracted crack path from the coupled global-local simulation for three different peridynamic discretization width. Results from a coarse peridynamic (PD) mesh are shown in (a), from a refined mesh in (b) and from a twice refined mesh in (c).

In the following, we compare the extracted crack tip positions as well as the corresponding global displacements, using the finest peridynamic solution as reference. Specifically, we compute the differences in crack tip positions

$$e_{tip}(\delta_i, t) = \left| \frac{crack\_pos(\delta_i, t) - crack\_pos(\delta_3, t)}{crack\_pos(\delta_3, t)} \right| \qquad i \in \{1, 2\} \tag{5.7}$$

and the corresponding differences between PUM solutions

$$e_u(\delta_i, t) := \sqrt{\frac{\sum_j \left(u_3(p_j) - u_i(p_j)\right)^2}{\left(u_3(p_j)\right)^2}} \qquad i \in \{1, 2\} \tag{5.8}$$

on points p_j of the visualization mesh at four time steps.

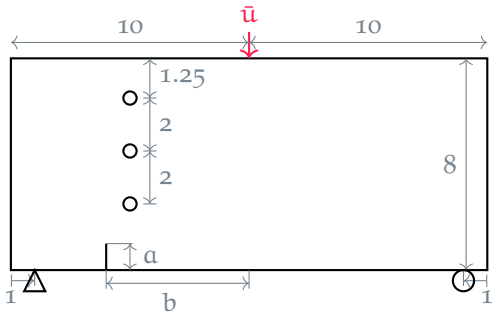
Table 5.3: The absolute difference of the crack tip positions, see Equation (5.7), with respect to the finest peridynamic discretization with $\delta_3 = 0.125$. And relative difference (5.8) between PUM solutions with the different peridynamic crack paths.

Time step	$e_{tip}(\delta_1, t)$	$e_{tip}(\delta_2, t)$	$e_u(\delta_1, t)$	$e_u(\delta_2, t)$
43000	7.40×10^{-2}	8×10^{-3}		
44000	1.50×10^{-1}	2.40×10^{-2}	1.20×10^{-1}	2.10×10^{-2}
45000	2.00×10^{-1}	1.60×10^{-2}		
46000	4×10^{-1}	1.50×10^{-2}	2.50×10^{-1}	2.20×10^{-2}

Looking at the results in Table 5.3, we can see that the coarsest peridynamic solution fails to capture the crack growth process as it computes much smaller crack increments. This can also be observed in Figure 5.9. As expected, big differences in crack tip positions produce big differences in the computed displacement. We therefore conclude that a sufficiently small mesh width is important in peridynamics, not only to predict accurate crack paths but also to allow for sufficiently fast crack growth.

5.3.4 *Three Point Bending*

As a first validation of the proposed coupling method against real world experimental data, we reproduce the three point bending experiments in [54]. These are frequently used to validate numerical methods, for example peridynamics in [93, 124], the phase-field model in [32, 84, 89, 142] and the GFEM in [91]. In the experiments, a plexiglass sheet supported at two points has its top center displaced downwards and features an initial crack offset to the left, compare Figure 5.10. The actual experiment was conducted with a three-dimensional specimen, but many of the above-mentioned references reproduce it with a two-dimensional simulation, as do we. This simplification yields sufficiently accurate results, because the thickness of the sheet at half an inch is only five percent of its length and the applied load induces no effects in the third dimension. The experiment further consists of several separate cases in which the initial crack position is varied and three holes are introduced into the geometry. We reproduce one case without and two cases with the presence of holes, compare Figure 5.10.



Case	a (in)	b (in)
1	1	6
2	1	6
3	1.5	5

Figure 5.10: Geometry of the three point bending experiment, all lengths in inches. Crack length and position varies across the three cases with the given parameters. Note that the three holes are not present in the first case.

The plexiglass sheet is modeled with linear elastic and isotropic material with Young's modulus $E = 2.40 \times 10^9 \text{GPa}$ and Poisson's ratio $\nu = 1/3$. Regarding boundary conditions, we apply a maximal displacement of $2 \times 10^{-3}\text{m}$ at the last load step in the top center of the domain. This is modeled as a point Dirichlet condition in the PUM simulation. We apply Dirichlet boundary conditions at two more points of the domain, where the specimen is supported: At the left bottom point, one inch away from the left boundary, we fix both x and y displacement to zero. At the right bottom point, one inch away from the right boundary, we only fix the y component to zero and thus allow horizontal movement of the domain here. We discretize the global PUM problem on level six with linear polynomials, but add one additional level around the crack and the holes. Along the crack, we apply a Heaviside enrichment and use the tip enrichments (3.3) on patches in a radius around the tip bounded by the distance to the next kink as the crack is grows. During crack growth, we thus apply further refinements near the tip, such that the patches there are smaller than the distance to the first kink in the crack path. We set up the local peridynamics problem in a region around the crack tip and move it with the propagating crack. The size of the local problem is roughly two by two inches. However, we increase the size for each case and load

step separately to ensure that holes are always fully contained inside the local domain if they intersect it. Throughout all cases and local domain sizes, we use a mesh width of $h_{PD} = 5\times10^{-4}$m, a horizon size of $\delta = 4\times10^{-3}$m and a time step size of $t_s = 2\times10^{-8}$s. Boundary data from the PUM is applied on a layer of two horizons along the local boundary. Again, we do not simulate the coupled method in time yet, but solve quasi-static crack growth. That is, we compute global PUM solutions with increasing load and approximate a local peridynamics equilibrium solution by slowly increasing the applied boundary data up to its maximum. We apply rather coarse load increments at five percent steps of the maximal load. For each load step, we extract the crack path from the local peridynamics solution. We then use the updated crack geometry for the next global PUM solution and also apply it as initial crack on the next local problem. Once the crack grows, the peridynamics damage reaches up to approximately 200 in the simulation and we consider only damage above 20 for the extraction of the crack path to be sure that the bond is actually 'broken' and not only softened. This moves the enforced sharp crack path a little behind the process zone and gives the peridynamics solution more room to model the crack. The actual necessity and influence of such an approach will have to be examined in future research. We presume that it also depends on the frequency at which we exchange information between the methods. Note that at the moment, we are not yet interested in the speed of the crack propagation but are only interested in the crack path.

In all three experiments, the crack grows successfully and the results are displayed in Figure 5.11. Next to the results of our coupling approach, we present the crack paths obtained in the physical experiments in [54]. For further comparison, we add the results produced by [93], which were computed using a FEM-peridynamics coupling method. In this method, both models are coupled across a transition region, and locally, the overall solution is given by the peridynamics solution exclusively. Additionally, and different to our approach, the peridynamic region spans the whole area in which the crack grows and is not moved. The mesh width of $h = 6.35\times10^{-4}$m in their peridynamics region is comparable to ours, though.

Both coupling approaches qualitatively capture the correct crack path: We obtain the same crack growth direction and the crack grows into the same hole as in the physical experiment. Looking at the details, though, we do have slight differences in accuracy between the three cases. The first case is the least difficult as no holes are present in the domain. There, our coupling method predicts more or less the reference crack path in the first half of the growth process. In the second half, we end up slightly right of the reference. Unfortunately, we cannot present results from [93] for this case, as we extracted the paths from the images presented in the paper: Due to the missing holes in this case, we had not enough reference locations in the picture to extract the crack path. In case two, which introduces the holes, our crack path is again slightly right of the reference but still fairly close. The crack path from the FEM coupling is slightly left of the reference, but predicts the entry point in the second hole more accurately. Such small differences can easily be the result of small errors during extraction of the crack path, since we also prescribe the extracted path as initial crack in the next load step. Once we add automatic crack path extraction to our implementation, we will have to revisit this issue and see if we can improve upon the presented

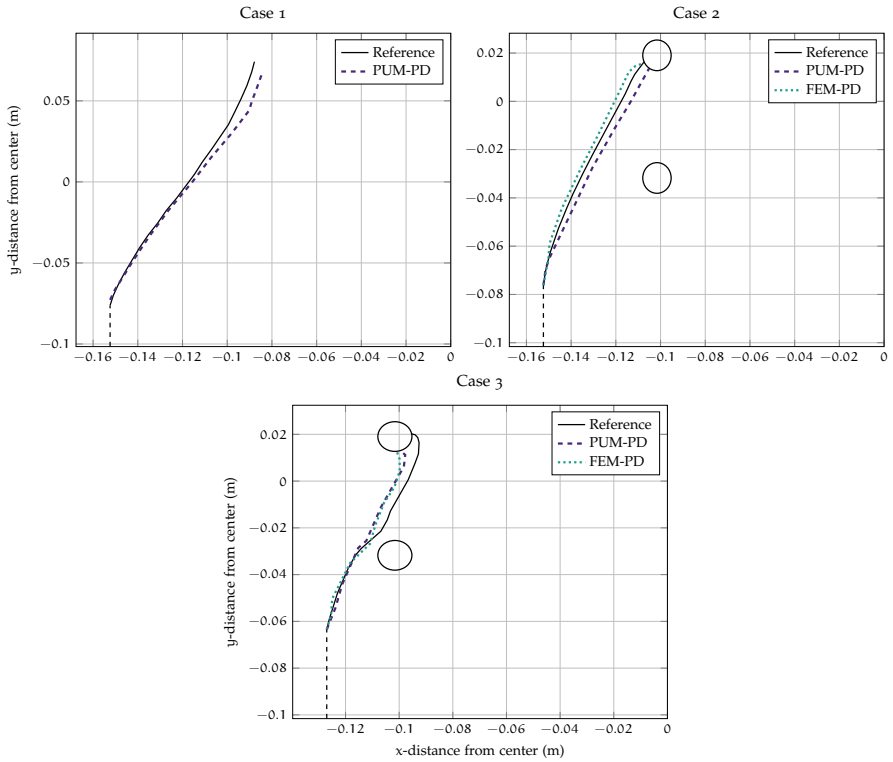


Figure 5.11: Crack paths in the three cases of the three point bending example. The reference is taken from physical experiments [54]. Dashed lines indicate the results of the global-local coupling of the PUM with peridynamics (PUM-PD). Dotted lines indicate the crack paths obtained by the FEM with peridynamics coupling (FEM-PD) in [93].

results. The last case is the most difficult of the three, since the initial crack is located closer to the holes, which influence the crack path. Both coupling approaches seem to underestimate the influence of the first hole because the crack paths are not pulled as far to the right as in the physical experiment. Consequently, the simulated crack paths reach the second hole to the left of the correct entry point. The overall shape of the paths still resembles the reference, though. We therefore still consider it a success that our global-local coupling predicts the same crack path as the direct coupling approach. Even more so as our peridynamics region is around a quarter the size of their region and moves with the propagating crack. In the FEM coupling, in contrast, both holes are contained in the peridynamics region during the whole simulation.

