# Enriched finite element methods for fracture-based analysis and design

Zhang, J.

**DOI**
10.4233/uuid:e26d63d4-7cf7-4951-bf55-fcc9c86b0f5d

**Publication date**
2021

**Document Version**
Final published version

**Citation (APA)**
Zhang, J. (2021). *Enriched finite element methods for fracture-based analysis and design*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:e26d63d4-7cf7-4951-bf55-fcc9c86b0f5d

# Enriched finite element methods for fracture-based analysis and design

# Enriched finite element methods for fracture-based analysis and design

## Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof.dr.ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
Monday 24 January 2022 at 15:00 o'clock

by


## Jian ZHANG

Master of Engineering in Engineering Mechanics
Dalian Univeristy of Technology, China
born in Anshan, China

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

| | |
|---|---|
| Rector Magnificus, | chairperson |
| Prof.dr.ir. A. van Keulen, | Delft University of Technology, promotor |
| Dr. A.M. Aragón, | Delft University of Technology, copromotor |
| | |
| Independent members: | |
| Prof.dr.ir. L.J. Sluys | Delft University of Technology |
| Prof.dr. A. Simone | University of Padova, Italy |
| Prof.dr. C.A. Duarte | University of Illinois at Urbana-Champaign, USA |
| Prof.dr. K. Maute | University of Colorado Boulder, USA |
| Dr. D. Dias-da-Costa, | The University of Sydney, Australia |
| Prof.dr.ir. P.G. Steeneken, | Delft University of Technology, reserve member |



The work presented in this dissertation was carried out in the Structural Optimization and Mechanics (SOM) group at Delft University of Technology. This research was funded by China Scholarship Council (CSC).

*Keywords:* Linear elastic fracture mechanics, Enriched finite element methods, Geometric engine, Stress recovery, Topology optimization, Level sets, Topological derivatives

*Printed by:* IPSKAMP printing

*Cover design:* Jian Zhang



ISBN 000-00-0000-000-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

*Dedicated to my beloved parents*
*for their endless love, support and encouragement*

致亲爱的爸爸妈妈
感谢你们给予我无穷无尽的爱，
义无反顾的支持和鼓励

# Contents

# Summary

Cracks, which could nucleate and propagate in engineering structures, could have an adverse effect on mechanical performance and even lead to catastrophic failure. Thus, it is critical to investigate structural behavior under fracture, which requires an appropriate modeling methodology for fracture analysis. Furthermore, designing structures that are resilient to fracture is highly desired, where it is necessary to find a computational design technique for improving the structural fracture resistance. This thesis focuses on realizing these two goals.

As an alternative to experiments, numerical simulation has been widely used to predict structural responses, since it is inexpensive and takes less time. Among numerical techniques for solving fracture mechanics problems, enriched finite element methods (EFEMs) have become increasingly popular over the last two decades. Unlike the standard finite element method (FEM), which requires a mesh fitting the crack geometry, EFEMs fundamentally change the modeling of discontinuous problems via decoupling cracks from the discretization. EFEMs add extra enrichment terms to the standard FEM formulation for capturing the discontinuous solution and/or gradient fields. The Discontinuity-Enriched Finite Element Method (DE-FEM) was developed to handle both weak and strong discontinuities via a unified formulation. As the method was shown to behave well in solving 2-D fracture mechanics problems, here we extend it to 3-D problems including cracks and/or material interfaces. Noteworthy, we demonstrate that DE-FEM is intrinsically endowed with stability, where the condition number of the stiffness matrix grows with mesh size $h$ as $\mathcal{O}\left(h^{-2}\right)$. Although EFEMs provide great flexibility in solving discontinuous problems, the complexity of generating the fitted discretization in standard FEM is transferred to the construction of the enriched formulation, which requires an interaction between a background mesh (usually structured) and discontinuities. Thus, an efficient and general geometric engine is proposed to perform advanced computational geometry operations that are necessary for a robust implementation.

For the computational design of structures with tailored fracture resistance, it would be desired to consider all potential cracks with arbitrary orientations during an optimization procedure, where DE-FEM could provide an elegant analysis tool for predicting the structural response. However, too many finite element analyses would be required, resulting in a computationally intractable approach. Moreover, custom-made meshes and specific enrichment functions could be required to capture singularities at crack tips. In order to circumvent these difficulties, topological derivatives based on the stress field from a single finite element analysis of the uncracked body are used to evaluate energy release rates. For simplicity, cracks are assumed to nucleate only perpendicularly to the structural boundary at the locations of enriched nodes within the discretized model. A novel topology optimization procedure is then proposed to design the material layout under linear elastic frac-

ture mechanics (LEFM) assumptions for brittle materials. A level set approach that can provide a crisp and clear boundary description is used to represent the structural topology, and the structural analysis is performed with EFEM for resolving the boundary (not the cracks). As this procedure highly depends on the stress field, peak stresses, which could arise in elements with bad aspect ratios and/or relatively tiny areas, could degenerate the optimized design. Therefore, a recovery technique is proposed to improve the stress approximation, where peak stresses can be eliminated completely. By introducing this technique to topology optimization, the recovered stress field leads to a more accurate evaluation of objective function at the expense of a more involved sensitivity analysis formulation. Results show that the proposed topology optimization methodology cannot only improve the structural fracture resistance, but can also introduce fracture anisotropy into the final design if desired.

# Samenvatting

Scheuren die ontstaan en zich uitbreiden in technische constructies kunnen een negatief effect hebben op de mechanische prestaties van de constructie en zelfs leiden tot het catastrofaal falen van de constructie. Daarom is het van cruciaal belang om het gedrag van een constructie te onderzoeken wanneer er scheuren zijn ontstaan. Dit vereist een geschikte modelleringsmethodologie voor de breukanalyse. Daarnaast zijn constructies die bestand zijn tegen scheuren zeer gewenst en is het noodzakelijk om een numerieke ontwerptechniek te vinden om de breukweerstand van constructies te verbeteren. Dit proefschrift richt zich op het realiseren van deze twee doelen.

Als alternatief voor experimenteel onderzoek wordt numerieke simulatie op grote schaal toegepast om de respons van een constructie te voorspellen, omdat het goedkoop en tijdbesparend is. Onder numerieke technieken voor het oplossen van breukmechanische-problemen zijn de verrijkte eindige-elementenmethoden (EFEMs) in de afgelopen twee decennia steeds populairder geworden. In tegenstelling tot de klassieke eindige-elementenmethode (FEM), waar een mesh voor nodig is die past bij de scheurgeometrie, verandert de EFEM fundamenteel de modellering van discontinue problemen door scheuren te ontkoppelen van de discretisatie. De EFEMs voegen extra termen toe aan de klassieke FEM-formulering voor het vastleggen van de discontinue oplossing en/of gradiëntvelden. De Discontinuity-Enriched Finite Element Method (DE-FEM) is ontwikkeld om met zowel zwakke als sterke discontinuïteiten om te gaan via een geünificeerde formulering. Omdat het is aangetoond dat de methode goed werkt bij het oplossen van 2-D breukmechanische problemen, breiden we deze in dit proefschrift uit tot 3-D problemen met scheuren en/of materiaalinterfaces. We tonen aan dat de DE-FEM zeer stabiel is, waarbij het conditienummer van de stijfheidsmatrix groeit met de grootte van de mesh. Hoewel de EFEMs grote flexibiliteit biedt bij het oplossen van discontinue problemen, wordt de complexiteit van het genereren van een gepaste discretisatie in de klassieke FEM overgebracht naar de constructie van de verrijkte formulering. Dit vereist een interactie tussen een achtergrondmesh (meestal gestructureerd) en discontinuïteiten. Om deze reden wordt een efficiënte en algemene geometrische engine voorgesteld om geavanceerde numerieke geometriebewerkingen uit te voeren die nodig zijn voor een robuuste implementatie.

Voor het numerieke ontwerpproces van constructies met een op het ontwerp aangepaste breukweerstand zou het wenselijk zijn om alle mogelijke scheuren met willekeurige oriëntaties in overweging te nemen tijdens een optimalisatieprocedure. Hiervoor zou de DE-FEM een elegant analyse-instrument kunnen zijn voor het voorspellen van de respons van de constructie. Om dit te doen zouden echter te veel eindige-elementenanalyses nodig zijn, wat zou resulteren in een wat rekenprocedure betreft onhandelbare aanpak. Bovendien kunnen op maat gemaakte

meshes en specifieke verrijkingsfuncties nodig zijn om singulariteiten bij scheurtippen vast te leggen. Om deze problemen te omzeilen worden topologische afgeleiden op basis van het spanningsveld van een enkele eindige-elementenanalyse van het lichaam zonder scheuren gebruikt om de energieafgiftesnelheden te evalueren. Voor de eenvoud wordt aangenomen dat scheuren alleen loodrecht op structuurgrenzen ontstaan op de locaties van verrijkte knooppunten binnen het gediscretiseerde model. Vervolgens wordt een nieuwe topologie-optimalisatieprocedure voorgesteld om de materiaalopbouw te ontwerpen onder aannames van lineaire elastische breukmechanica voor brosse materialen. Om de structurele topologie weer te geven, wordt een level-set aanpak ingesteld die een scherpe en duidelijke grensbeschrijving kan bieden. De analyse van de constructie wordt uitgevoerd met EFEM voor het oplossen van de grens (niet de scheuren). Aangezien deze procedure sterk afhankelijk is van het spanningsveld, kunnen piekspanningen, die kunnen optreden in elementen met slechte geometrieverhoudingen en/of kleine oppervlakken, het geoptimaliseerde ontwerp verslechteren. Daarom wordt een reconstructietechniek voorgesteld om de spanningsbenadering te verbeteren, waarbij piekspanningen volledig kunnen worden geëlimineerd. Door deze techniek te introduceren in topologie-optimalisatie, leidt het gereconstrueerde spanningsveld tot een nauwkeurigere evaluatie van de objectieve functie ten koste van een complexere formulering van de gevoeligheidsanalyse. De resultaten tonen aan dat de voorgestelde topologie-optimalisatiemethodologie niet alleen de breukweerstand van de constructie kan verbeteren, maar indien gewenst ook anisotropie in het scheurgedrag in het uiteindelijke ontwerp kan introduceren.

# 1

# Introduction

*In this chapter, we first introduce the motivation of this PhD project, where we give a brief introduction to fracture mechanics, enriched finite element methods for modeling discontinuities, and the structural optimization. Then, we list the outline as a guideline for readers.*

1

## 1.1. Motivation

Micro-cracks could materialize during manufacturing and/or service life of a structure, then grow and coalesce into major cracks that could deteriorate structural performance and even result in catastrophic failures [1, 2]. For example, the Quebec bridge collapsed in 1907 because of the fracture failure of the anchoring supporting element [3]. A total of 2700 liberty ships were built in USA for transporting the supplies needed to the war front during World War II; however, approximately 400 vessels suffered catastrophic brittle fracture [4]. On July 6, 1996, a Delta Airlines aircraft (Flight 1288) experienced uncontained, catastrophic turbine engine failure caused by the left engine's compressor fan hub fracture [5]. These and other incidents have raised awareness behind failure due to fracture and contributed to improve our understanding and development of fracture mechanics.

The science of fracture mechanics developed and attained maturity gradually during the 20th century. In the 1920s, Alan Arnold Griffith, who is regarded as "the founding father of modern fracture theory", put forward the theory behind brittle fracture that led to the introduction of linear elastic fracture mechanics (LEFM) [6]. As glass was chosen as the model material, his theory was not suitable for certain engineering materials such as metals. In order to generalize Griffith's theory, Irwin considered the energy dissipated by local plastic flow for the fracture of ductile materials [7]. Later, Irwin developed the energy release rate (ERR) concept based on Griffith's model, which is more convenient for solving engineering problems [8], and proposed the stress intensity factor (SIF) to express the stress state around a crack tip [9]. Since LEFM assumes small-scale yielding at a crack tip, it is quite restrictive for materials that undergo significant plastic deformation. Therefore, elastoplastic fracture mechanics arose from studying the fracture properties of these materials, where the size of plastic zone near the crack tip is comparable to the crack length [10]. The crack tip opening displacement (CTOD), as an important characteristic parameter, was proposed for the determination of fracture toughness [11]. Later, the path-independent $J$-integral was proposed as another fracture parameter to characterize the stress and strain fields at crack tips in nonlinear materials [12].

While researchers studied the principles of fracture mechanics, others turned to improve structural fracture resistance by relying on materials with high fracture toughness. A lot of research has thus been conducted for designing materials with enhanced fracture toughness [13–18], where a popular approach is to get inspiration from nature [19]. For instance, bone [20] and nacre [21] are renowned for their high fracture toughness, which can exceed that of their individual components by orders of magnitude. By studying their toughening mechanisms [20, 22], it was found that the complex hierarchical structures that span multiple length scales and heterogeneous compositions play a fundamental role. Thus, several artificial materials that mimic these characteristics were proposed [13, 23–25]. However, due to the numerous levels of their hierarchical structure, experimental tests and measurements are difficult to realize. Moreover, their counterintuitive nature and complex structure even render ineffective state-of-the-art computational tools for their analysis.