We started all three simulations at 75 percent of the final load and, as mentioned before, added five percent points every load step. The crack reached the hole at 100 and 95 percent of the final load in case two and three, respectively. However, in case one, several more steps at the final load were necessary to approach the desired length of the crack. This may be due to the mesh being too coarse here, as we already observed in Section 5.3.3. In general, we can conclude that the interaction frequency of the two methods, in applied load or in time steps, must be sufficiently high in order for the global-local coupling to work: Since the prescribed displacement on the boundary of the local problem is

constant for each load step, the crack propagation of the local model is limited. In a coupled simulation in time, however, the possibility to easily have multi-rate time integration is an advantage of the global-local coupling approach.

In summary, we believe that these experiments demonstrate the validity of the presented global-local coupling method, especially with peridynamics. To arrive at a fully applicable method, further research should investigate the coupling in time of the two models as well as identify a robust implementation of automatic crack path extraction.

5.3.5 *Phase-Field Coupling*

Since coupling via the global-local method is not limited to local peridynamic problems, we present initial results of coupling a global linear elasticity problem with a local phase-field model, both discretized by the PUM and simulated in PUMA. We simulate classical linear elasticity on the global problem and adopt the phase-field model from [90] on the local problem, which models brittle fracture. That is, on the local problem we iteratively solve a nonlinear equation for the phase-field and a linear elasticity PDE that has the phase-field solution incorporated to alter the material response.

The specific problem is taken from [146] and features a hundred by sixty rectangle that is initially cracked from the left to the center, similar to the problem in Section 5.3.3. The material has Young's modulus $E = 2.31\times10^{5}$, Poisson's ratio $\nu = 4.29\times10^{-1}$ and critical energy release rate $G_c = 2.70$. The length parameter in the phase-field model characterizing the width of the crack is set to $l = 5\times10^{-3}$. We solve the global problem on level five and the local on level seven, as the phase-field model requires a finer resolution to produce a narrow crack path. In an outer loop, we apply increasing displacements on the top and bottom of the domain with an initial displacement of 3.20×10^{-1} and decreasing increments down to 1×10^{-2}, which cracks the geometry completely after twelve steps. In each load step, we solve two global-local iterations and at most four staggered iterations, unless the change in the displacement and phase-field solutions is less than 1×10^{-2}. We employ an LU-factorization as linear solver both on the global and local linear elasticity problems and in the Newton solver of the local phase-field problem. To allow the phase-field model to handle the crack tip zone, we model the crack on the local problem by a Heaviside function only, but use both enrichments on the global problem.

Figure 5.12 shows the global and local solutions warped by their displacements in time steps four, nine, eleven and twelve, where the crack reaches the right boundary. We can observe that damage is only high ahead of the crack tip, as the already fully cracked part is modeled by the Heaviside function. Furthermore, the damaged area ahead of the crack is wider than it would be in linear elastic fracture mechanics because the phase-field solution models the entire area in which the material exhibits stress, instead of limiting it to the crack tip. We want to highlight here that the crack branches close to the right boundary with the phase-field model, which is impossible to predict in linear elastic fracture mechanics. However, this behavior vanishes once we apply the Heaviside enrichment too close to the front of the damaged area. The interaction of the crack enrichments with the phase-field model should therefore be investigated further to propose a suitable enrichment scheme for the coupled

method. For example, with a small local problem around the crack tip, it may be desirable to update the crack path only when the local region moves.

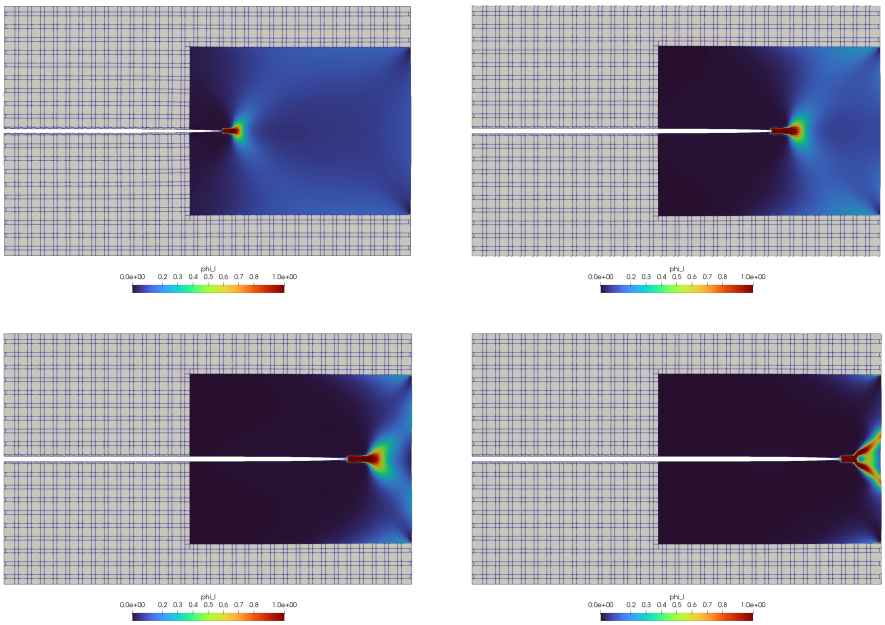


Figure 5.12: Local phase-field solution with global and local problem warped by their respective displacements for simulation steps 4, 9, 11 and 12. The phase field ranges from zero to one.

Coming back to our results, updates of the crack path are performed automatically based on the 0.98 isoline of the phase-field solution. To visualize this, Figure 5.13 plots the extracted 0.8, 0.9 and 0.98 isolines and the phase-field close to the crack tip for the same four load steps as in Figure 5.12. As can be observed there, we update the crack tip to the x-coordinate where the damage field starts to widen, which we assume to coincide with a kind of process zone. Whether this is the correct and physical approach, still has to be verified, though. In general, several edge cases have to be understood and implemented for the robust automatic extraction of crack paths for the coupling method. One topic here is branching and coalescence of cracks, which was demonstrated to be feasible in [90]. Small bubbles of high damage that appear disconnected from the overall crack path constitute another possible challenge, as can be noticed in Figure 5.13.

To finish this section, we simulate the growth process of the penny shaped crack introduced in Section 3.3.4 with the phase-field method discretized by the PUM in three space dimensions. We apply the same phase-field model as before and discretize it on level six with two additional refinements towards the crack front. On the linear elasticity problem, the initial crack configuration is modeled by a Heaviside function and crack growth is exclusively handled by the phase-field solution. Boundary conditions are slightly different here, because instead of a load, we apply an increasing displacement on the top and bottom of the domain. The domain is further downsized to a box with radius 0.3 to reduce the computational cost, since a fine resolution is required to arrive at a

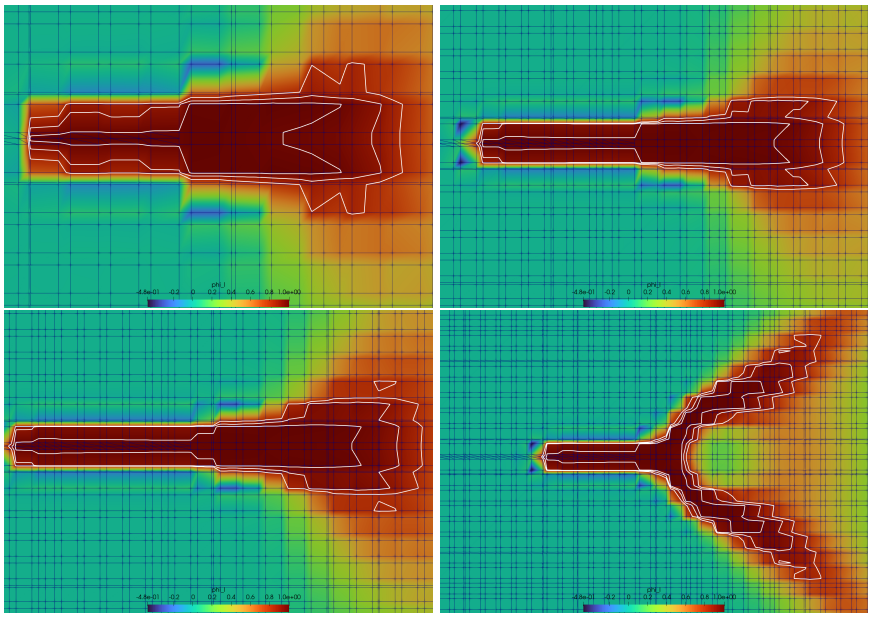


Figure 5.13: Iso-lines 0.8, 0.9 and 0.98 around local phase-field solutions for simulation steps 4, 9, 11 and 12. Compared to Figure 5.12, we applied appropriate zoom levels to each step individually to highlight the process zone.

reasonable width of the damage distribution. We set no boundary conditions on the phase-field problem. Figure 5.14 shows the 0.95-contour of the phase field in load step eleven and eighteen, together with a slice through the domain that is warped by the computed displacement. The predicted crack path looks fine though slightly different than what we computed in Section 3.3.4 due to the different boundary conditions. Applying the global-local method to this and similar three-dimensional fracture problems should be straight forward, but is left for future work.

We conclude that combining the PUM with the phase-field method through global-local coupling is also possible. This allows to use the variety of phase-field material models [74] to predict crack growth in a cheap global simulation of linear elasticity. For the extension to three space dimension, only investigating the robust extraction of the crack geometry should be necessary.

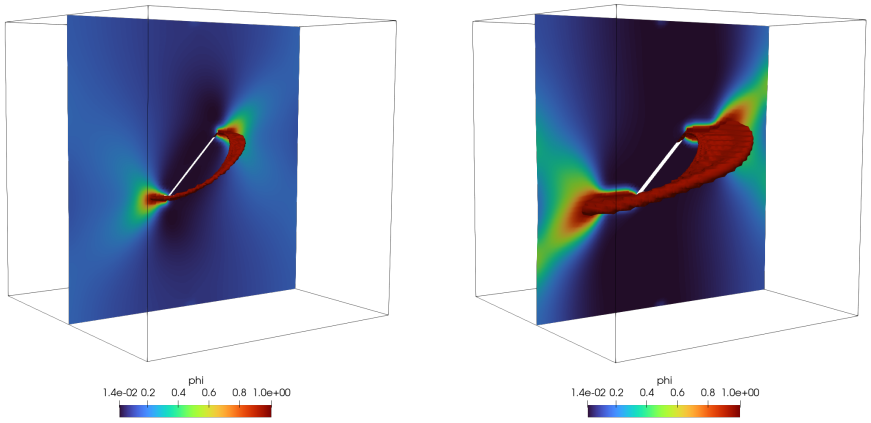


Figure 5.14: Phase-field solution of the three-dimensional penny shaped crack problem of Section 3.3.4 visualized on a slice through the domain. The phase field ranges from zero to one and we show its 0.95-contour to depict the crack geometry.

6 PRACTICAL USE CASES

Before concluding this thesis, we want to take the opportunity to highlight new research topics in the PUM and with PUMA, that are enabled by the presented work in this thesis. Most notably this includes the new possibility to simulate fracture in three space dimensions. Additionally, other localized features than fracture that can also benefit from the application of the global-local method. It should be noted, that we present only first results but no in-depth study of the presented topics.

LAMINATED COMPOSITES

Failure analysis of laminated composites is of high interest for example in the aerospace industry, where damage might be present due to manufacturing defects or occur through repeated loading [53], at bolted joints [82] or through impact [98, 147]. Damage propagation here includes areas with matrix cracks and delamination zones between material layers. The global-local method studied in this thesis could be used to speed up numerical simulation of such delamination processes, by condensing degrees of freedom in localized crack clusters. Furthermore, the implementation work done in PUMA during this thesis lays the foundation for future work in this area, as now simulation of complex crack geometries in three space dimensions is possible.

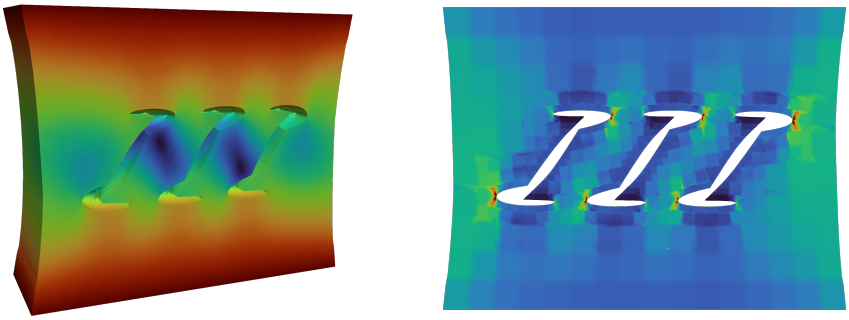


Figure 6.1: Triple h-cracked coupon warped by computed displacement and colored by displacement magnitude (left) and von Mises stress (right).

For example, Figure 6.1 shows a cluster of matrix cracks through a material layer with beginning delamination above and below, and Figure 6.2 shows a cluster of penny shaped delamination zones between material layers. These are simplified examples with layers of linear elastic material, where we used the bimaterial enrichment functions (3.5) at crack fronts between layers. Furthermore, atypical loading for this application is applied: In the presented examples, we stretch the material, while in practice bending or compression are of more interest. To this end, however, only contact between the crack surfaces would have to be implemented.

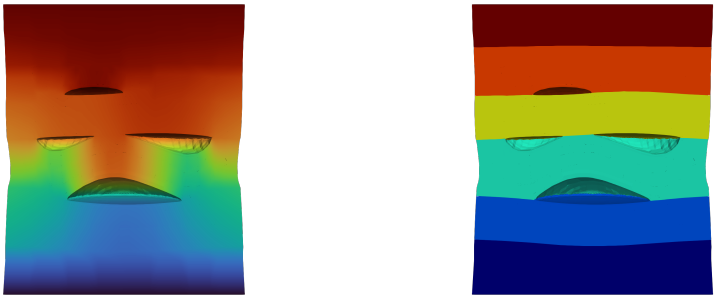


Figure 6.2: Delaminating penny cracks with alternating hard and soft linear elastic material layers. Warped by computed displacement and colored by displacement magnitude (left) and material layers (right).

PRESSURE DRIVEN CRACK GROWTH

Throughout this thesis, we studied fracture propagation caused by the application of external loads on the cracked domain. However, crack growth can also be driven by internal pressure from inside the crack. In the numerical simulation of this process, a second PDE or model has to be solved inside the crack and interacts with the outer simulation through the boundary given by the crack surface. As we implemented three-dimensional crack surface handling and crack growth for this thesis, this would be a natural extension of the current capabilities of the software framework PUMA.

After implementing this, the following processes could be simulated with PUMA. While freezing, water in fractured rock expands and applies frost-heaving pressure to the surrounding material. This process has, for example, already been successfully simulated with the XFEM [148]. In hydraulic fracturing artificial cracks or perforations are created along a wellbore through which a viscous fluid is pumped to drive crack growth through the pressure. Simulations of this system of coupled PDEs are of interest to understand the effects of varying relative crack locations and fluid viscosities and the GFEM has already been successfully applied to this class of problems [92, 122, 123]. Furthermore, peridynamics has also been successfully applied to the simulation of hydraulic fracturing [100], such that the coupling with peridynamics, presented in the last chapter, could be applied to this problem, too. A similar process is the extraction of geothermal energy from the ground. In fact, a hydraulic fracturing model has already been applied in this context [145], to study the effects of fracture morphology on heat extraction numerically.

As an example, we present a crack growth simulation with an initial crack configuration taken from [92], where fracture propagation in hydraulic fracturing is studied with the XFEM. We did not implement solving the fluid equation in arbitrary three-dimensional crack geometries for this thesis, but we show preliminary results with a single equation of linear elasticity and rather atypical loading for this application, as we stretch the domain. That is, we essentially repeat the example shown in Section 3.3.4 with three penny shaped cracks next to each other. As sketched in Figure 6.3, we simulate three vertical penny shaped cracks in a box with radius $L = 1$ of isotropic linear elastic material with Young's modulus $E = 1.00 \times 10^3$ and Poisson's ratio $\nu = 1/3$. After some crack

growth steps we increase the domain radius to minimize finite domain effects. The cracks all have an initial radius of 1×10^{-1} and are placed with a distance of 1×10^{-1} next to each other. On the sides of the box we pull with unit stress $\sigma = 1$ to force crack growth.

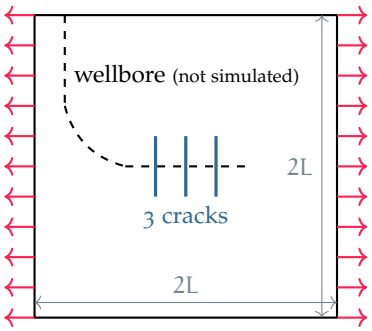


Figure 6.3: Side view of unit cube with three penny shaped cracks around a wellbore as in hydraulic fracturing. Wellbore and fluid are not simulated and boundary conditions are applied to force crack growth.

Figure 6.4 shows the initial crack configuration and discretization. We discretize on level four and apply three levels of h-refinement towards the crack fronts as well as further h-refinement to ensure that every patch contains at most one crack front vertex. For the cracks, we use the crack enrichments described in Section 3.1 and employ a direct solver as the resulting linear system is small enough. We extract stress intensity factors using the DCM, see Section 2.2.2 and update the crack geometries based on Schöllmanns criterion, see Section 2.2.3.