Instead of using materials with high fracture toughness, an alternative is to simply adjust the structural layout to mitigate fracture. While the engineer can use the trial-and-error empirical design approach to avoid structural failure, computational tools have come a long way in obtaining designs that mitigate fracture or other mechanisms that compromise structural integrity. Instead of performing experiments, computational design provides a viable alternative to study structural fracture behavior, and also cut cost, save time, and provide guidance for experimental validation.

In general, computational design is composed of two components: structural analysis and an optimization strategy for improving the design. For the former, finite element methods (FEMs) have been widely used to predict structural behavior, evaluate stress intensity factors (SIFs) and energy release rates (ERRs) [26–28]. The simulation of discontinuous models (for instance, structures with cracks and/or material interfaces) could be performed by means of the standard FEM with fitted or geometry-conforming meshes, ensuring that the sides of finite elements align to discontinuities [29]. However, the generation of matching discretizations can be quite demanding [30], especially for 3-D problems. In addition, more effort should be put into controlling the mesh quality as elements with bad aspect ratios could deteriorate the approximation accuracy [31]. Moreover, for modeling evolving discontinuities with intricate configurations, it is necessary to update the mesh to match discontinuities at each step throughout the analysis, which could drastically increase computational cost and thus must be avoided.

Motivated by these obstacles, enriched FEMs have been proposed to decouple the geometric representation of discontinuities from finite element meshes. The eXtended/Generalized Finite Element Method (X/GFEM), which is based on the partition of unity method [32], handles discontinuous models with a background mesh (usually structured) that does not conform to discontinuities [33–35]. Enriched degrees of freedom (DOFs) are added to background mesh nodes, and their corresponding enrichment functions are constructed to capture the discontinuous primal and/or gradient fields. Although X/GFEM provides great flexibility in modeling discontinuities, it also faces several challenges, such as the need for special formulations for enforcing non-zero essential (Dirichlet) boundary conditions [36], and keeping accuracy in "blending elements" (non-cut elements sharing enriched DOFs) by choosing appropriate enrichment functions [37]. Since several enrichment functions could result in ill-conditioned system matrices, Stable GFEM (SGFEM) has been proposed to recover stability [38, 39].

An enriched FEM is always desired for both retaining the mesh-geometry decoupling and avoiding the aforementioned issues. The Interface-enriched Generalized Finite Element Method (IGFEM) provides an alternative way that is devised for problems with weak discontinuities [40, 41], such as multi-phase materials with a discontinuous gradient field at phase interfaces. Contrary to X/GFEM, this methodology creates enriched nodes at the intersections between material interfaces and element edges. With IGFEM, it is straightforward to prescribe non-homogeneous essential boundary conditions on discontinuities [42]. Moreover, as enrichment functions are constructed with Lagrange shape functions of cut subdomains (or

integration elements), there are no blending elements existing. More importantly, IGFEM is intrinsically stable with regard to the condition number of stiffness matrices [43]. IGFEM has been successfully explored in many applications, such as modeling fiber-reinforced composites and actively-cooled microvascular materials [41], studying the multiscale damage evolution of heterogeneous adhesives [44], and solving problems with curved material interfaces [45, 46]. Furthermore, IGFEM has been introduced to gradient-based shape [47, 48] and topology [49, 50] optimization. The Hierarchical Interface-enriched Finite Element Method (HIFEM) was also proposed as the successor of IGFEM to handle multiple material interfaces within a single element by means of hierarchical enrichments [51, 52].

As IGFEM only focuses on solving problems with material interfaces, the Discontinuity Enriched Finite Element Method (DE-FEM) was later developed to deal with both weak and strong discontinuities by means of a unified formulation, where discontinuous enrichment functions are also proposed to resolve the displacement jumps along cracks [53]. DE-FEM was shown to perform well to solve the fracture mechanics problems in 2-D. Therefore, the first challenge is to "**Extend DE-FEM to 3-D**" because an accurate finite element analysis of fracture is necessary for designing structures that are resilient for fracture. However, since it is more complicated to take care of the interactions between the background mesh and discontinuities in 3-D, the second challenge lies on "**Robust and efficient computational geometry routines**".

The second component of computational design is structural optimization, which is typically classified into three categories: size, shape and topology optimization [54]. Size optimization focuses on geometrical parameters of the components within a structure [55], such as cross-sectional areas of the elements in a truss structure. In shape optimization the design variables are parameters used to describe the geometric layout [56], such as the coordinates of the surface nodes. Topology optimization (TO) finds an optimal distribution of material within a specified design domain, without making any *a priori* assumptions about the geometric configuration of the final design [54]. As the design space is less restricted, the extra freedom in design usually gives topology optimization the edge over size and shape optimization. Therefore, topology optimization that is considered as a powerful design tool has been widely used in many contexts, including additive manufacturing [57–59], compliant mechanisms [60–62], heat transfer [63, 64], acoustic-structure interaction [65, 66], and photonic and phononic crystals [67–69]. More importantly, topology optimization has also been explored to improve structural fracture resistance [70–77].

Most of works incorporating fracture criteria into topology optimization predefine the location of cracks in the computational domain at the beginning. The optimization algorithm is then used to update the structural topology for mitigating the crack's effect [70, 71, 75]. Rather than optimizing the topology of the structure with cracks, the alternative is to design an adhesively bonded patch, which is fixed to the structure for improving the structural fracture resistance [72]. However, cracks nucleate at locations with high energy release rates, which could evolve during the optimization. Therefore, a better design approach would assume that cracks could

appear anywhere in the design domain. Nevertheless, it is also not needed to consider the entire domain since cracks most likely will nucleate at material interfaces or at the boundaries. However, significant finite element analyses are required to model all potential cracks, which results in the third task about "**Designing structures with tailored fracture resistance via using minimal computational resource**". Since the evaluation of characteristic fracture parameters, such as SIFs and ERRs, is highly dependent on the stress field, it is critical to obtain an accurate stress approximation. However, stress overestimation could arise while using enriched FEMs, in integration elements with bad aspect ratios and/or relatively tiny areas, which could mislead the optimizer at best and hinder finding an optimized solution at worse. Therefore, this leads to the last target for "**Improving the accuracy of the stress approximation**". Addressing these challenges forms the core of this thesis.

## 1.2. Outline

This dissertation, which pursues a fundamental advancement in both numerical modeling and topology optimization, is composed of four self-contained articles. Chapter 2 describes a novel enriched finite element methodology for solving 3-D fracture mechanics problems built upon IGFEM, named the Discontinuity Enriched Finite Element Method (DE-FEM). The chapter gives a detailed explanation on DE-FEM's formulation, which can solve problems with both weak and strong discontinuities. Moreover, the evaluation of enrichment functions for capturing the discontinuous displacement field is illustrated thoroughly, and an ordered tree data structure for hierarchical bookkeeping is described in detail. While extending DE-FEM to handle 3-D discontinuous models, one of the main challenges lies in performing the computational geometry operations to create the enriched model (enriched nodes and integration elements).

Chapter 3 introduces an efficient and robust object-oriented geometric engine especially designed for unfitted/immersed/enriched methods. A detailed description of the geometric engine is given together with pseudo-code, focusing on finding intersections between element edges and discontinuities, creating integration elements, assigning elements to physical groups, and modifying the background mesh data structure. Both implicit and explicit methods are introduced to represent the geometric configuration of discontinuities. For the former, a level set approach is applied to describe the morphology of material interfaces. The latter adopts geometric entities, such as line segments, triangles and polygons, or a combination thereof in the form of a lower-dimensional finite element mesh. In order to ensure robustness, the use of tolerance-based predicates is mitigated wherever possible. Regarding efficiency, a $k$-d tree and a dual graph data structure are used to reduce the computational complexity of the geometric operations; an ordered tree data structure is used to store cut and children elements and to efficiently iterate over elements for the assembly of local arrays. In addition, the new discretizations generated from the geometric engine are used with IGFEM/DE-FEM for solving problems with material interfaces/cracks.

Chapter 4 deals with the challenge on improving the accuracy of the stress approximation; the chapter introduces a stress improvement procedure (SIP) to recover the discontinuous stress fields obtained from IGFEM. As material interfaces split the background mesh element into subcells to which different material properties are assigned, the calculation domain, which is used to improve the stress distribution adjoining to the interface, is carefully constructed by only considering a patch of (integration) elements with the same material properties. It is shown that the recovered stresses are more accurate than the directly calculated stresses. Moreover, IGFEM with the recovery technique provides more accurate approximations than directly calculated stresses in standard FEM on matching meshes. More importantly, this stress recovery procedure is fundamental for the following chapter since an accurate stress field is required for the evaluation of energy release rate.

Chapter 5 proposes a novel topology optimization procedure to design structures with tailored fracture resistance, where a level set function is used to represent the structural topology and IGFEM that resolves the boundary is adopted to obtain the structural response. Regarding practicality and computational costs, a formulation derived from topological derivatives is used to evaluate the energy release rates of all potential cracks, where only a single finite element analysis of the uncracked domain is required. This procedure significantly reduces the computational cost needed throughout the optimization. An aggregate function is used to group energy release rates into a single objective, and the corresponding sensitivity associated with design variables is derived analytically by using an adjoint formulation. Results show the proposed approach performs well in minimizing the maximum energy release rate and in introducing fracture resistance anisotropy to the final design.

Finally, the conclusions of this dissertation are presented in Chapter 6. Recommendations for future work are also provided at the end of this dissertation.

## References


[1] E. Ossa, *Failure analysis of a civil aircraft landing gear,* Engineering Failure Analysis **13**, 1177 (2006).

[2] M. V. Biezma and F. Schanack, *Collapse of steel bridges,* Journal of Performance of Constructed Facilities **21**, 398 (2007).

[3] C. Pearson and N. Delatte, *Collapse of the quebec bridge, 1907,* Journal of performance of constructed facilities **20**, 84 (2006).

[4] T. L. Anderson, *Fracture mechanics: fundamentals and applications* (CRC press, 2017).

[5] J. Hall, *Aircraft accident report: Uncontained engine failure delta air lines flight 1288 mcdonnell douglas md-88,* Washington, DC **20594** (1996).

[6] A. A. Griffith, *The phenomena of rupture and flow in solids,* Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character **221**, 163 (1921).

[7] G. R. Irwin, *Fracture dynamics,* Fracturing of metals (1947).

[8] G. R. Irwin, *Onset of fast crack propagation in high strength steel and aluminum alloys*, Tech. Rep. (NAVAL RESEARCH LAB WASHINGTON DC, 1956).

[9] G. R. Irwin, *Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate,* Journal of Applied Mechanics **24**, 361 (2021).

[10] N. Perez, *Elastic-plastic fracture mechanics,* in *Fracture Mechanics* (Springer, 2017) pp. 257–287.

[11] A. Wells, *Proceedings of the crack propagation symposium,* Cranfield, England **1**, 210 (1961).

[12] J. R. Rice *et al.*, *A path independent integral and the approximate analysis of strain concentration by notches and cracks,* (ASME, 1968).

[13] E. Munch, M. E. Launey, D. H. Alsem, E. Saiz, A. P. Tomsia, and R. O. Ritchie, *Tough, bio-inspired hybrid materials,* Science **322**, 1516 (2008).

[14] O. Kolednik, J. Predan, F. D. Fischer, and P. Fratzl, *Bioinspired design criteria for damage-resistant materials with periodically varying microstructure,* Advanced Functional Materials **21**, 3634 (2011).

[15] L. S. Dimas, G. H. Bratzel, I. Eylon, and M. J. Buehler, *Tough composites inspired by mineralized natural materials: computation, 3d printing, and testing,* Advanced Functional Materials **23**, 4629 (2013).

[16] J. Knott, *11 - design and material issues in improving fracture/fatigue resistance and structural integrity in power plants,* in *Structural Alloys for Power Plants*, Woodhead Publishing Series in Energy, edited by A. Shirzadi and S. Jackson (Woodhead Publishing, 2014) pp. 319–354.

[17] O. Kolednik, J. Predan, F. Fischer, and P. Fratzl, *Improvements of strength and fracture resistance by spatial material property variations,* Acta Materialia **68**, 279 (2014).

[18] M. Mirkhalaf, A. K. Dastjerdi, and F. Barthelat, *Overcoming the brittleness of glass through bio-inspiration and micro-architecture,* Nature communications **5**, 1 (2014).

[19] B. Bhushan, *Biomimetics: lessons from nature–an overview,* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **367**, 1445 (2009).

[20] M. E. Launey, M. J. Buehler, and R. O. Ritchie, *On the mechanistic origins of toughness in bone,* Annual review of materials research **40**, 25 (2010).

[21] K. Bertoldi, D. Bigoni, and W. Drugan, *Nacre: an orthotropic and bimodular elastic material,* Composites Science and Technology **68**, 1363 (2008).