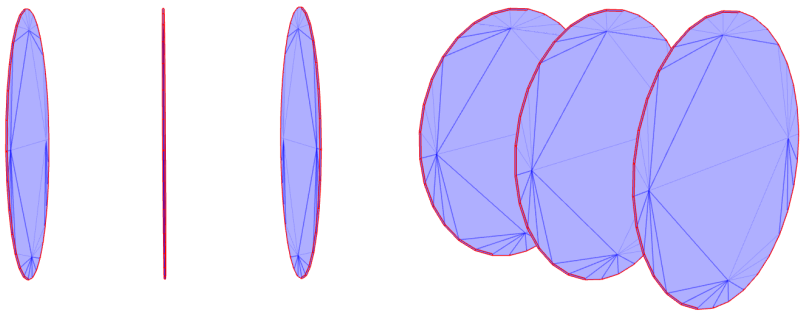


Figure 6.4: Initial crack configuration of the hydraulic fracture example.

Figure 6.5 shows the crack configuration after 22 growth steps. As expected, the middle crack stays flat while the other two bend away from it. Results with crack growth driven by fluid pressure and isotropic remote stress in this crack configuration show similar results, compare [92]. While time did not permit to further explore this application, the simulations should be possible once the following two features are implemented: Simulation of the fluid pressure inside the crack geometry and the possibility to apply boundary conditions on the crack surface to use that pressure in the simulation of the materials.

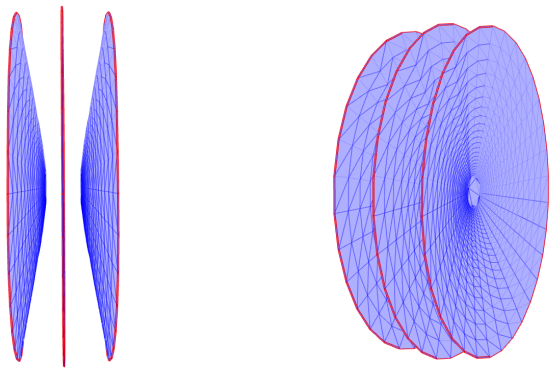


Figure 6.5: Crack configuration of the hydraulic fracture example after 22 growth steps.

LASER HEAT SOURCES

While we focused on cracks as the defining features for local problems in this thesis, another common problem are localized heat sources such as lasers. In [97] for example, the global-local method is applied to problems in which a linear elastic material is thermally loaded by a sharp gaussian laser source. Figure 6.6 shows a reproduction of one of the examples.

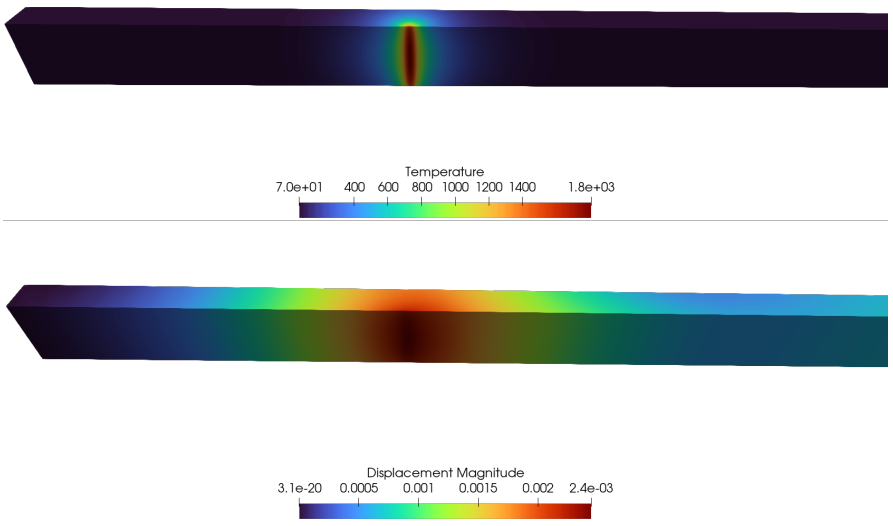


Figure 6.6: Temperature distribution induced by a laser heat source (top) and material displacement resulting from the thermal stress (bottom).

As with fracture, the degrees of freedom required to resolve the sharp thermal gradient are moved to the local problem, which allows the global problem to be discretized with h larger than the laser width. Similar to fracture propagation, movement of the laser source would require remeshing in direct solutions, which again is circumvented by simply moving the local problem and computing a new

enrichment. Localized laser heat sources are for example present in the process of additive manufacturing [108], where a laser melts a layer of metal powder which then iteratively builds the structure on solidification. Numerical simulation can help to understand and optimize this process. In such a simulation, we expect further runtime improvements from the application of the global-local method, since the involved material model is nonlinear. Thus, the solver is more expensive and we have more opportunity to benefit from the reduced number of global degrees of freedom. Furthermore, several lasers can be used to build a single structure introducing an opportunity to parallelize across several local problems.

CRACKED SHELL COUPLING

Thin geometries, where one physical dimension is much smaller than the other two, such as metal sheets, can be more efficiently simulated with reduced two-dimensional models, which incorporate assumptions on the through-the-thickness behavior of the deformation. Those assumptions, however, may not hold in certain situations, for example in the vicinity of a crack tip. In such a case, we may combine two- and three-dimensional models for different regions of the material, as shown in Figure 6.7.

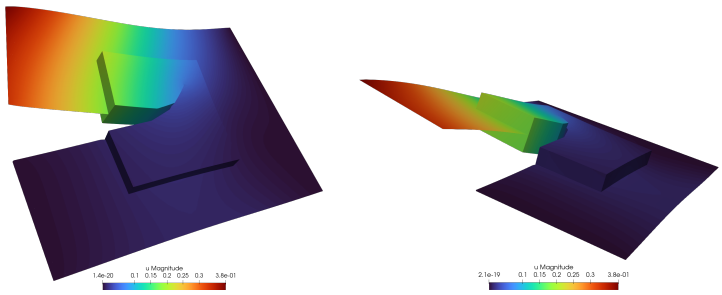


Figure 6.7: Solution of a simple two to three dimensions coupling example with a crack.

Here, we combine the two-dimensional Kirchhoff-Love plate model with three-dimensional linear elasticity in a subdomain including the crack tip. We couple the two displacement fields with Nitsche's method [43]. The crack is represented by the usual crack enrichments in both models and crosses the interface. In the three-dimensional subdomain, crack growth out of the plane perpendicular to the plate is then possible. One could either study this as a technique to accelerate the simulation of three-dimensional crack growth, by modeling parts of the domain only in two space dimensions, or as the opportunity to introduce fracture with complex crack geometries to the simulation of shells.

7 CONCLUDING REMARKS

In this thesis, we demonstrated that the global-local enrichments method in a flat-top partition of unity method can improve linear elastic fracture simulation in two ways: Either by offloading the degrees of freedom necessary for the resolution of cracks in linear elastic fracture mechanics on separate problems or by enabling the use of expensive models with natural crack initiation and propagation only locally. We achieved this in three steps. First, we adopted the simulation of complex three-dimensional fracture and fracture propagation in the PUM in Chapter 3 and implemented it in the software framework PUMA. Second, we adapted the global-local method from the finite element and thus mesh based GFEM^{gl} setting into the flat-top PUM in Chapter 4. And finally, we generalized global-local enrichments to allow a different model on the local problem and presented results using peridynamics or a phase-field model locally in Chapter 5. Specifically, we introduced and implemented the following.

To enable the simulation of crack surfaces in three space dimensions in PUMA, we first of all implemented a representation of the crack geometry by triangular meshes. This is used to generate integration cells that respect the crack geometry. The generation builds on the fact that in a first pass rectangular integration cells resolving the patch boundaries are created efficiently and locally to each processor based on the construction of the underlying cover. Thus, tetrahedrons resolving the crack geometry can then be created locally for each crack intersecting rectangular cell, using a constrained delaunay construction. The representation further serves the evaluation of the enrichment functions which model the crack in the simulation. Here, we made sure to implement all necessary queries on the crack geometry efficiently, since they are called for each integration point or cell in the evaluation of the front enrichments or Heaviside type functions, respectively. Regarding the implementation of the front enrichments, a question is whether and how to adapt to kinks in the crack front, as the type of the singularity is not known analytically there. To this end, we implemented and benchmarked four approaches to handle those kinks in the evaluation of the front enrichments. Yet, our experiments in Section 3.3.3 confirm the findings in [102] that accuracy is above all determined by the local resolution in terms of h-refinement of the domain discretization and not by additional interpolation along the crack front. Another feature required especially for crack growth in linear elastic fracture mechanics was the extraction of stress intensity factors around the crack front. To this end, we implemented the CIM and DCM and demonstrated their ability to accurately extract the stress intensity factors in three space dimensions in the PUM. Of course, both methods were implemented to work in parallel by distributing crack front segments according to their position in the overall cover patch distribution. Here, special care had to be taken to evaluate inside the current processes patches. For the CIM we further implemented integration cells for a one dimensional path around crack front segments and demonstrated the path independence of extraction in numerical experiments. To predict the crack growth direction based on the

extracted stress intensity factors, we implemented Schöllmann's criterion and added algorithms to update the mesh representing the crack geometry during crack growth simulations.