[22] F. Barthelat, H. Tang, P. Zavattieri, C.-M. Li, and H. Espinosa, *On the mechanics of mother-of-pearl: a key feature in the material hierarchical structure,* Journal of the Mechanics and Physics of Solids **55**, 306 (2007).

[23] G. M. Luz and J. F. Mano, *Biomimetic design of materials and biomaterials inspired by the structure of nacre,* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **367**, 1587 (2009).

[24] F. Bouville, E. Maire, S. Meille, B. Van de Moortèle, A. J. Stevenson, and S. Deville, *Strong, tough and stiff bioinspired ceramics from brittle constituents,* Nature materials **13**, 508 (2014).

[25] U. G. Wegst, H. Bai, E. Saiz, A. P. Tomsia, and R. O. Ritchie, *Bioinspired structural materials,* Nature materials **14**, 23 (2015).

[26] D. M. Tracey, *Finite elements for determination of crack tip elastic stress intensity factors,* Engineering Fracture Mechanics **3**, 255 (1971).

[27] N. Sukumar, N. Moës, B. Moran, and T. Belytschko, *Extended finite element method for three-dimensional crack modelling,* International journal for numerical methods in engineering **48**, 1549 (2000).

[28] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydlo, *A generalized finite element method for the simulation of three-dimensional dynamic crack propagation,* Computer Methods in Applied Mechanics and Engineering **190**, 2227 (2001).

[29] J. Wood, *Finite element analysis of composite structures,* Composite Structures **29**, 219 (1994).

[30] J. R. Shewchuk, *What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures*, Tech. Rep. (In Proc. of the 11th International Meshing Roundtable, 2002).

[31] Q. Du and D. Wang, *Recent progress in robust and quality delaunay mesh generation,* Journal of Computational and Applied Mathematics **195**, 8 (2006).

[32] I. Babuška and J. M. Melenk, *The partition of unity method,* International journal for numerical methods in engineering **40**, 727 (1997).

[33] J. T. Oden, C. A. Duarte, and O. C. Zienkiewicz, *A new cloud-based hp finite element method,* Computer Methods in Applied Mechanics and Engineering **153**, 117 (1998).

[34] N. Moës, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing,* International Journal for Numerical Methods in Engineering **46**, 131 (1999).

[35] T. P. Fries and T. Belytschko, *The extended/generalized finite element method: An overview of the method and its applications,* International Journal for Numerical Methods in Engineering **84**, 253.

[36] N. Moës, E. Béchet, and M. Tourbier, *Imposing Dirichlet boundary conditions in the extended finite element method,* International Journal for Numerical Methods in Engineering **67**, 1641 (2006).

[37] T. P. Fries, *A corrected XFEM approximation without problems in blending elements,* International Journal for Numerical Methods in Engineering **75**, 503 (2008).

[38] V. Gupta, C. Duarte, I. Babuška, and U. Banerjee, *Stable GFEM (SGFEM): Improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **289**, 355 (2015).

[39] K. Kergrene, I. Babuška, and U. Banerjee, *Stable generalized finite element method and associated iterative schemes; application to interface problems,* Computer Methods in Applied Mechanics and Engineering **305**, 1 (2016).

[40] S. Soghrati, A. M. Aragón, C. A. Duarte, and P. H. Geubelle, *An interface-enriched generalized fem for problems with discontinuous gradient fields,* International Journal for Numerical Methods in Engineering **89**, 991 (2012).

[41] S. Soghrati and P. H. Geubelle, *A 3d interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries,* Computer Methods in Applied Mechanics and Engineering **217**, 46 (2012).

[42] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions,* International Journal for Numerical Methods in Engineering **120**, 1163 (2019).

[43] A. M. Aragón, B. Liang, H. Ahmadian, and S. Soghrati, *On the stability and interpolating properties of the hierarchical interface-enriched finite element method,* Computer Methods in Applied Mechanics and Engineering **362**, 112671 (2020).

[44] A. M. Aragón, S. Soghrati, and P. H. Geubelle, *Effect of in-plane deformation on the cohesive failure of heterogeneous adhesives,* Journal of the Mechanics and Physics of Solids **61**, 1600 (2013).

[45] S. Soghrati, C. A. Duarte, and P. H. Geubelle, *An adaptive interface-enriched generalized fem for the treatment of problems with curved interfaces,* International Journal for Numerical Methods in Engineering **102**, 1352 (2015).

[46] M. H. Tan, M. Safdari, A. R. Najafi, and P. H. Geubelle, *A nurbs-based interface-enriched generalized finite element scheme for the thermal analysis and design of microvascular composites,* Computer Methods in Applied Mechanics and Engineering **283**, 1382 (2015).

[47] A. R. Najafi, M. Safdari, D. A. Tortorelli, and P. H. Geubelle, *A gradient-based shape optimization scheme using an interface-enriched generalized fem,* Computer Methods in Applied Mechanics and Engineering **296**, 1 (2015).

[48] M. H. Y. Tan and P. H. Geubelle, *3d dimensionally reduced modeling and gradient-based optimization of microchannel cooling networks,* Computer Methods in Applied Mechanics and Engineering **323**, 230 (2017).

[49] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *An interface-enriched generalized finite element method for level set-based topology optimization,* Structural and Multidisciplinary Optimization **63**, 1 (2021).

[50] J. Zhang, F. van Keulen, and A. M. Aragón, *On tailoring fracture resistance of brittle structures: A level set interface-enriched topology optimization approach,* Computer Methods in Applied Mechanics and Engineering (2021).

[51] S. Soghrati, *Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations,* Journal of Computational Physics **275**, 41 (2014).

[52] S. Soghrati and H. Ahmadian, *3d hierarchical interface-enriched finite element method: Implementation and applications,* Journal of Computational Physics **299**, 45 (2015).

[53] A. M. Aragón and A. Simone, *The discontinuity-enriched finite element method,* International Journal for Numerical Methods in Engineering **112**, 1589 (2017).

[54] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications* (Springer Science & Business Media, 2013).

[55] J. Schutte and A. Groenwold, *Sizing design of truss structures using particle swarms,* Structural and Multidisciplinary Optimization **25**, 261 (2003).

[56] T. Lindby and J. Santos, *Shape optimization of three-dimensional shell structures with the shape parametrization of a cad system,* Structural optimization **18**, 126 (1999).

[57] M. Langelaar, *Topology optimization of 3d self-supporting structures for additive manufacturing,* Additive Manufacturing **12**, 60 (2016).

[58] M. Langelaar, *An additive manufacturing filter for topology optimization of print-ready designs,* Structural and multidisciplinary optimization **55**, 871 (2017).

[59] X. Guo, J. Zhou, W. Zhang, Z. Du, C. Liu, and Y. Liu, *Self-supporting structure design in additive manufacturing through explicit topology optimization,* Computer Methods in Applied Mechanics and Engineering **323**, 27 (2017).

[60] O. Sigmund, *On the design of compliant mechanisms using topology optimization,* Journal of Structural Mechanics **25**, 493 (1997).

[61] Z. Luo, L. Tong, M. Y. Wang, and S. Wang, *Shape and topology optimization of compliant mechanisms using a parameterization level set method,* Journal of Computational Physics **227**, 680 (2007).

[62] Q. Xia and T. Shi, *Topology optimization of compliant mechanism and its support through a level set method,* Computer Methods in Applied Mechanics and Engineering **305**, 359 (2016).

[63] M. Zhou, J. Alexandersen, O. Sigmund, and C. B. Pedersen, *Industrial application of topology optimization for combined conductive and convective heat transfer problems,* Structural and Multidisciplinary Optimization **54**, 1045 (2016).

[64] T. Dbouk, *A review about the engineering design of optimal heat transfer systems using topology optimization,* Applied Thermal Engineering **112**, 841 (2017).

[65] G. H. Yoon, J. S. Jensen, and O. Sigmund, *Topology optimization of acoustic–structure interaction problems using a mixed finite element formulation,* International journal for numerical methods in engineering **70**, 1049 (2007).

[66] C. B. Dilgen, S. B. Dilgen, N. Aage, and J. S. Jensen, *Topology optimization of acoustic mechanical interaction problems: a comparative review,* Structural and Multidisciplinary Optimization **60**, 779 (2019).

[67] H. Men, K. Y. Lee, R. M. Freund, J. Peraire, and S. G. Johnson, *Robust topology optimization of three-dimensional photonic-crystal band-gap structures,* Optics express **22**, 22632 (2014).

[68] G. Yi and B. D. Youn, *A comprehensive survey on topology optimization of phononic crystals,* Structural and Multidisciplinary Optimization **54**, 1315 (2016).

[69] W. Li, F. Meng, Y. Chen, Y. f. Li, and X. Huang, *Topology optimization of photonic and phononic crystals and metamaterials: a review,* Advanced Theory and Simulations **2**, 1900017 (2019).

[70] G. X. Gu, L. Dimas, Z. Qin, and M. J. Buehler, *Optimization of composite fracture properties: method, validation, and applications,* Journal of Applied Mechanics **83**, 071006 (2016).

[71] Z. Kang, P. Liu, and M. Li, *Topology optimization considering fracture mechanics behaviors at specified locations,* Structural and Multidisciplinary Optimization **55**, 1847 (2017).

[72] A. Klarbring, B. Torstenfelt, U. Edlund, P. Schmidt, K. Simonsson, and H. Ansell, *Minimizing crack energy release rate by topology optimization,* Structural and Multidisciplinary Optimization **58**, 1695 (2018).

[73] J. B. Russ and H. Waisman, *Topology optimization for brittle fracture resistance,* Computer Methods in Applied Mechanics and Engineering **347**, 238 (2019).

[74] J. B. Russ and H. Waisman, *A novel topology optimization formulation for enhancing fracture resistance with a single quasi-brittle material,* International Journal for Numerical Methods in Engineering **121**, 2827 (2020).

[75] J. Hu, S. Yao, N. Gan, Y. Xiong, and X. Chen, *Fracture strength topology optimization of structural specific position using a bi-directional evolutionary structural optimization method,* Engineering Optimization **52**, 583 (2020).

[76] D. Da and J. Yvonnet, *Topology optimization for maximizing the fracture resistance of periodic quasi-brittle composites structures,* Materials **13**, 3279 (2020).

[77] C. Wu, J. Fang, S. Zhou, Z. Zhang, G. Sun, G. P. Steven, and Q. Li, *A path-dependent level set topology optimization with fracture criterion,* Computers & Structures **249**, 106515 (2021).

# 2

# A Stable Discontinuity-Enriched Finite Element Method for 3-D Problems containing Weak and Strong Discontinuities

*An enriched finite element technique, named the Discontinuity-Enriched Finite Element Method (DE-FEM), was introduced for solving problems with both weak and strong discontinuities in 2-D. In this mesh-independent procedure, enriched degrees of freedom are added to new nodes collocated at the intersections between discontinuities and the sides of finite elements of the background mesh. In this chapter we extend DE-FEM to 3-D and describe in detail the implementation of a geometric engine capable of handling interactions between discontinuities and the background mesh. Several numerical examples in linear elastic fracture mechanics demonstrate the capability and performance of DE-FEM in handling discontinuities in a fully mesh-independent manner. We compare convergence properties and the ability to extract stress intensity factors with standard FEM. Most importantly, we show DE-FEM provides a stable formulation with regards to the condition number of the resulting system stiffness matrix.*

This chapter has been published in *Computer Methods in Applied Mechanics and Engineering*, 355, 1097-1123 (2019).

## 2.1. Introduction

Enriched finite element methods have fundamentally changed the modeling of problems containing discontinuities. By means of enrichment functions that incorporate the required field jumps, these methods are able to completely decouple the geometrical description of discontinuities from the underlying finite element (FE) discretization, eliminating the need for creating *matching* or *discontinuity-conforming* meshes. The Discontinuity-Enriched Finite Element Method (DE-FEM), proposed by Aragón and Simone for 2-D problems [1], uses a single formulation to model problems containing both weak and strong discontinuities (the $\mathcal{C}^{-1}-$continuous field) by placing enriched degrees of freedom (DOFs) *only* to nodes created along discontinuities. This new versatile procedure, which has several advantages over other commonly used enriched formulations, is demonstrated in this manuscript for 3-D problems in elastostatics.

Standard FEM can be used to model problems with discontinuities, but it requires a matching discretization where the sides of elements align to material interfaces or cracks. When dealing with fracture problems, mesh adaptivity in the neighborhood of crack tips is usually needed to improve the FE approximation properties at the expense of a more complex mesh creation. Alternatively, the stress singularity can be captured by using special elements arranged concentrically at the crack tip, *e.g.*, quadrilateral finite elements where two nodes in parametric space map to the crack tip location in physical space [2], or quarter-point finite elements, where the mid-size nodes of quadratic Lagrange finite elements are moved closer to the crack tip [3]. Although these special meshes can capture the stress singularity, generating them is a time-consuming process. Creating matching meshes in general can be quite challenging depending on the morphology of the problem, particularly because of the strict conditions on mesh quality that are required—elements with bad aspect ratios which could reduce the approximation accuracy are not allowed [4]; moreover, robustness is still questionable, especially for 3-D FE meshes, as issues remain in generating meshes that correctly match boundaries [5]. Numerical techniques such as Universal Meshes [6] and the Conforming to Interface Structured Adaptive Mesh Refinement (CISAMR) [7] have been proposed to modify meshes locally to discontinuities. These methods can therefore guarantee adequate discretizations for the analysis by ensuring elements with proper aspect ratios—which are crucial for recovering accurate gradient fields—while avoiding the need for global remeshing.

The eXtended/Generalized Finite Element Method (X/GFEM) [8–10] solves the aforementioned shortcomings in an elegant manner by enriching the approximation space with functions that can reproduce the discontinuities [11, 12]. In X/GFEM, the discretization is then decoupled from discontinuities by detecting cut elements, and by adding enriched DOFs to the standard mesh nodes of cut elements so that discontinuities can be resolved by the formulation. Although X/GFEM provides great flexibility in the choice of underlying FE discretizations, many properties that are usually taken for granted in standard FEM are lost, including the physical meaning of DOFs (unless shifting is used [13]), the straightforward imposition of essential (Dirichlet) boundary conditions [14, 15], and most importantly, the stability—here understood as the condition number of the resulting system matrix, which in stan-

dard FEM scales with mesh size $h$ as $\mathcal{O}\left(h^{-2}\right)$. Indeed, several enrichment functions could result in an unstable formulation where the condition number of the system matrix grows much faster than that of standard FEM. Pursuing Stable GFEM (SGFEM) that is devoid of this issue has been the focus of recent research in the field [16], with SGFEM proposed for weak discontinuities in [17, 18] and for strong discontinuities in [19–21]. Moreover, depending on the type of enrichment functions used, special techniques are needed in X/GFEM to prevent the loss of accuracy in *blending elements*, *i.e.*, non-cut elements that share enriched DOFs [22–24]. Finally, the computer implementation of X/GFEM is also more involved than that of standard FEM [25] since a variable number of DOFs per mesh node is required, and quadrature rules used depend on the type of enrichment functions—for fracture problems, for instance, elements with singular enrichments require special integration techniques [26].

Within the realm of enriched formulations, the Discontinuity-Enriched Finite Element Method presents a new paradigm for the mesh-independent analysis of weak and strong discontinuities [1]. DE-FEM can be seen as a technique that combines X/GFEM's most salient mesh-independent property—by decoupling the FE discretization from the problem's complex geometric features—while keeping the attractive properties of standard FEM. Indeed, in DE-FEM enrichment functions vanish at mesh nodes by construction, and as a consequence DOFs associated to the latter keep their physical interpretation, *e.g.*, displacement or temperature at the node location. In addition, essential boundary conditions can be prescribed directly as in standard FEM—both standard and enriched DOFs [27]. The computer implementation is also straightforward since the new enriched nodes are added to the same data structure used to store the original mesh nodes, and their corresponding enriched DOFs are retrieved from the solution vector in the same way as standard DOFs. Because DE-FEM was designed for simplicity in the formulation (and thus in the implementation), enrichment functions that capture the stress singularities along crack fronts are not used (although they may also be added). As a result, convergence rates for singular problems are not optimal.

In the absence of strong discontinuities, the method simplifies to the Interface-enriched Generalized Finite Element Method (IGFEM) [28] or to the Hierarchical Interface-enriched Finite Element Method (HIFEM) [29], which were devised for weak discontinuities alone [30–36]. In fact, HIFEM builds on IGFEM to resolve multiple weak discontinuities within a single element—and even intersecting discontinuities — by means of a hierarchical construction of enrichment functions. Given the comprehensive literature on the study of weak discontinuities with IGFEM/HIFEM, here we devote ourselves to problems containing strong discontinuities and to their interactions with weak discontinuities.

DE-FEM was first proposed for analyzing 2-D problems in elastostatics. In this chapter we extend DE-FEM to 3-D, and we discuss in detail its computer implementation in a displacement-based finite element code. In particular, we discuss algorithmic considerations for a 3-D geometric engine that handles the interactions between the background (original) mesh and material interfaces and/or cracks. By means of a discontinuous patch test, we demonstrate DE-FEM can generate two

independent kinematic fields as long as proper care is taken in the construction of a *conforming integration mesh*. A convergence study then demonstrates the accuracy and convergence rates of the method, which is then used to extract stress intensity factors (SIFs). Most importantly, we show that DE-FEM is a stable formulation, *i.e.*, the condition number of the stiffness matrix grows at roughly the same rate as that of standard FEM. Finally, we showcase DE-FEM in the challenging problem of intersecting weak and strong discontinuities, which is handled effortlessly by means of the hierarchical implementation *à la* HIFEM.

## 2.2. Formulation

Consider the elastostatics boundary value problem for a cracked body $\Omega \subset \mathbb{R}^3$ with closure $\overline{\Omega}$ and boundary $\partial\Omega \equiv \Gamma = \overline{\Omega} \setminus \Omega$. The body is subjected to a non-homogeneous essential boundary condition (BC) $\bar{\boldsymbol{u}} : \Gamma_u$ and to a traction BC $\bar{\boldsymbol{t}} : \Gamma_t$, the latter having a zero value along a crack $\Gamma_c \subset \Gamma_t$, as shown in Figure 2.1. The

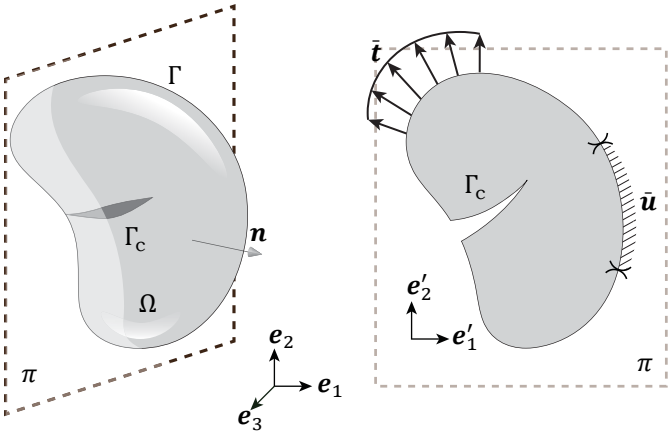



Figure 2.1: A 3-D body $\Omega$ containing a traction-free crack $\Gamma_c$ (left); Prescribed Dirichlet and Neumann boundary conditions, $\bar{\boldsymbol{u}}$ on $\Gamma_u$ and $\bar{\boldsymbol{t}}$ on $\Gamma_t$, respectively, are visualized on a plane $\pi$ that slices the domain with local coordinates $\{\boldsymbol{e}'_i\}$ (right).

abstract form of the weak (variational) formulation is: Given the displacement field $\boldsymbol{u} = \boldsymbol{v} + \tilde{\boldsymbol{u}}$, find $\boldsymbol{v} \in \mathcal{V}_0(\Omega)$ such that

$$B(\boldsymbol{v}, \boldsymbol{w}) = L(\boldsymbol{w}) - B(\tilde{\boldsymbol{u}}, \boldsymbol{w}) \qquad \forall\, \boldsymbol{w} \in \mathcal{V}_0, \tag{2.1}$$

where $L(\cdot)$ and $B(\cdot)$ are the linear and bilinear forms, respectively, $\tilde{\boldsymbol{u}}$ is a vector-valued function that satisfies $\tilde{\boldsymbol{u}}|_{\Gamma_u} = \bar{\boldsymbol{u}}$, and $\mathcal{V}_0$ is the vector-valued function space with components in $\mathcal{H}^1_0(\Omega)$ (Sobolev space that satisfies homogeneous essential boundary conditions on $\Gamma_u$). The linear and bilinear forms are given, respectively, by

$$L(\boldsymbol{w}) = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{w}\, \mathrm{d}\Omega + \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \boldsymbol{w}\, \mathrm{d}\Gamma. \tag{2.2}$$

and

$$B(\boldsymbol{v},\boldsymbol{w}) = \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{w}) \, \mathrm{d}\Omega, \tag{2.3}$$

In Equation (2.2), $\boldsymbol{b} : \Omega \to \mathbb{R}^3$ is the body force, and in Equation (2.3) the stress tensor $\boldsymbol{\sigma} : \overline{\Omega} \to \mathbb{R}^3 \times \mathbb{R}^3$ obeys Hooke's law on the linearized strain, *i.e.*, $\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}(\boldsymbol{v}) = \frac{1}{2}\left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\intercal}\right)$. In order to solve the finite-dimensional form of Equation (2.1), the domain is discretized into finite elements $e_i$ (simplexes in particular) such that $\cup_i \overline{e}_i = \overline{\Omega}^h \approx \overline{\Omega}$. Following a Bubnov-Galerkin approach, both the trial solution and the weight function are chosen from the DE-FEM space:

$$\mathcal{S}_e^h = \left\{ \boldsymbol{v}^h(\boldsymbol{x}) \middle| \; \boldsymbol{v}^h(\boldsymbol{x}) = \underbrace{\sum_{i \in \iota_h} N_i(\boldsymbol{x}) \boldsymbol{U}_i}_{\text{standard FEM}} + \underbrace{\overbrace{\sum_{i \in \iota_w} \psi_i(\boldsymbol{x}) \boldsymbol{\alpha}_i}^{\text{weak}} + \overbrace{\sum_{i \in \iota_s} \chi_i(\boldsymbol{x}) \boldsymbol{\beta}_i}^{\text{strong}}}_{\text{enrichments}}, \; \boldsymbol{U}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \in \mathbb{R}^3 \right\}, \tag{2.4}$$

where the standard FE space (first term) is augmented by an enriched space with special enrichment functions that can capture the jump in the gradient (weak term) and the primal (strong term) fields. In the standard FEM term, $\iota_h$ is the index set containing all original mesh nodes; $N_i$ denotes the $i$th Lagrange shape function and $\boldsymbol{U}_i$ the corresponding standard DOFs. In the enriched term, $\iota_w$ and $\iota_s$ are index sets of enriched nodes created along discontinuities. For weak discontinuities, these nodes are associated with enrichment functions $\psi_i$ and corresponding enriched DOFs $\boldsymbol{\alpha}_i$. In addition to these, strong discontinuities add a new set of nodes associated with enrichment functions $\chi_i$ and enriched DOFs $\boldsymbol{\beta}_i$.

In order to have an arbitrary number of discontinuities interact with a single element, we follow the work of Soghrati [29]. The procedure is schematically shown in Figure 2.2, where an element $e_i$ is split by two discontinuities. The first discon-

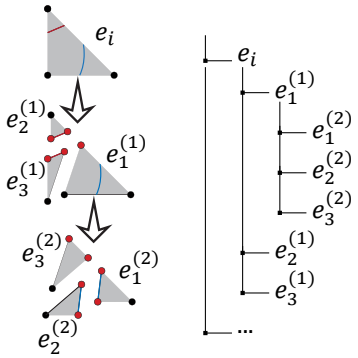



Figure 2.2: Schematic representation of the hierarchical creation of integration subdomains for an element $e_i$ split by two discontinuities ($D = 2$). The hierarchical information is stored in an ordered tree that is also used for constructing enrichment functions during the numerical quadrature of local element arrays.

tinuity creates a first level of integration subdomains, and these in turn become the parent elements for a second discontinuity. These relationships are stored in an ordered tree data structure, which is used during numerical quadrature to build the enrichment functions in a hierarchical manner. Therefore, for the $e$th cut element, the approximation $\boldsymbol{u}^h \in \mathcal{S}_e^h$ can be written as

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i\in\iota_h} N_i(\boldsymbol{x})\,\boldsymbol{U}_i + \sum_{k\in h}\sum_{i\in\iota_w} \psi_{ki}(\boldsymbol{x})\,\boldsymbol{\alpha}_{ki} + \sum_{k\in h_s}\sum_{i\in\iota_s} \chi_{ki}(\boldsymbol{x})\boldsymbol{\beta}_{ki}, \tag{2.5}$$

where $h = \{1, \dots, D\}$ is the index set used to represent $D$ discontinuities (both weak and strong) that interact with the element, and $h_s \subseteq h$ only includes the indices for strong discontinuities. The formulation in uncut elements remains the same as in standard FEM, *i.e.*, only the first term in Equation (2.5) is used.