In order to speed up fracture simulation by moving the degrees of freedom required to resolve the crack singularity onto a separate problem, we adapted the global-local enrichments method first introduced in the context of the GFEM^{gl} [23] to our flat-top and meshfree PUM. The central challenge here was that in the PUM h-refinement generates non-nested ansatz space due to the construction of the underlying cover. This prevents efficient numerical integration of overlapping computed enrichment in parallel the way it was proposed in the GFEM^{gl}. To overcome this limitation, we presented an efficient parallel implementation of the global-local method in the meshfree, flat-top PUM, based on the following decisions. Compared to the implementation in the GFEM^{gl} we restricted ourselves to non-overlapping local problems and we thus have only one local problem per feature. We, however, solve each local problem in parallel and exploit the parallelism inherent in the global-local method by solving each local problem on only a subset of the overall available processes. In consequence, several parallel local problems can be solved simultaneously. To this end, several changes were necessary in PUMA. We implemented the search of neighbor relationships between function spaces on different communicators. Moreover, we added an integration cell handler that allows to use the local problems' integration cells in order to take advantage of their parallel distribution. In the global cells, a whole local problem could be contained inside the region of a single process. Lastly, we implemented the automatic setup and distribution of local problems based on the global cover in parallel. The parallel efficiency of the resulting method is excellent, especially its weak scaling, as we demonstrated in numerical experiments. When adding local features to a simulation, it achieves constant runtime by adding only a flat number of processes for each. We want to emphasize here that our implementation not only allows the performant application of the global-local method in the PUM, but also addresses the challenges introduced by a distributed memory parallel implementation on top of the shared memory parallel implementation presented in the GFEM^{gl} [66]. Regarding the accuracy of the global-local method, we confirmed that after sufficient global-local iterations to improve the boundary data on the local problem, it achieves comparable accuracy to applying the local degrees of freedom directly on the global problem by h-refinement. As such, it is a viable alternative to directly refined simulations, with runtime tradeoffs favoring the global-local method given enough local features, as suggested by our runtime estimates and experiments. We further analyzed the effect of a bufferzone and Robin boundary conditions on the quality of the boundary data and confirmed that both can reduce the number of required iterations for a targeted accuracy. With regard to Robin boundary conditions, we were further able to confirm the optimality of our scheme (4.11) to compute the parameter κ, which we first introduced in [9] and now applied in three space dimensions, at least on simple problems. Here, the underlying assumption is that the parameter is more sensitive to the energy in the global solution along the local boundary than on the discretization level, as was assumed in [68]. However, we also discovered an example that does not have a single optimal value of κ depending on the feature of the solution we are interested in.

As a third contribution of this thesis, we investigated the coupling opportunity in the global-local method where the local problem does not have to use the same model or discretization as the global, but merely has to accept boundary data from it and be able to provide some form of enrichment in return. We thus proposed the global-local as a coupling method and presented first results of coupling the PUM with peridynamics and a phase-field model. Both offer natural crack initiation, growth and branching, which is not possible in linear elastic fracture mechanics. As it is unclear whether the local solutions minimize the global linear elastic energy due to the different underlying material models, we proposed to only use the extracted crack path as enrichment, which then is modeled by the standard crack enrichments on the global problem. Crack path extraction has to be performed anyways once the local problem moves with the crack tip or front. The fundamental issue here is that in peridynamics and phase-field models the crack geometry is given only implicitly and at most up to around the discretization width. Hence, it is fuzzy and not sharp. While several algorithms for automatic crack path extraction from peridynamics and phase-field models exist in the literature [33, 90, 138], their robustness has to be investigated in future research. We want to note that the proposed global-local coupling method is similar in nature to the XFIELD [36], which combines the XFEM with a local phase-field model, except that they apply additional boundary conditions on the phase field.

We presented initial results of combining global linear elasticity with a local phase-field model, both discretized by the PUM, and automatic crack path extraction based on isolines of the phase field. Primarily, however, we presented results from the ongoing collaboration with P. Diehl and R. Lipton on coupling the PUM with peridynamics. In addition to a summary of results of our first publication on that subject [8], we presented new, so far unpublished experiments validating the proposed method against real world experimental data. The primary difficulty introduced by coupling linear elasticity with peridynamics are the different material models. To this end, we demonstrated the compatibility of both models sufficiently far away from the crack in numerical experiments. A second challenge, for the application of boundary conditions, is the non-locality of the peridynamics model, which made minor modifications of the global-local method necessary. In numerical experiments of increasing difficulty, we then demonstrated that exchanging information, i. e. boundary data and crack path, between the two methods is possible and produces the desired results. In one example, we also confirmed that our approach realizes the expected runtime improvements from localizing peridynamics. Especially the last comparison against real world experiments demonstrates the validity and potential of the proposed coupling method, but several challenges have to be addressed in future research. First of all, coupling the dynamic formulations of the two models in time has to be investigated. In particular, reasonable time step sizes for each method as well as the synchronization frequency have to be determined. Second, full coupling of the simulation codes, especially in parallel, needs to be implemented to speed up future research. To this end, a robust automatic algorithm for crack path extraction from the peridynamic damage field has to be determined and implemented. Currently, the most promising candidate seems to be the θ-simplified medial axis algorithm [10]. Third, constructing shape functions from the peridynamics solution that model the discontinuity across

the crack could be investigated as an alternative to crack path extraction, as was suggested in [119]. Lastly, on the PUM side, we have to determine a reasonable crack update length and implement a stable time integrator for the evolving ansatz space in crack growth. Otherwise, shock waves can be introduced due to a sudden increase in crack length. Overall, and based on the promising results presented in this thesis, we believe that further developing global-local enrichments as a coupling method provides an excellent research opportunity.

Regarding further future research, we first of all want to highlight the new possibilities enabled by the support of three-dimensional fracture now implemented in PUMA. In Chapter 6, we already listed several problems to be looked at in the future, which can also benefit in runtime by the application of the global-local method. Within the global-local method, it could be worthwhile to look into linear solvers that take advantage of the update structure of the global linear system over global-local iterations, as presented in [67]. Furthermore, constructing the local domain as the union of the global, enriched patches could be investigated to better adapt the shape of the local problem to the feature it tries to resolve. Another topic is using the global-local method as means to improve enrichments in a simulation. For example, precomputed enrichments from the approach presented in [2] may need to be adapted to the specifics of the current simulation, such as material parameters, and could thus be improved by the application of global-local iterations. A second source of suboptimal enrichments could be functions proposed by machine learning algorithms for a given simulation. From the perspective of the global-local method these would improve the quality of the boundary data in the first iteration.

BIBLIOGRAPHY

[1] R. G. Ahangar and Y. Verreman. "Assessment of mode I and mode II stress intensity factors obtained by displacement extrapolation and interaction integral methods." In: *Journal of Failure Analysis and Prevention* 19 (2019), pp. 85–97.

[2] I. Babuska and R. Lipton. "Optimal local approximation spaces for generalized finite element methods with application to multiscale problems." In: *Multiscale Modeling & Simulation* 9.1 (2011), pp. 373–406.

[3] I. Babuška and U. Banerjee. "Stable generalized finite element method (SGFEM)." In: *Computer methods in applied mechanics and engineering* 201 (2012), pp. 91–111.

[4] I. Babuška and J. M. Melenk. "The partition of unity method." In: *International journal for numerical methods in engineering* 40.4 (1997), pp. 727–758.

[5] É. Béchet, H. Minnebo, N. Moës, and B. Burgardt. "Improved implementation and robustness study of the X-FEM for stress analysis around cracks." In: *International journal for numerical methods in engineering* 64.8 (2005), pp. 1033–1056.

[6] T. Belytschko and T. Black. "Elastic crack growth in finite elements with minimal remeshing." In: *International journal for numerical methods in engineering* 45.5 (1999), pp. 601–620.

[7] T. Belytschko, Y. Y. Lu, and L. Gu. "Element-free Galerkin methods." In: *International journal for numerical methods in engineering* 37.2 (1994), pp. 229–256.

[8] M. Birner, P. Diehl, R. Lipton, and M. A. Schweitzer. "A fracture multiscale model for peridynamic enrichment within the partition of unity method." In: *Advances in Engineering Software* 176 (2023), p. 103360.

[9] M. Birner and M. A. Schweitzer. "Global-Local Enrichments in PUMA." In: *International Workshop on Meshfree Methods for Partial Differential Equations*. Springer. 2017, pp. 167–183. DOI: 10.1007/978-3-030-15119-5_10.

[10] H. Blum. "Models for the perception of speech and visual form." In: *ch. A transformation for extracting new descriptors of shape* (1967), pp. 362–380.

[11] B. Bourdin, G. A. Francfort, and J.-J.Marigo. "Numerical experiments in revisited brittle fracture." In: *Journal of the Mechanics and Physics of Solids* 48.4 (2000), pp. 797–826.

[12] B. Bourdin, G. A. Francfort, and J.-J.Marigo. "The variational approach to fracture." In: *Journal of elasticity* 91 (2008), pp. 5–148.

[13] D. Braess. "Theory, fast solvers, and applications in elasticity theory." In: *Finite Elements*. Cambridge University Press Cambridge, 2007.

[14] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*. Vol. 3. Springer, 2008.

[15] R. P. Brent. *Chapter 4: An algorithm with guaranteed convergence for finding a zero of a function, algorithms for minimization without derivatives*. 1973.

[16] M. Cervera, G. B. Barbat, M. Chiumenti, and J.-Y. Wu. "A comparative review of XFEM, mixed FEM and phase-field models for quasi-brittle cracking." In: *Archives of Computational Methods in Engineering* 29.2 (2022), pp. 1009–1083.

[17] E. Chahine, P. Laborde, and Y. Renard. "Crack tip enrichment in the XFEM using a cutoff function." In: *International journal for numerical methods in engineering* 75.6 (2008), pp. 629–646.

[18] M. D'Elia, X. Li, P. Seleson, X. Tian, and Y. Yu. "A review of local-to-nonlocal coupling methods in nonlocal diffusion and nonlocal mechanics." In: *Journal of Peridynamics and Nonlocal Modeling* (2021), pp. 1–50.

[19] P. Diehl. *EMU-nodal discretization*. 2020. DOI: 10.6084/m9.figshare.12320726.v1.

[20] P. Diehl, P. K. Jha, H. Kaiser, R. Lipton, and M. Lévesque. "An asynchronous and task-based implementation of peridynamics utilizing HPX—the C++ standard library for parallelism and concurrency." In: *SN Applied Sciences* 2.12 (2020), p. 2144. ISSN: 2523-3971. DOI: 10.1007/s42452-020-03784-x. URL: https://doi.org/10.1007/s42452-020-03784-x.

[21] P. Diehl, R. Lipton, T. Wick, and M. Tyagi. "A comparative review of peridynamics and phase-field models for engineering fracture mechanics." In: *Computational Mechanics* (2022), pp. 1–35. DOI: 10.1007/s00466-022-02147-0.

[22] C. A. Duarte, I. Babuška, and J. T. Oden. "Generalized finite element methods for three-dimensional structural mechanics problems." In: *Computers & Structures* 77.2 (2000), pp. 215–232.

[23] C. A. Duarte, D.-J. Kim, and I. Babuška. "A global-local approach for the construction of enrichment functions for the generalized FEM and its application to three-dimensional cracks." In: *Advances in meshfree techniques*. Springer, 2007, pp. 1–26.

[24] C. A. Duarte and J. T. Oden. "An hp adaptive method using clouds." In: *Computer methods in applied mechanics and engineering* 139.1-4 (1996), pp. 237–262.

[25] T. Elguedj, A. Gravouil, and H. Maigre. "An explicit dynamics extended finite element method. Part 1: mass lumping for arbitrary enrichment functions." In: *Computer Methods in Applied Mechanics and Engineering* 198.30-32 (2009), pp. 2297–2317.

[26] E. Emmrich and O. Weckner. "On the well-posedness of the linear peridynamic model and its convergence towards the Navier equation of linear elasticity." In: *Communications in Mathematical Sciences* 5.4 (2007), pp. 851–864.

[27] C. A. Felippa. "Introduction to finite element methods." In: *University of Colorado, USA* (2001).

[28] G. A. Francfort and J.-J.Marigo. "Revisiting brittle fracture as an energy minimization problem." In: *Journal of the Mechanics and Physics of Solids* 46.8 (1998), pp. 1319–1342.

[29] M. J. Gander and F. Kwok. "Best Robin parameters for optimized Schwarz methods at cross points." In: *SIAM Journal on Scientific Computing* 34.4 (2012), A1849–A1879.

[30] J. Garzon, C. A. Duarte, and J. P. Pereira. "Extraction of stress intensity factors for the simulation of 3-D crack growth with the generalized finite element method." In: *Key Engineering Materials*. Vol. 560. Trans Tech Publ. 2013, pp. 1–36.

[31] R. Geelen, J. Plews, M. Tupek, and J. Dolbow. "An extended/generalized phase-field finite element method for crack growth with global-local enrichment." In: *International Journal for Numerical Methods in Engineering* 121.11 (2020), pp. 2534–2557.

[32] T. Gerasimov, U. Römer, J. Vondřejc, H. G. Matthies, and De L. Lorenzis. "Stochastic phase-field modeling of brittle fracture: computing multiple crack patterns and their probabilities." In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113353.

[33] I .N. Giannakeas, T. K. Papathanasiou, A. S. Fallah, and H. Bahai. "Coupling XFEM and Peridynamics for brittle fracture simulation: part II—adaptive relocation strategy." In: *Computational Mechanics* 66 (2020), pp. 683–705.

[34] I. N. Giannakeas, T. K. Papathanasiou, and H. Bahai. "Wave reflection and cut-off frequencies in coupled FE-peridynamic grids." In: *International Journal for Numerical Methods in Engineering* 120.1 (2019), pp. 29–55.

[35] I. N. Giannakeas, T. K. Papathanasiou, A. S. Fallah, and H. Bahai. "Coupling XFEM and peridynamics for brittle fracture simulation—part I: feasibility and effectiveness." In: *Computational Mechanics* 66 (2020), pp. 103–122.

[36] B. Giovanardi, A. Scotti, and L. Formaggia. "A hybrid XFEM–Phase field (Xfield) method for crack propagation in brittle elastic materials." In: *Computer Methods in Applied Mechanics and Engineering* 320 (2017), pp. 396–420.

[37] P. Gosselet, M. Blanchard, O. Allix, and G. Guguin. "Non-invasive global–local coupling as a Schwarz domain decomposition method: acceleration and generalization." In: *Advanced Modeling and Simulation in Engineering Sciences* 5.1 (2018), p. 4.

[38] A. Gravouil, T. Elguedj, and H. Maigre. "An explicit dynamics extended finite element method. Part 2: Element-by-element stable-explicit/explicit dynamic scheme." In: *Computer Methods in Applied Mechanics and Engineering* 198.30-32 (2009), pp. 2318–2328.

[39] A. Gravouil, N. Moës, and T. Belytschko. "Non-planar 3D crack growth by the extended finite element and level sets—Part II: Level set update." In: *International journal for numerical methods in engineering* 53.11 (2002), pp. 2569–2586.

[40] M. Griebel and G. Zumbusch. "Hash based adaptive parallel multilevel methods with space-filling curves." In: *NIC Symposium*. Vol. 9. 2001, pp. 479–492.

[41] D. Gross and T. Seelig. *Bruchmechanik*. Vol. 2. Springer.

[42] G. V. Guinea, J. Planas, and M. Elices. "KI evaluation by the displacement extrapolation technique." In: *Engineering fracture mechanics* 66.3 (2000), pp. 243–255.

[43] Y. Guo and M. Ruess. "Nitsche's method for a coupling of isogeometric thin shells and blended shell structures." In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 881–905.

[44] P. Gupta, C. A. Duarte, and A. Dhankhar. "Accuracy and robustness of stress intensity factor extraction methods for the generalized/eXtended Finite Element Method." In: *Engineering Fracture Mechanics* 179 (2017), pp. 120–153.

[45] V. Gupta. *Improved conditioning and accuracy of a two-scale generalized finite element method for fracture mechanics*. University of Illinois at Urbana-Champaign, 2014.

[46] V. Gupta and C. A. Duarte. "On the enrichment zone size for optimal convergence rate of the Generalized/Extended Finite Element Method." In: *Computers & Mathematics with Applications* 72.3 (2016), pp. 481–493.

[47] V. Gupta, C. A. Duarte, I. Babuška, and U. Banerjee. "A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics." In: *Computer methods in applied mechanics and engineering* 266 (2013), pp. 23–39.

[48] V. Gupta, C. A. Duarte, I. Babuška, and U. Banerjee. "Stable GFEM (SGFEM): Improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics." In: *Computer methods in applied mechanics and engineering* 289 (2015), pp. 355–386.

[49] V. Gupta, D.-J. Kim, and C. A. Duarte. "Analysis and improvements of global–local enrichments for the generalized finite element method." In: *Computer Methods in Applied Mechanics and Engineering* 245 (2012), pp. 47–62.

[50] L. He, A. J. Valocchi, and C. A. Duarte. "A transient global-local generalized FEM for parabolic and hyperbolic PDEs with multi-space/time scales." In: *Journal of Computational Physics* 488 (2023), p. 112179.

[51] T. Hiroshi, P. C. Paris, and G. R. Irwin. *The stress analysis of cracks handbook*. 2000.

[52] T. Y. Hou and X.-H. Wu. "A multiscale finite element method for elliptic problems in composite materials and porous media." In: *Journal of computational physics* 134.1 (1997), pp. 169–189.

[53] E. V. Iarve, M. R. Gurvich, D. H. Mollenhauer, C. A. Rose, and C. G. Dávila. "Mesh-independent matrix cracking and delamination modeling in laminated composites." In: *International journal for numerical methods in engineering* 88.8 (2011), pp. 749–773.

[54] A. R. Ingraffea and M. Grigoriu. "Probabilistic fracture mechanics: A validation of predictive capability." In: (1990).

[55] G. R. Irwin. "Analysis of stresses and strains near the end of a crack traversing a plate." In: (1957).

[56] J. Nitsche. "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind." In: *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*. Vol. 36. 1. Springer. 1971, pp. 9–15.

[57] J. Jaśkowiec and N. Sukumar. "High-order cubature rules for tetrahedra." In: *International Journal for Numerical Methods in Engineering* 121.11 (2020), pp. 2418–2436.

[58] A. Javili, R. Morasata, E. Oterkus, and S. Oterkus. "Peridynamics review." In: *Mathematics and Mechanics of Solids* 24.11 (2019), pp. 3714–3739.

[59] P. K. Jha and R. Lipton. "Numerical analysis of nonlocal fracture models in holder space." In: *SIAM Journal on Numerical Analysis* 56.2 (2018), pp. 906–941.

[60] X. Jiao. "Face offsetting: A unified approach for explicit moving interfaces." In: *Journal of computational physics* 220.2 (2007), pp. 612–625.

[61] M. F. Kanninen and C. H. Popelar. *Advanced Fracture Mechanics*. 1985.