For the sake of simplicity, we denote by $\boldsymbol{\varphi} \equiv \begin{bmatrix} N_1 \dots & \psi_1 \dots & \chi_1 \dots \end{bmatrix} = \begin{bmatrix} \boldsymbol{N} & \boldsymbol{\psi} & \boldsymbol{\chi} \end{bmatrix}$ a vector that stacks shape and enrichment functions that are non-zero on the $e$th element and by $\boldsymbol{B} \equiv \boldsymbol{\partial}\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\partial N} & \boldsymbol{\partial\psi} & \boldsymbol{\partial\chi} \end{bmatrix}$ the strain-displacement matrix, which is obtained by applying to $\boldsymbol{\varphi}$ the differential operator $\boldsymbol{\partial}$:

$$\boldsymbol{\partial} \equiv \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^{\intercal}. \tag{2.6}$$

The element local stiffness matrix and the nodal load vector are given, respectively, by

$$\boldsymbol{k}_e = \int_e \boldsymbol{B}^{\intercal}\mathbf{C}\boldsymbol{B}\,\mathrm{d}e \quad \text{and} \quad \boldsymbol{f}_e = \int_e \boldsymbol{\varphi}^{\intercal}\boldsymbol{b}\,\mathrm{d}e + \int_{\partial e} \boldsymbol{\varphi}^{\intercal}\bar{\boldsymbol{t}}\,\mathrm{d}\partial e, \tag{2.7}$$

where $\mathbf{C}$ is the constitutive matrix given by

$$\mathbf{C} = \begin{bmatrix} \lambda\mathbb{1} + \mu\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mu\mathbf{I} \end{bmatrix}, \tag{2.8}$$

$\lambda$ and $\mu$ are the Lamé constants, and $\mathbb{1}$ and $\mathbf{I}$ are the unity and identity matrices, respectively. Details about the construction of $\boldsymbol{k}_e$ and $\boldsymbol{f}_e$, together with pseudo-code for constructing the enrichment functions in a hierarchical manner, are given later in Chapter 2.4.

The finite-dimensional form obtained by taking into account all finite elements

$$\sum_{e\in\Omega^h} B_e\left(\boldsymbol{v}^h, \boldsymbol{w}^h\right) = \sum_{e\in\Omega^h} L_e\left(\boldsymbol{w}^h\right) - \sum_{e\in\Omega^h} B_e\left(\tilde{\boldsymbol{u}}^h, \boldsymbol{w}^h\right) \qquad \forall\, \boldsymbol{w}^h \in \mathcal{S}_e^h, \tag{2.9}$$

leads to an augmented system of linear equations

$$\underbrace{\begin{bmatrix} \boldsymbol{K}_{uu} & \boldsymbol{K}_{u\alpha} & \boldsymbol{K}_{u\beta} \\ \boldsymbol{K}_{u\alpha}^{\intercal} & \boldsymbol{K}_{\alpha\alpha} & \boldsymbol{K}_{\alpha\beta} \\ \boldsymbol{K}_{u\beta}^{\intercal} & \boldsymbol{K}_{\alpha\beta}^{\intercal} & \boldsymbol{K}_{\beta\beta} \end{bmatrix}}_{\boldsymbol{K}} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{F}_u \\ \boldsymbol{F}_\alpha \\ \boldsymbol{F}_\beta \end{bmatrix}}_{\boldsymbol{F}}. \tag{2.10}$$

In Equation (2.10) we made explicit the partition of global arrays into standard and enriched parts. Aragón and Simone [1] discuss the formulation in more detail for 2-D elastostatics, considering also the case of cohesive tractions acting on cracks.

## 2.3. Geometric engine



The complexity of mesh generation in standard FEM is transferred in DE-FEM to a geometric engine that takes care of the interaction between a background mesh (usually structured) and the discontinuities. The engine needs to be designed for handling complex scenarios with multiple discontinuities, as shown for instance in Figure 2.3, and to handle not only cracks (strong discontinuities) but also material interfaces (weak discontinuities).

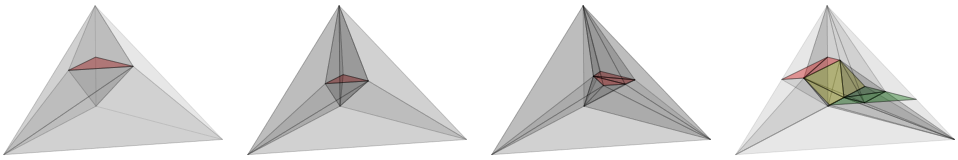



Figure 2.3: Configurations considered when designing the geometric engine include (a) partially cut, (b) internal, (c) irregular, and (d) multiple discontinuities.

The operations of the geometric engine are summarized in the flow chart of Figure 2.4. The engine takes as input the background mesh and the set of discontinuities. In this work, FE models are discretized using linear tetrahedra, cracks are described explicitly through planar regions (analogous to line segments in 2-D), and material interfaces are described implicitly through level set functions.

The engine first loops over discontinuities and then loops over mesh elements for each discontinuity. Tetrahedral elements cut by the first discontinuity are detected. The spatial locations of new enriched nodes is then determined by performing intersection tests between the discontinuity and element sides. At these locations we create the corresponding number of enriched nodes depending on the type of discontinuity; one enriched node per intersection (associated with weak DOFs) is generated if the discontinuity is weak. If the discontinuity is strong, two enriched nodes are created per intersection (each associated with either weak or strong DOFs) unless the node is at the crack front (associated with weak enriched DOFs only). These enriched nodes, together with those of the original mesh, are later used to create tetrahedral integration elements.

It is worth noting that the engine described here would not be efficient when dealing with a great number of discontinuities. Conducting intersection tests for every element-discontinuity pair leads to the so called *all-pairs weakness*, which results in an algorithm of complexity $\mathcal{O}(DE)$ for $E$ mesh elements and $D$ discontinuities. The efficiency of the interaction can be greatly improved by performing heuristics on bounding boxes. Furthermore, space partitioning techniques can speedup

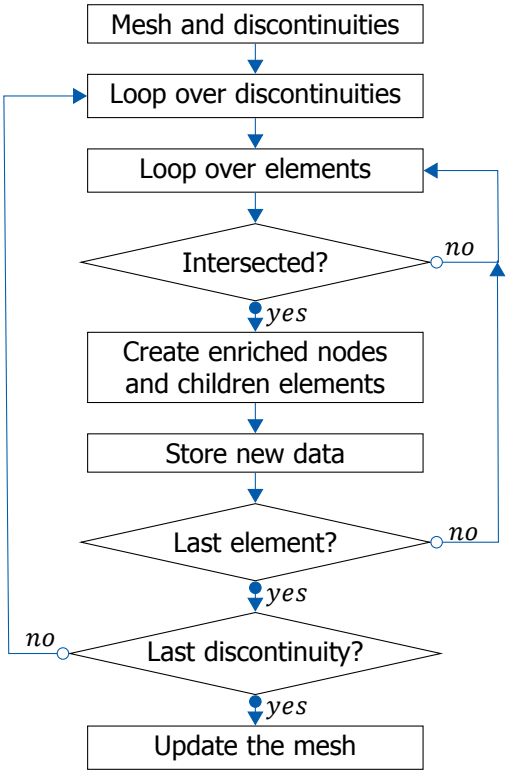



Figure 2.4: Geometric engine's procedure for handling models with multiple discontinuities hierarchically.

finding elements intersected by discontinuities. Among these one could use octrees, $k$-d trees, R*-trees, and grids [37]. Finally, these techniques could be supplemented with fast marching methods and graph traversal algorithms to look only for neighboring elements of already cut elements.

A crucial task of the geometric engine is to determine the location of crack fronts. Figure 2.5 shows that a planar region, which represents a 3-D crack, intersects with a tetrahedral element partially. In order to determine the location of nodes along

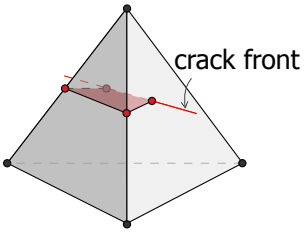



Figure 2.5: The enriched nodes along the crack front are generated when a linear tetrahedral element is split by a crack partially.

the crack front, intersections tests are performed between the crack and all faces of the tetrahedron. Numerical precision is of major concern when dealing with floating point arithmetic, especially for determining whether a point lies on an edge or on a face. In this work we adopt a tolerance that depends on the mesh size $h$ for finding intersections. Other operations performed by the geometric engine include creating groups of nodes and elements that are associated with discontinuities (these are used for assigning the correct material properties), generating integration elements, and updating the original mesh data structure.



### 2.3.1. Creation of integration elements

Although it is not strictly necessary, creating an *integration tetrahedral mesh* is convenient for four reasons: *i)* Lagrange shape functions in integration tetrahedra are used to construct the enrichment functions. Storing integration elements in an ordered tree data structure further facilitates the construction of enrichment functions at each level of hierarchy—the resulting recursive quadrature algorithm is general and its computer implementation is straightforward; *ii)* By creating integration elements at either side of the discontinuities we ensure the enrichment functions are smooth, and thus they can be integrated with the minimum number of integration points (parent shape and enrichment functions are linear, and thus their corresponding derivatives are constant); *iii)* Tetrahedral elements are the simplest 3-D elements to split, so they ease the handling of multiple discontinuities within a single parent mesh element; and *iv)* Integration elements help in the post-processing stage so that results can be visualized correctly. It is worth noting that the creation of this *integration mesh* is a procedure bound locally to discontinuities, and as such it is completely different than performing remeshing.

Efficient algorithms for splitting a tetrahedron is one of our main concerns. There are four distinct configurations that result from completely splitting a tetrahedron with a planar region [34], as shown in Figures 2.6(a)-(d): $(a)$ two tetrahedra; $(b)$ a tetrahedron and a pyramid; $(c)$ a tetrahedron and a prism; and $(d)$ two prisms. A pyramid can be considered to be composed by two tetrahedra (Figure 2.6(e)). Therefore, we can get three new tetrahedral elements in case $(b)$. Normally, tetrahedralizing a prism (Figure 2.6(f)) can produce three tetrahedra under the premise that any two diagonals of three quadrilateral sides share the same vertex. However, in the case that the diagonals of the body are already prescribed (for example by the fact that neighboring elements have been tetrahedralized), it might be impossible to create new tetrahedra directly. This situation is alleviated by adding an extra vertex (called Steiner point) inside the prism (see Figure 2.6(g)). Then, this prism can be split into eight tetrahedra that share this vertex.

While fully splitting a tetrahedron into several tetrahedra can be accomplished by using basic splitting rules [38], the complexity of the geometric operations increases rapidly when dealing with partial splits (see Figure 2.7) or when a crack resides completely inside a tetrahedron. The latter situation could be alleviated by refining the mesh, but this in turn increases the computational cost. Hence, in this work we use constrained Delaunay tetrahedralization for creating new tetrahedral subdomains [39].

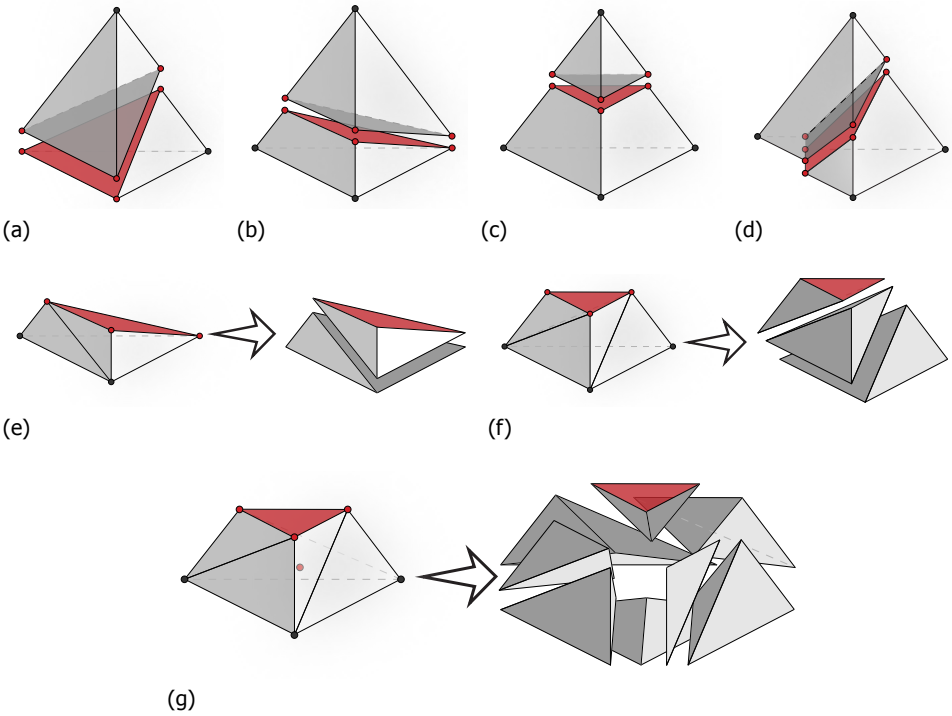



Figure 2.6: Configurations that result from splitting completely a tetrahedron: (a) two tetrahedra; (b) one tetrahedron and one pyramid; (c) one tetrahedron and one prism; and (d) two prisms. One of possible situations of splitting (e) a pyramid into two tetrahedra and (f) a prism into three tetrahedra. (g) An untetrahedralizable case for subdividing a prism, which requires an extra vertex (called Steiner point) to create eight tetrahedra.