[62] D.-J. Kim and C. A. Duarte. "A new generalized finite element method for two-scale simulations of propagating cohesive fractures in 3-D." In: *International Journal for Numerical Methods in Engineering* 104.13 (2015), pp. 1139–1172.

[63] D.-J. Kim, C. A. Duarte, and J. P. Pereira. "Analysis of interacting cracks using the generalized finite element method with global-local enrichment functions." In: *Journal of Applied Mechanics* 75.5 (2008).

[64] D.-J. Kim, C. A. Duarte, and S. P. Proença. "A generalized finite element method with global-local enrichment functions for confined plasticity problems." In: *Computational Mechanics* 50 (2012), pp. 563–578.

[65] D.-J. Kim, C. A. Duarte, and S. P. B. Proença. "Generalized finite element method with global-local enrichments for nonlinear fracture analysis." In: *Mechanics of Solids in Brazil* 2009 (2009), pp. 289–314.

[66] D.-J. Kim, C. A. Duarte, and N. A. Sobh. "Parallel simulations of three-dimensional cracks using the generalized finite element method." In: *Computational Mechanics* 47.3 (2011), pp. 265–282.

[67] D.-J. Kim, S.-G. Hong, and C. A. Duarte. "Generalized finite element analysis using the preconditioned conjugate gradient method." In: *Applied Mathematical Modelling* 39.19 (2015), pp. 5837–5848.

[68] D.-J. Kim, J. P. Pereira, and C. A. Duarte. "Analysis of three-dimensional fracture mechanics problems: A two-scale approach using coarse-generalized FEM meshes." In: *International Journal for Numerical Methods in Engineering* 81.3 (2010), pp. 335–365.

[69] I. A. Kunin. *Elastic Media with Microstructure II: Three-Dimensional Models (Springer Series in Solid-State Sciences)*. Softcover reprint of the original 1st ed. 1983. Springer, 2012. ISBN: 9783642819629.

[70] P. Ladevèze, O. Loiseau, and D. Dureisseix. "A micro–macro and parallel computational strategy for highly heterogeneous structures." In: *International Journal for Numerical Methods in Engineering* 52.1-2 (2001), pp. 121–138.

[71] B. Lecampion, A. Bunger, and X. Zhang. "Numerical methods for hydraulic fracture propagation: A review of recent trends." In: *Journal of natural gas science and engineering* 49 (2018), pp. 66–83.

[72] E. H. Lee. "Elastic-plastic deformation at finite strains." In: (1969).

[73] H. Li and C. A. Duarte. "A two-scale generalized finite element method for parallel simulations of spot welds in large structures." In: *Computer Methods in Applied Mechanics and Engineering* 337 (2018), pp. 28–65.

[74] P. Li, W. Li, B. Li, S. Yang, Y. Shen, Q. Wang, and K. Zhou. "A review on phase field models for fracture and fatigue." In: *Engineering Fracture Mechanics* (2023), p. 109419.

[75] R. Lipton. "Dynamic brittle fracture as a horizon limit of peridynamics." In: *Journal of Elasticity* 117.1 (2014), pp. 21–50. DOI: 10.1007/s10659-013-9463-0.

[76] R. Lipton. "Cohesive dynamics and brittle fracture." In: *Journal of Elasticity* 124.2 (2016), pp. 143–191. DOI: 10.1007/s10659-015-9564-z.

[77] R. Lipton and P. K. Jha. "Nonlocal elastodynamics and fracture." In: *Nonlinear Differential Equations and Applications NoDEA* 28.3 (2021), p. 23.

[78] W. K. Liu, S. Jun, and Y. F. Zhang. "Reproducing kernel particle methods." In: *International journal for numerical methods in fluids* 20.8-9 (1995), pp. 1081–1106.

[79] S. Loehnert. "A stabilization technique for the regularization of nearly singular extended finite elements." In: *Computational Mechanics* 54.2 (2014), pp. 523–533.

[80] W. E. Lorensen and H. E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." In: *Seminal graphics: pioneering efforts that shaped the field*. 1998, pp. 347–353.

[81] M. Malekan and F. B. Barros. "Well-conditioning global–local analysis using stable generalized/extended finite element method for linear elastic fracture mechanics." In: *Computational Mechanics* 58 (2016), pp. 819–831.

[82] C. T. McCarthy and M. A. McCarthy. "Design and failure analysis of composite bolted joints for aerospace composites." In: *Polymer Composites in the Aerospace Industry*. Elsevier, 2015, pp. 295–334.

[83] J. M. Melenk and I. Babuška. "The partition of unity finite element method: basic theory and applications." In: *Computer methods in applied mechanics and engineering* 139.1-4 (1996), pp. 289–314.

[84] A. Mesgarnejad, B. Bourdin, and M. Khonsari. "Validation simulations for the variational approach to fracture." In: *Computer Methods in Applied Mechanics and Engineering* 290 (2015), pp. 420–437.

[85] H. Minnebo, J. Majérus, and L. Noels. "Displacement extrapolation method: An alternative to J-integral for stress intensity factors computation using X-FEM." In: *IV European Conference on Computational Mechanics (ECCM2010)*. 2010.

[86] N. Moës, J. Dolbow, and T. Belytschko. "A finite element method for crack growth without remeshing." In: *International journal for numerical methods in engineering* 46.1 (1999), pp. 131–150.

[87] N. Moës, A. Gravouil, and T. Belytschko. "Non-planar 3D crack growth by the extended finite element and level sets—Part I: Mechanical model." In: *International journal for numerical methods in engineering* 53.11 (2002), pp. 2549–2568.

[88] C. D. Mote. "Global-local finite element." In: *International Journal for Numerical Methods in Engineering* 3.4 (1971), pp. 565–574.

[89] M. A. Msekh, J. M. Sargado, M. Jamshidian, P. M. Areias, and T. Rabczuk. "Abaqus implementation of phase-field model for brittle fracture." In: *Computational Materials Science* 96 (2015), pp. 472–484.

[90] A. Muixí, O. Marco, A. Rodríguez-Ferran, and S. Fernández-Méndez. "A combined XFEM phase-field computational model for crack growth without remeshing." In: *Computational Mechanics* 67 (2021), pp. 231–249.

[91] F. M. Mukhtar, P. D. Alves, and C. A. Duarte. "Validation of a 3-D adaptive stable generalized/eXtended finite element method for mixed-mode brittle fracture propagation." In: *International Journal of Fracture* 225.2 (2020), pp. 129–152.

[92] F. M. Mukhtar, N. Shauer, and C. A. Duarte. "Propagation mechanisms and parametric influence in multiple interacting hydraulic fractures: A 3-DG/XFEM hydro-mechanical modeling." In: *International Journal for Numerical and Analytical Methods in Geomechanics* 46.11 (2022), pp. 2033–2059.

[93] T. Ni, M. Zaccariotto, Q.-Z. Zhu, and U. Galvanetto. "Static solution of crack propagation problems in Peridynamics." In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 126–151. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.11.028.

[94] A. K. Noor. "Global-local methodologies and their application to nonlinear analysis." In: *Finite Elements in Analysis and Design* 2.4 (1986), pp. 333–346.

[95] L. Novelli, T. S. de Oliveira, H. A. da Silveira Monteiro, G. M. Fonseca, R. L. da Silva Pitangueira, and F. B. Barros. "Stable Generalized/eXtended Finite Element Method with global–local enrichment for material nonlinear analysis." In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113429.

[96] P. O'Hara, C. A. Duarte, and T. Eason. "Generalized finite element analysis of three-dimensional heat transfer problems exhibiting sharp thermal gradients." In: *Computer Methods in Applied Mechanics and Engineering* 198.21-26 (2009), pp. 1857–1871.

[97] P. O'Hara, C. A. Duarte, and T. Eason. "Transient analysis of sharp thermal gradients using coarse finite element meshes." In: *Computer Methods in Applied Mechanics and Engineering* 200.5-8 (2011), pp. 812–829.

[98] S. Omairey, N. Jayasree, and M. Kazilas. "Injection repair of advanced composites: a prospective method for delamination damage repair." In: *Sustainable Biopolymer Composites*. Elsevier, 2022, pp. 183–211.

[99] E. Orowan. "Fracture and strength of solids." In: *Reports on progress in physics* 12.1 (1949), p. 185.

[100] H. Ouchi, A. Katiyar, J. York, J. T. Foster, and M. M. Sharma. "A fully coupled porous flow and geomechanics model for fluid driven cracks: a peridynamics approach." In: *Computational Mechanics* 55 (2015), pp. 561–576.

[101] P. C. Paris. "A rational analytic theory of fatigue." In: *Trends Engin* 13 (1961), pp. 9–14.

[102] J. Pereira, C. A. Duarte, X. Jiao, and D. Guoy. "Generalized finite element method enrichment functions for curved singularities in 3D fracture mechanics problems." In: *Computational Mechanics* 44 (2009), pp. 73–92. DOI: 10.1007/s00466-008-0356-1.

[103] J. P. Pereira, C. A. Duarte, D. Guoy, and X. Jiao. "hp-Generalized FEM and crack surface representation for non-planar 3-D cracks." In: *International Journal for Numerical Methods in Engineering* 77.5 (2009), pp. 601–633.

[104] J. P. Pereira, C. A. Duarte, and X. Jiao. "Three-dimensional crack growth with hp-generalized finite element and face offsetting methods." In: *Computational Mechanics* 46.3 (2010), pp. 431–453.