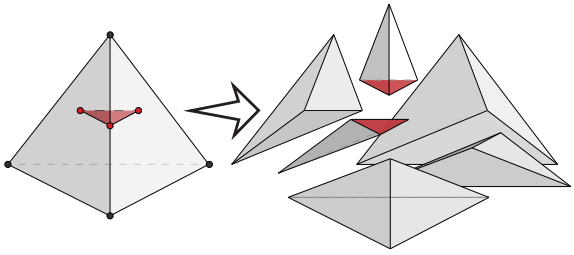


Figure 2.7: Example of a partially split tetrahedron, where six new tetrahedra are created.

Since each background element intersected by discontinuities is tetrahedralized in sequence, the *constrained* Delaunay algorithm is needed to avoid the creation of *a non-conforming integration mesh*, which as explained later has important consequences. Figure 2.8 shows such a case, where two contiguous tetrahedral elements split by a discontinuity could have sub-tetrahedra with non-matching faces. In other

words, there exists the possibility of creating integration tetrahedra in element $e_m$ with triangular faces $(1,2,4)$ and $(2,3,4)$, whereas those tetrahedra in element $e_n$ could have faces $(1,2,3)$ and $(1,3,4)$. In order to avoid this situation, the constrained Delaunay function sets the faces of already created integration elements as the constraints for new elements sharing the same face. It should be emphasized that we are not conducting remeshing since the partition of unity of the background mesh is kept intact, and constrained Delaunay is only used to locally generate integration elements; because these are just used for the numerical quadrature, their requirements in terms of aspect ratios are less strict than those needed for Delaunay meshes.

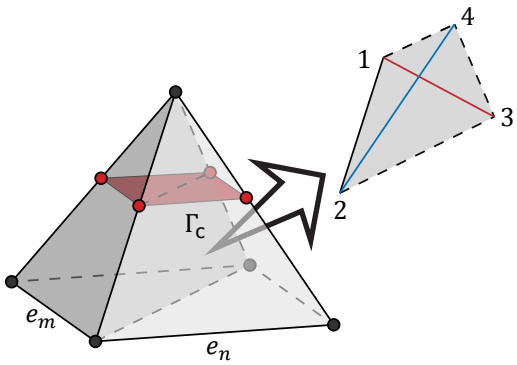



Figure 2.8: Two contiguous elements $e_m$ and $e_n$ are split by a discontinuity $\Gamma_c$. By not taking proper care, it is possible that a *non-conforming integration mesh* below the discontinuity is created, whereby integration tetrahedra at one side of surface (1,2,3,4) have faces (1,2,4) and (2,3,4), and at the other side (1,2,3) and (1,3,4).

The discussion above deals only with volumetric elements. It should be noted, however, that lower-dimensional elements (3-node triangles and 2-node lines) are also needed at times, *e.g.*, to apply boundary conditions—consider a distributed pressure applied to triangular faces of tetrahedra split by a crack. Therefore, lower-dimensional elements are also extracted from the created volumetric integration elements.

### 2.3.2. Hierarchical bookkeeping

An ordered tree data structure is used to store the relationships between original (parent) and new (children) elements. This hierarchical structure later facilitates the calculation of weak and strong enrichment functions during the numerical quadrature of local element arrays. Children integration elements can in turn become parents of new children elements deeper in the hierarchy created by subsequent discontinuities.

The procedure is illustrated in Figure 2.9 for a tetrahedral element $e_i$ cut by two discontinuities $\Gamma_{c1}$ and $\Gamma_{c2}$. This mesh element becomes the parent element for integration elements generated at the first hierarchical level by the first discon-

tinuity $\Gamma_{c1}$. These children elements in turn become the parents for newly created integration elements at the second hierarchical level by a second discontinuity $\Gamma_{c2}$. The final hierarchical structure is also illustrated in Figure 2.9.



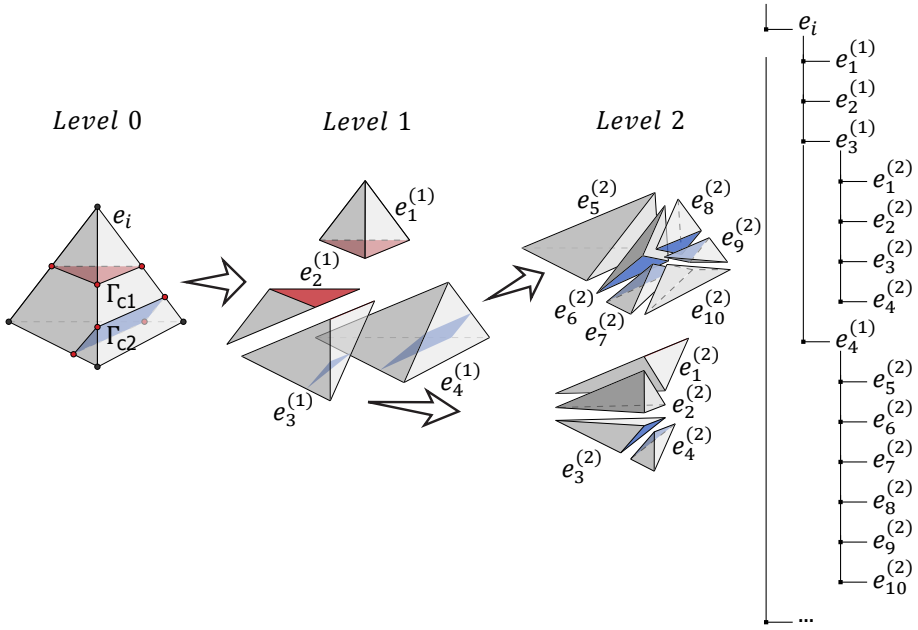



Figure 2.9: Mesh tetrahedral element $e_i$ intersected first by $\Gamma_{c1}$ (which creates the first level of integration elements) and then by $\Gamma_{c2}$ (which creates the second level). The generated children elements are stored in an ordered tree hierarchical data structure (right).

This procedure can be used to create integration elements for mesh elements split by an arbitrary number of discontinuities. An element split by $n$ discontinuities has $n+1$ hierarchical levels in the tree, and this tree has the depth of the element split with the most number of discontinuities.

## 2.4. Enrichment functions

In order to calculate the element stiffness matrix $\boldsymbol{k}_e$ and force vector $\boldsymbol{f}_e$ in Equation (2.7), we need to evaluate shape and enrichment functions, together with their derivatives. Lagrange shape functions $\boldsymbol{N}$ are obtained as in standard FEM, following an iso-parametric formulation. Here we focus on the evaluation of enrichment functions $\boldsymbol{\psi}$ and $\boldsymbol{\chi}$.

### 2.4.1. Single discontinuity

Consider a tetrahedral element $e_i$ intersected by a single crack $\Gamma_c$, as shown in Figure 2.10. Because the element is split by a single discontinuity, all children elements

created will be leaves (integration elements) in the ordered tree. The orientation of the crack $\Gamma_c$, given by its normal $\boldsymbol{n}$, is used to define positive and negative regions. Three intersection points, labeled $\boldsymbol{x}_i, \boldsymbol{x}_j$, and $\boldsymbol{x}_k$, are determined by the geometric engine. Enriched nodes are then created at these locations and denoted with the same symbols henceforth for simplicity[1]. A positive (negative) sign superscript is used if the node is referred in the positive (negative) side of the crack. In addition, $\Gamma_c$ subdivides $e_i$ into four children (integration) elements such that $e_1^{(1)}$ lays on at the positive side, and $e_2^{(1)}, e_3^{(1)}$, and $e_4^{(1)}$ on the negative side. Enrichment functions will be constructed with the aid of standard Lagrange shape functions in these integration elements.

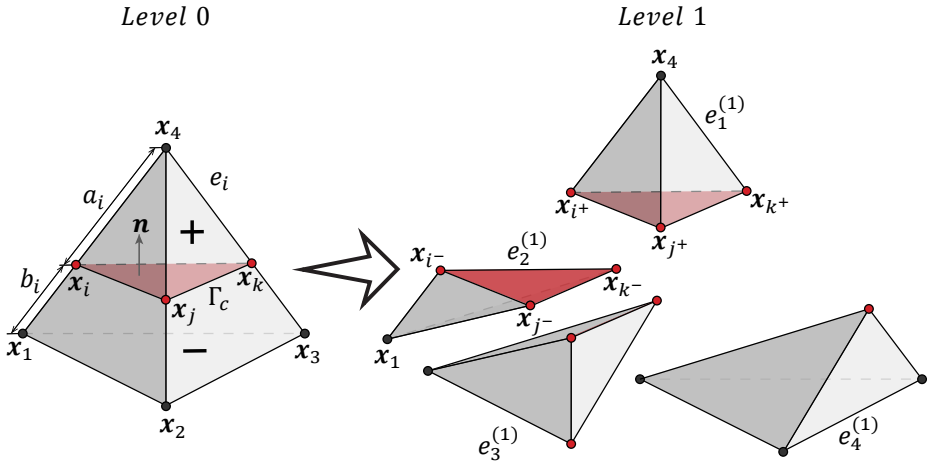



Figure 2.10: Three enriched nodes $\boldsymbol{x}_i, \boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are created when a tetrahedral element $e_i$ is subdivided into four children element $e_1^{(1)}, e_2^{(1)}, e_3^{(1)}$ and $e_4^{(1)}$ by one discontinuity $\Gamma_c$.

Now consider the leaf integration element $e_1^{(1)}$ on the positive side, which has three enriched nodes $\boldsymbol{x}_{i^+}, \boldsymbol{x}_{j^+}$, and $\boldsymbol{x}_{k^+}$. The Lagrange shape functions in this element, *i.e.*, $N_{\iota^+}^{(1)}(\boldsymbol{x}), \iota = \{i, j, k\}$, are used to create the weak and strong enrichment functions acting on this element. These functions can be expressed as

$$\psi_{i^+} = w_{i^+} N_{i^+}^{(1)}(\boldsymbol{x}), \qquad \psi_{j^+} = w_{j^+} N_{j^+}^{(1)}(\boldsymbol{x}), \qquad \psi_{k^+} = w_{k^+} N_{k^+}^{(1)}(\boldsymbol{x}), \tag{2.11}$$

and

$$\chi_{i^+} = s_{i^+} N_{i^+}^{(1)}(\boldsymbol{x}), \qquad \chi_{j^+} = s_{j^+} N_{j^+}^{(1)}(\boldsymbol{x}), \qquad \chi_{k^+} = s_{k^+} N_{k^+}^{(1)}(\boldsymbol{x}), \tag{2.12}$$

where $w_{\iota^+}$ and $s_{\iota^+}$ are weak and strong scaling (weight) factors, respectively, yet to be determined.

[1] As mentioned earlier, in the computer implementation it is convenient to create double nodes at each location when dealing with strong discontinuities; weak DOFs are associated to one of the nodes and strong DOFs to the other.

Similarly, for a children element on the negative side, such as $e_2^{(1)}$, weak and strong enrichment functions acting on the element are given by

$$\psi_{i^-} = w_{i^-} N_{i^-}^{(1)}(\boldsymbol{x}), \qquad \psi_{j^-} = w_{j^-} N_{j^-}^{(1)}(\boldsymbol{x}), \qquad \psi_{k^-} = w_{k^-} N_{k^-}^{(1)}(\boldsymbol{x}), \tag{2.13}$$

and

$$\chi_{i^-} = s_{i^-} N_{i^-}^{(1)}(\boldsymbol{x}), \qquad \chi_{j^-} = s_{j^-} N_{j^-}^{(1)}(\boldsymbol{x}), \qquad \chi_{k^-} = s_{k^-} N_{k^-}^{(1)}(\boldsymbol{x}), \tag{2.14}$$

where $w_{\iota^-}$ and $s_{\iota^-}$ are the $\iota^-$th enriched node's scaling factors for the negative side, and $N_{\iota^-}^{(1)}(\boldsymbol{x})$ are the Lagrange shape functions of this integration element. In order to enforce continuity of the weak enrichment function, the weak scaling factors in both sides are given by the same expression (*i.e.*, $w_{\iota^+} = w_{\iota^-} = w_\iota$). For the weak scaling, we use a theoretically derived optimal function in order to preserve stability in the formulation [40]: $w_\iota = \sqrt{2\zeta_\iota(1-\zeta_\iota)}$, where $0 \le \zeta_\iota \le 1$ is the relative length between the enriched and original nodes measured along the corresponding element edge. For enriched node $\boldsymbol{x}_i$, for instance, $\zeta_i$ is either $a_i/(a_i+b_i)$ *or* $b_i/(a_i+b_i)$ (since $w_\iota$ is symmetric), where $a_i, b_i$ are the Euclidean distances between $\boldsymbol{x}_i$ and $\boldsymbol{x}_4, \boldsymbol{x}_1$ measured along the edge of the parent tetrahedron (see Figure 2.10). The strong scaling factors are responsible for the field jump. Similarly to DE-FEM in 2-D [1], we choose the strong scaling factors so that the field jump has a unit magnitude, *i.e.*, $[\![\chi_\iota]\!] = \chi|_{\boldsymbol{x}_{\iota^+}} - \chi|_{\boldsymbol{x}_{\iota^-}} = 1$. As a result, the associated strong DOFs $\boldsymbol{\beta}_\iota$ physically represent the displacement jump at $\boldsymbol{x}_\iota$. For example, the strong scalings corresponding to the enriched node $\boldsymbol{x}_i$ are accomplished by setting $s_{i^+} = -\zeta_i$ and $s_{i^-} = 1-\zeta_i$. The weak and strong scaling factors for enriched nodes $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are obtained analogously.

For enriched node $\boldsymbol{x}_i$, which is connected to integration elements $e_1^{(1)}$ and $e_2^{(1)}$, the enrichment functions $\psi_i$ and $\chi_i$ are:

$$\psi_i = \begin{cases} w_i N_{i^+}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_1^{(1)} \\ w_i N_{i^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_2^{(1)} \end{cases}, \tag{2.15}$$

and

$$\chi_i = \begin{cases} s_{i^+} N_{i^+}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_1^{(1)} \\ s_{i^-} N_{i^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_2^{(1)} \end{cases}, \tag{2.16}$$

respectively. Figure 2.11 shows a schematic representation of weak and strong enrichment functions neglecting scaling factors. For modeling a discontinuous field gradient, the value of weak enrichment function $\psi_i$ arrives maximum at $\boldsymbol{x}_i$, and decreases linearly to 0 at other element nodes (both standard and enriched nodes).

It is worth mentioning that in DE-FEM the condition $[\![\chi_\iota]\!] = \chi|_{\boldsymbol{x}_{\iota^+}} - \chi|_{\boldsymbol{x}_{\iota^-}} = 1, \iota = \{i, j, k\}$ is fulfilled by using the partition of unity property of the background mesh element [1]. Specifically, for enriched node $\boldsymbol{x}_i$, Equation (2.16) is equivalent to using one parent nodal shape function on one side of the discontinuity and the

negative of another parent nodal shape function on the other side. To wit,

$$\chi_i = \begin{cases} N_1^{(0)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_1^{(1)} \\ -N_4^{(0)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_2^{(1)} \end{cases}, \tag{2.17}$$

where the superscript $(0)$ is used to indicate that the function belongs to the parent mesh element.

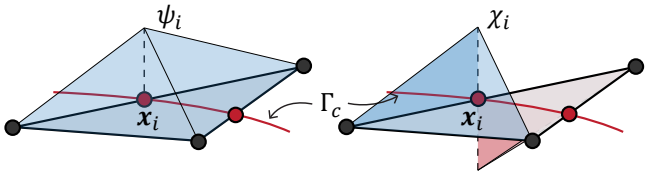



Figure 2.11: Weak and strong enrichment functions $\psi_i$ and $\chi_i$ of enriched node $\boldsymbol{x}_i$ acting on a triangular surface of a split tetrahedron.

### 2.4.2. Multiple discontinuities

We now consider the more general situation where a mesh element is cut by more than one discontinuity. For constructing enrichment functions in a hierarchical way, we traverse the tree starting from leaf elements. Consider the case shown in Figure 2.12, which shows a mesh element crossed by two non-intersected discontinuities. The figure shows the hierarchical relationship between $e_i$ (mesh element),

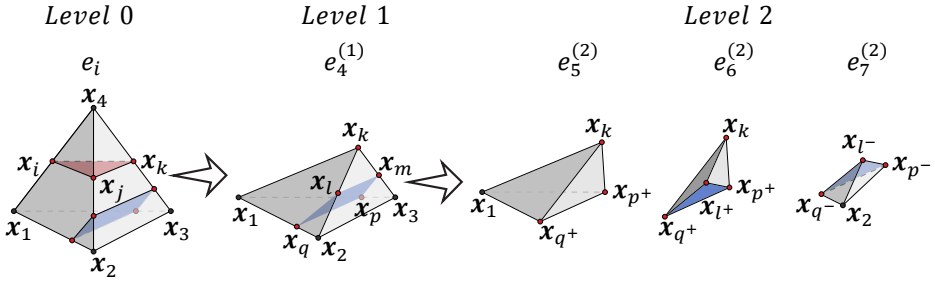



Figure 2.12: The hierarchical relationship between the original parent element $e_i$, the element $e_4^{(1)}$ generated by the first discontinuity, the integration elements $e_5^{(2)}$, $e_6^{(2)}$, and $e_7^{(2)}$ in the last level. The enriched nodes $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$ are created by the first discontinuity, and $\boldsymbol{x}_l$, $\boldsymbol{x}_m$, $\boldsymbol{x}_p$, and $\boldsymbol{x}_q$ are generated when handling the second one.

$e_4^{(1)}$ (at hierarchical level 1), and $e_5^{(2)}$, $e_6^{(2)}$, and $e_7^{(2)}$ (at hierarchical level 2). We take enriched node $\boldsymbol{x}_q$ created after splitting children element $e_4^{(1)}$ by a second discontinuity. This enriched node is associated with leaf integration elements $e_5^{(2)}$, $e_6^{(2)}$ and $e_7^{(2)}$ at the second hierarchical level. We follow the same strategy as that used for handling one discontinuity to construct the enrichment functions associated with

$\boldsymbol{x}_q$. To wit,

$$\psi_q = \begin{cases} w_q N_{q^+}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_5^{(2)} \\ w_q N_{q^+}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_6^{(2)} \\ w_q N_{q^-}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_7^{(2)} \end{cases}, \tag{2.18}$$

and

$$\chi_q = \begin{cases} s_{q^+} N_{q^+}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_5^{(2)} \\ s_{q^+} N_{q^+}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_6^{(2)} \\ s_{q^-} N_{q^-}^{(2)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_7^{(2)} \end{cases}. \tag{2.19}$$

where $w_q$, $s_{q^\pm}$ are, as before, the weight factors, and $N_\iota^{(2)}(\boldsymbol{x}), \iota = \{q^+, q^-\}$ the Lagrange shape functions. It should be mentioned that an enrichment function at the second level of hierarchy—deepest level in this example—is non-zero only in elements connected to its corresponding enriched node (the enriched node's support). The function is exactly zero in integration elements created at shallower hierarchical levels outside the enriched node's support.

The same strategy is used to obtain enrichment functions for any enriched node at any hierarchical level. Integration element $e_4^{(1)}$, that is created by the first discontinuity, is connected to mesh nodes $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ and to enriched node $\boldsymbol{x}_k$. For the latter, the weak and strong enrichment functions $\psi_k$ and $\chi_k$ are obtained, respectively, by

$$\psi_k = \begin{cases} w_k N_{k^+}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_1^{(1)} \\ w_k N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_2^{(1)} \\ w_k N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_3^{(1)} \\ w_k N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_4^{(1)} \end{cases}, \tag{2.20}$$

and

$$\chi_k = \begin{cases} s_{k^+} N_{k^+}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_1^{(1)} \\ s_{k^-} N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_2^{(1)} \\ s_{k^-} N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_3^{(1)} \\ s_{k^-} N_{k^-}^{(1)}(\boldsymbol{x}) & \text{for } \boldsymbol{x} \in e_4^{(1)} \end{cases}. \tag{2.21}$$

After evaluating enrichment functions (and their derivatives), we append them to the Lagrange shape functions of parent elements to calculate the vector $\boldsymbol{\varphi}$ (and the derivatives are appended to those of parent shape functions to get the strain-displacement matrix $\boldsymbol{B}$). For instance, for one of the integration elements in the second level of hierarchy, such as $e_5^{(2)}$, the arrays $\boldsymbol{\varphi}$ and $\boldsymbol{B}$ are given by

$$\boldsymbol{\varphi} = \left[ N_1^{(0)} \ldots N_4^{(0)} \quad w_k N_{k^-}^{(1)} \quad w_q N_{q^+}^{(2)} \quad w_p N_{p^+}^{(2)} \quad s_{k^-} N_{k^-}^{(1)} \quad s_{q^+} N_{q^+}^{(2)} \quad s_{p^+} N_{p^+}^{(2)} \right], \tag{2.22}$$

$$\boldsymbol{B} = \boldsymbol{\partial}\boldsymbol{\varphi}, \tag{2.23}$$

where $N_\iota^{(0)}(\boldsymbol{x}), \iota = \{1, 2, 3, 4\}$ are the Lagrange shape functions from the parent mesh element $e_i$.

Elements not intersected by discontinuities are handled as in the standard FEM. On cut elements, the hierarchical structure adopted for handling an arbitrary number of intersections eases the computer implementation for conducting numerical quadrature. As an alternative to recursion, Algorithms 2.1 and 2.2 outline pseudocode using an iterative loop for obtaining local arrays by Gauss integration. The algorithm shown is the same irrespective of the number of discontinuities.



**Algorithm 2.1** Local stiffness matrix and force vector for an integration element

**Input:** Modified mesh $\mathcal{M} = \{\mathcal{N} \cup \mathcal{N}_e, \mathcal{E} \cup \mathcal{H}\}$, ordered tree $\mathcal{H}$, scaling factor map $\mathcal{W}$, leaf integration element $e^{(k)}$ at hierarchical level $k$, quadrature point weights $\boldsymbol{\gamma}$ and coordinates $\boldsymbol{\xi}$, constitutive matrix $\mathbf{C}$ and body force $\boldsymbol{b}$;

```
1: function getLocalArrays(M, H, W, e^(k), γ, ξ, C, b)
2:         {k, f} ← {0, 0} – initialize local arrays
3:     for i ← 1, ..., |ξ| do – loop over integration points
4:         {j, N, B} ← functions(M, H, W, e^(k), ξ_i)
5:                   – get shape/enrichment functions (and their derivatives) and Jacobian
6:         k ← k+γ_i j B^T C B – update stiffness matrix
7:         f ← f+γ_i j N^T b – update nodal force vector
8:     return k, f
9: end function
```

**Output:** Local stiffness matrix $\boldsymbol{k}$ and local force vector $\boldsymbol{f}$.

In order to construct the local arrays for a (leaf) integration element $e^{(k)}$ at hierarchical level $k$, we input the modified mesh $\mathcal{M}$ consisting of nodes and elements. The node set is composed by the original mesh nodes $\mathcal{N}$ and the newly created enriched nodes $\mathcal{N}_e$. Similarly, the element set $\mathcal{E}$ is composed by the original uncut mesh elements and the hierarchical tree data structure $\mathcal{H}$, which contain not only all children elements but also cut mesh elements. The function `getLocalArrays` proceeds as in standard FEM, where the stiffness matrix $\boldsymbol{k}$ and the force vector $\boldsymbol{f}$ are obtained by performing Gauss quadrature. Specifically, at each quadrature point $\boldsymbol{\xi}_i$, the contributions to the local arrays are obtained by evaluating arrays of shape and enrichment functions $\boldsymbol{N}$ (and their derivatives $\boldsymbol{B}$), weighted by the product between the quadrature weight $\gamma_i$ and the jacobian $j$.

The differences with respect to standard FEM are accounted for in `functions`. For the given quadrature point, the idea is to traverse the hierarchy from the leaf element $e^{(k)}$ towards its root parent mesh element $e^{(0)}$, so that all enrichment functions (together with their derivatives) are calculated and stacked in their corresponding arrays along the way. Only then the partition of unity shape functions (and their derivatives) of the parent mesh element are accounted for.