[105] S. M. Pizer, K. Siddiqi, G. Székely, J. N. Damon, and S. W. Zucker. "Multiscale medial loci and their properties." In: *International Journal of Computer Vision* 55 (2003), pp. 155–179.

[106] J. A. Plews and C. A. Duarte. "Bridging multiple structural scales with a generalized finite element method." In: *International Journal for Numerical Methods in Engineering* 102.3-4 (2015), pp. 180–201.

[107] J. A. Plews and C. A. Duarte. "A two-scale generalized finite element approach for modeling localized thermoplasticity." In: *International Journal for Numerical Methods in Engineering* 108.10 (2016), pp. 1123–1158.

[108] S. D. Proell, W. A. Wall, and C. Meier. "On phase change and latent heat models in metal additive manufacturing process simulation." In: *Advanced Modeling and Simulation in Engineering Sciences* 7 (2020), pp. 1–32.

[109] H. A. Richard, M. Fulland, and M. Sander. "Theoretical crack path prediction." In: *Fatigue & fracture of engineering materials & structures* 28.1-2 (2005), pp. 3–12.

[110] A. Sadeghirad, D. L. Chopp, X. Ren, E. Fang, and J. Lua. "A novel hybrid approach for level set characterization and tracking of non-planar 3D cracks in the extended finite element method." In: *Engineering Fracture Mechanics* 160 (2016), pp. 1–14. ISSN: 0013-7944. DOI: https://doi.org/10.1016/j.engfracmech.2016.03.027. URL: https://www.sciencedirect.com/science/article/pii/S001379441630131X.

[111] H. Sagan. *Space-filling curves*. Springer Science & Business Media, 2012.

[112] Fraunhofer SCAI. *PUMA – Rapid Enriched Simulation Application Development*. URL: www.scai.fraunhofer.de/de/geschaeftsfelder/meshfree-multiscale-methods/produkte/puma.html (visited on 09/20/2023).

[113] M. Schöllmann, H. A. Richard, G. Kullmer, and M. Fulland. "A new criterion for the prediction of crack development in multiaxially loaded structures." In: *International Journal of Fracture* 117.2 (2002), pp. 129–141.

[114] M. A. Schweitzer. *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*. Vol. 29. Lecture Notes in Computational Science and Engineering. Springer, 2003.

[115] M. A. Schweitzer. "Meshfree and generalized finite element methods (Habilitation thesis)." In: *Universität Bonn* (2008).

[116] M. A. Schweitzer. "An algebraic treatment of essential boundary conditions in the particle–partition of unity method." In: *SIAM Journal on Scientific Computing* 31.2 (2009), pp. 1581–1602.

[117] M. A. Schweitzer. "Stable enrichment and local preconditioning in the particle-partition of unity method." In: *Numerische Mathematik* 118.1 (2011), pp. 137–170.

[118] M. A. Schweitzer. "Variational mass lumping in the partition of unity method." In: *SIAM Journal on Scientific Computing* 35.2 (2013), A1073–A1097.

[119] M. A. Schweitzer and S. Wu. "A moving least squares approach to the construction of discontinuous enrichment functions." In: *Singular Phenomena and Scaling in Mathematical Models*. Springer, 2014, pp. 347–360. DOI: 10.1007/978-3-319-00786-1_15.

[120] M. A. Schweitzer and A. Ziegenhagel. "Rapid Enriched Simulation Application Development with PUMA." In: *Scientific Computing and Algorithms in Industrial Simulations*. Springer, 2017, pp. 207–226. DOI: 10.1007/978-3-319-62458-7_11.

[121] M. Shaat, E. Ghavanloo, and S. A. Fazelzadeh. "Review on nonlocal continuum mechanics: physics, material applicability, and mathematics." In: *Mechanics of Materials* 150 (2020), p. 103587.

[122] N. Shauer and C. A. Duarte. "A generalized finite element method for three-dimensional hydraulic fracture propagation: Comparison with experiments." In: *Engineering Fracture Mechanics* 235 (2020), p. 107098.

[123] N. Shauer and C. A. Duarte. "A three-dimensional Generalized Finite Element Method for simultaneous propagation of multiple hydraulic fractures from a wellbore." In: *Engineering Fracture Mechanics* 265 (2022), p. 108360.

[124] P. Sheikhbahaei, F. Mossaiby, and A. Shojaei. "An efficient peridynamic framework based on the arc-length method for fracture modeling of brittle and quasi-brittle problems with snapping instabilities." In: *Computers & Mathematics with Applications* 136 (2023), pp. 165–190.

[125] H. Si. "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator." In: *ACM Trans. Math. Softw.* 41.2 (2015). ISSN: 0098-3500. DOI: 10.1145/2629697. URL: https://doi.org/10.1145/2629697.

[126] S. A. Silling. "Reformulation of elasticity theory for discontinuities and long-range forces." In: *Journal of the Mechanics and Physics of Solids* 48.1 (2000), pp. 175–209.

[127] S. A. Silling and E. Askari. "A meshfree method based on the peridynamic model of solid mechanics." In: *Computers & structures* 83.17-18 (2005), pp. 1526–1535.

[128] S. A. Silling and R. B. Lehoucq. "Convergence of peridynamics to classical elasticity theory." In: *Journal of Elasticity* 93 (2008), pp. 13–37.

[129] W. S. Slaughter. *The linearized theory of elasticity*. Springer Science & Business Media, 2012.

[130] M. Stolarska, D. L. Chopp, N. Moës, and T. Belytschko. "Modelling crack growth by level sets in the extended finite element method." In: *International journal for numerical methods in Engineering* 51.8 (2001), pp. 943–960.

[131] T. Strouboulis, L. Zhang, and I. Babuška. "Generalized finite element method using mesh-based handbooks: application to problems in domains with many voids." In: *Computer Methods in Applied Mechanics and Engineering* 192.28-30 (2003), pp. 3109–3161.

[132] T. Strouboulis, L. Zhang, and I. Babuška. "p-version of the generalized FEM using mesh-based handbooks with applications to multiscale problems." In: *International Journal for Numerical Methods in Engineering* 60.10 (2004), pp. 1639–1672.

[133] N. Sukumar, D. L. Chopp, E. Béchet, and Moës. "Three-dimensional non-planar crack growth by a coupled extended finite element and fast marching method." In: *International journal for numerical methods in engineering* 76.5 (2008), pp. 727–748.

[134] N. Sukumar, J. E. Dolbow, and N. Moës. "Extended finite element method in computational fracture mechanics: a retrospective examination." In: *International Journal of Fracture* 196 (2015), pp. 189–206.

[135] N. Sukumar, Z. Y. Huang, J.-H. Prévost, and Z. Suo. "Partition of unity enrichment for bimaterial interface cracks." In: *International journal for numerical methods in engineering* 59.8 (2004), pp. 1075–1102.

[136] C. T. Sun and K. M. Mao. "A global-local finite element method suitable for parallel computations." In: *Computers & structures* 29.2 (1988), pp. 309–315.

[137] B. Szabó and I. Babuška. "Finite element analysis(Book)." In: *New York, Wiley-Interscience, 1991, 383* (1991).

[138] E. Tamayo-Mas and A. Rodríguez-Ferran. "A medial-axis-based model for propagating cracks in a regularised bulk." In: *International Journal for Numerical Methods in Engineering* 101.7 (2015), pp. 489–520.

[139] The CGAL Project. *CGAL User and Reference Manual*. 5.5.1. CGAL Editorial Board, 2022. URL: `https://doc.cgal.org/5.5.1/Manual/packages.html`.

[140] Y. Wang and H. Waisman. "Material-dependent crack-tip enrichment functions in XFEM for modeling interfacial cracks in bimaterials." In: *International Journal for Numerical Methods in Engineering* 112.11 (2017), pp. 1495–1518.

[141] J. D. Whitcomb. "Iterative global/local finite element analysis." In: *Computers & structures* 40.4 (1991), pp. 1027–1031.

[142] J.-Y. Wu. "Robust numerical implementation of non-standard phase-field damage models for failure in solids." In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 767–797.

[143] H. Xiao and Z. Gimbutas. "A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions." In: *Computers & mathematics with applications* 59.2 (2010), pp. 663–676.

[144] J. F. Yau, S. S. Wang, and H. T. Corten. "A mixed-mode crack analysis of isotropic solids using conservation laws of elasticity." In: (1980).

[145] L. Yu, X. Wu, N. M. S. Hassan, Y. Wang, W. Ma, and G. Liu. "Modified zipper fracturing in enhanced geothermal system reservoir and heat extraction optimization via orthogonal design." In: *Renewable Energy* 161 (2020), pp. 373–385.

[146] S. Zhang, W. Jiang, and M. R. Tonks. "Assessment of four strain energy decomposition methods for phase field fracture models using quasi-static and dynamic benchmark cases." In: *Materials Theory* 6.1 (2022), p. 6.

[147] X. Zhang. "Impact damage in composite aircraft structures-experimental testing and numerical simulation." In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 212.4 (1998), pp. 245–259.

[148] Y. Zhou, W. Ma, X. Tan, W. Chen, D. Yang, Z. Su, X. Zhang, and F. Xu. "Numerical simulation of fracture propagation in freezing rocks using the extended finite element method (XFEM)." In: *International Journal of Rock Mechanics and Mining Sciences* 148 (2021), p. 104963.

[149] A. Ziegenhagel. "Robust & Efficient Treatment of Industrial-Grade CAD Geometries for a Flat-Top Partition of Unity Method." PhD thesis. Universitäts- und Landesbibliothek Bonn, 2022.