Enrichment functions and their derivatives are obtained by first iterating over

**Algorithm 2.2** Shape and enrichment functions, together with their derivatives

**Input:** Modified mesh $\mathcal{M} := \{\mathcal{N} \cup \mathcal{N}_e, \mathcal{E} \cup \mathcal{H}\}$, ordered tree $\mathcal{H}$, scaling factor map $\mathcal{W}$, leaf integration element $e^{(k)}$ at hierarchical level $k$, quadrature point at this level labeled $\boldsymbol{\xi}_i^{(k)}$;

**function** functions$(\mathcal{M}, \mathcal{H}, \mathcal{W}, e^{(k)}, \boldsymbol{\xi}_i^{(k)})$
  $\boldsymbol{X} \leftarrow (\mathcal{N} \cup \mathcal{N}_e) \cap e^{(k)}$ – *get element coordinates*
  **for** $n_j \leftarrow \mathcal{N}_e \cap e^{(k)}$ **do** – *loop over element's enriched nodes*
    $\{w_j, s_j\} \leftarrow \mathcal{W}(n_j)$ – *get scaling factors for weak/strong enrichments at level $k$*
    **do**
      $\left\{\boldsymbol{N}^{(k)}, \boldsymbol{N}_{,\xi}^{(k)}\right\} \leftarrow$ Lagrange$(e^{(k)}, \boldsymbol{\xi}_i^{(k)})$ – *shape functions and their derivatives*
      $\boldsymbol{x} \leftarrow \boldsymbol{X}^{\mathsf{T}} \boldsymbol{N}^{(k)}$ – *obtain global coordinate*
      $\boldsymbol{J}^{(k)} \leftarrow \boldsymbol{X}^{\mathsf{T}} \boldsymbol{N}_{,\xi}^{(k)}$ – *obtain Jacobian matrix*
      **if not** $j$ **then** – *check if leaf Jacobian is set*
        $j \leftarrow \det\left(\boldsymbol{J}^{(k)}\right)$ – *save leaf Jacobian*
      $\boldsymbol{N}_{,x}^{(k)} \leftarrow \boldsymbol{N}_{,\xi}^{(k)} \left[\boldsymbol{J}^{(k)}\right]^{-1}$ – *compute derivatives with respect to $x$*
      $\left\{\psi^{(k)}, \boldsymbol{\psi}_{,x}^{(k)}\right\}, \left\{\chi^{(k)}, \boldsymbol{\chi}_{,x}^{(k)}\right\} \leftarrow$ enrich$(w_j, s_j, \boldsymbol{N}^{(k)}, \boldsymbol{N}_{,x}^{(k)})$
        – *compute enrichments/derivatives at level $k$*
      $\{\boldsymbol{F}, \boldsymbol{F}_{,x}\} \leftarrow \left\{\left[\boldsymbol{F} \quad \psi^{(k)} \quad \chi^{(k)}\right], \left[\boldsymbol{F}_{,x} \quad \boldsymbol{\psi}_{,x}^{(k)} \quad \boldsymbol{\chi}_{,x}^{(k)}\right]\right\}$
        – *stack functions and derivatives*
      $k \leftarrow k-1$ – *decrease level of hierarchy*
      $e^{(k)} \leftarrow \mathcal{H}(k, e^{(k)})$ – *get parent element in the hierarchy*
      $\boldsymbol{\xi}_i^{(k)} \leftarrow$ invert$(\boldsymbol{x}, e^{(k)})$ – *obtain local master coordinate (invert map)*
    **while** $k \neq 0$ – *if $k \leftarrow 0$ then parent mesh element reached*
  $\left\{\boldsymbol{N}^{(0)}, \boldsymbol{N}_{,\xi}^{(0)}\right\} \leftarrow$ Lagrange$(e^{(0)}, \boldsymbol{\xi}_i^{(0)})$ – *shape functions and their derivatives*
  $\{\boldsymbol{F}, \boldsymbol{F}_{,x}\} \leftarrow \left\{\left[\boldsymbol{N}^{(0)} \quad \boldsymbol{F}\right], \left[\boldsymbol{N}_{,x}^{(0)}, \boldsymbol{F}_{,x}\right]\right\}$ – *stack parent shape functions and their derivatives*
  $\{\boldsymbol{N}, \boldsymbol{B}\} \leftarrow \left\{\text{expand}\left(\boldsymbol{F}\right), \text{expand}\left(\boldsymbol{F}_{,x}\right)\right\}$ – *expand to multiple DOFs per node*
  **return** $j, \boldsymbol{N}, \boldsymbol{B}$
**end function**

**Output:** Jacobian determinant $j$, and arrays of shape and enrichment functions $\boldsymbol{N}$ together with their derivatives $\boldsymbol{B}$.

the enriched nodes belonging to the leaf element $e^{(k)}$. Depending on the type of discontinuity, contributions will be made for the weak term and possibly to the strong term when dealing with a strong discontinuity. Lagrange shape functions $\boldsymbol{N}^{(k)}$ and their derivatives $\boldsymbol{N}_{,\xi}^{(k)}$ are evaluated at $\boldsymbol{\xi}_i^{(k)}$. Shape functions are used to

calculate the global coordinates $\boldsymbol{x}$ and their derivatives to compute the Jacobian matrix $\boldsymbol{J}^{(k)}$ (which is used throughout to compute derivatives with respect to *global* coordinate $\boldsymbol{x}$). Notice that we save the determinant $j$ of Jacobian matrix of the leaf integration element only. In the function `enrich`, $\boldsymbol{N}^{(k)}$ and $\boldsymbol{N}^{(k)}_{,\boldsymbol{\xi}}$, together with the scaling factors $w_j$ and $s_j$ obtained from the map $\mathcal{W}$, are used to construct the enrichment functions $\psi^{(k)}, \chi^{(k)}$ and their derivatives $\boldsymbol{\psi}^{(k)}_{,\boldsymbol{x}}, \boldsymbol{\chi}^{(k)}_{,\boldsymbol{x}}$; these are then stacked to $\boldsymbol{F}$ and $\boldsymbol{F}_{,\boldsymbol{x}}$ arrays, respectively. Then, we obtain the parent element (either mesh element or another integration element) from the ordered tree $\mathcal{H}$ and the master coordinate at that element is obtained by an inverse mapping. The iteration continues until we reach the original mesh element (condition fulfilled by $k = 0$ in the algorithm).

Once all enrichment functions and their derivatives are computed, we add the contribution of the parent element through $\boldsymbol{N}^{(0)}$ and their derivatives $\boldsymbol{N}^{(0)}_{,\boldsymbol{\xi}}$, which are also stacked in their corresponding arrays. To account for multiple DOFs per node in elasticity, the arrays are expanded and stored in $\boldsymbol{N}$ and $\boldsymbol{B}$; in the end these are returned together with the saved jacobian $j$.

## 2.5. Numerical examples

In this section several numerical examples are provided to illustrate the performance of DE-FEM in 3-D. Unless explicitly stated, geometrical parameters, Young's moduli, and magnitudes of applied tractions are understood under any consistent unit system.

### 2.5.1. Discontinuous patch test

Following the reference [1], we first verify the DE-FEM formulation through a discontinuous patch test. Figure 2.13(a) shows a schematic, where a cube is split by a horizontal crack $\Gamma_c$ (the red plane) and tractions $\bar{\boldsymbol{t}} = t\boldsymbol{e}_1$ and $2\bar{\boldsymbol{t}}$ are applied to the front surface above and below the crack, respectively. The back surface is fixed and the top and bottom ones are constrained along the $\boldsymbol{e}_3$ direction as shown. An elastic modulus $E = 10$ and Poisson's ratio $\nu = 0$ are used. The original finite element discretization comprises 16 nodes and 42 linear tetrahedra (see Figure 2.13(b)); the new discretization after introducing the crack is also shown in Figure 2.13(c).

The resulting stress field is shown in Figure 2.13(d) on the deformed configuration. As apparent from the figure, DE-FEM passes this discontinuous patch test since there are two different constant stress states at either side of the crack. This means that DE-FEM is capable of generating two independent kinematic configurations in 3-D.

Since a non-conforming enriched mesh may be generated by the Delaunay algorithm during the creation of children integration elements. In order to display the effect of using a non-conforming mesh, consider two contiguous mesh elements, shown in Figure 2.14(a) in different colors. We modified the connectivity of their corresponding children elements below the crack by hand. Figure 2.14(b) shows that, although the stress distribution above the crack is still uniform, for the lower

part the stress is no longer constant. In this work we have chosen to use constrained Delaunay to avoid these situations altogether.



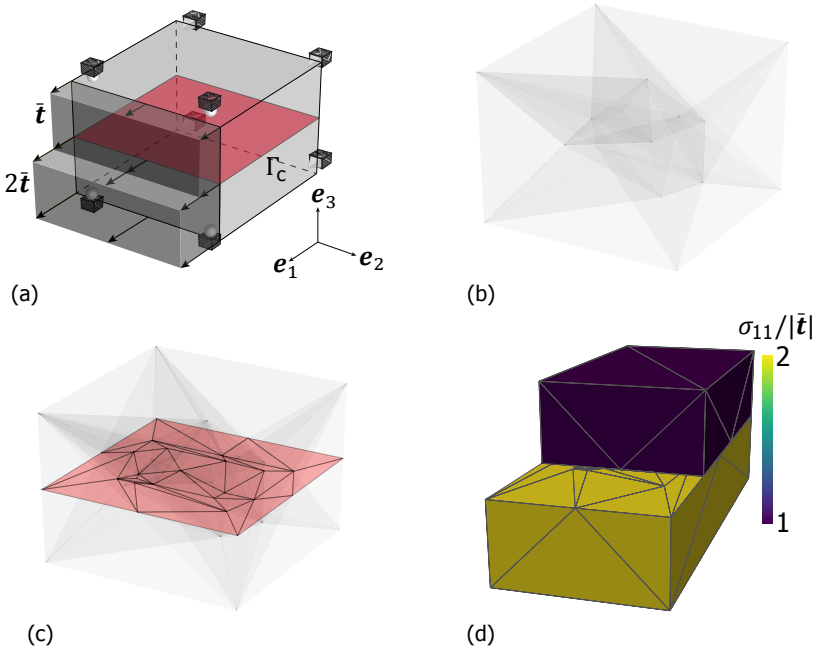



Figure 2.13: (a) The schematic of the discontinuous patch test where tractions $\bar{t}$ and $2\bar{t}$ are applied to the front surface above and below a crack $\Gamma_c$, respectively; (b) The original mesh composed by 42 linear tetrahedral elements; (c) New discretization after introducing the horizontal crack; (d) The normalized stress field on the deformed configuration, showing DE-FEM's capability of recovering independent kinematic fields.

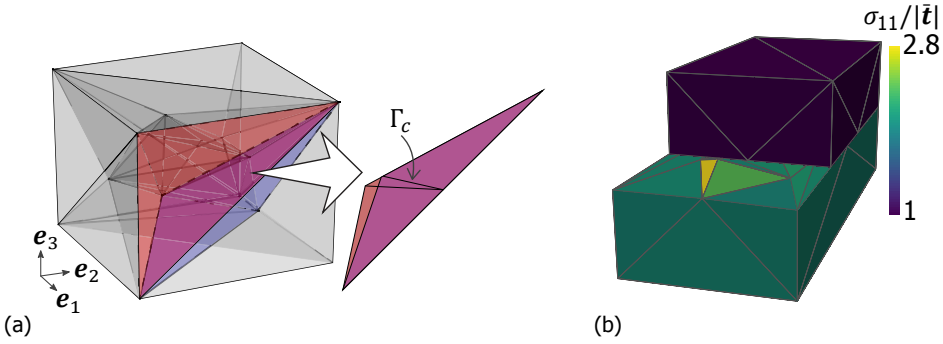



Figure 2.14: (a) For the two contiguous mesh elements shown in color, we force a non-conforming integration mesh by manually changing the connectivity of integration elements below the crack (see mismatch of triangles below $\Gamma_c$); (b) Uniform and non-uniform stress fields, above and below the crack, respectively. The latter is caused by the non-conforming mesh.

### 2.5.2. Convergence study

The convergence behavior of DE-FEM is investigated by comparing the convergence rates with those obtained by standard FEM on matching meshes for problems under modes I, II, and III. The computational domain used in this example consists of a cube discretized by linear tetrahedral elements. An elastic modulus $E = 10$ and Poisson's ratio $\nu = 0.3$ are used. The matching FE meshes used in standard FEM have 24, 384, 3072, 24 576, and 196 608 tetrahedra, as shown in Figures 2.15(a)-(e). Five different non-matching FE meshes, containing 6, 162, 2058, 20 250, and 178 746 elements as shown in Figures 2.15(f)-(j), are used in DE-FEM. The exact displacement field, given in Appendix A.1 for reference, is prescribed on the cube surface.

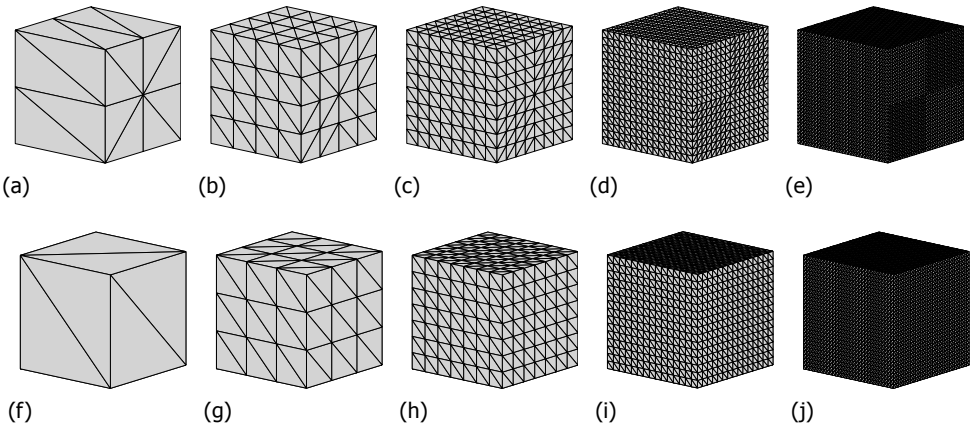



Figure 2.15: Matching meshes with (a) 24, (b) 384, (c) 3072, (d) 24 576, and (e) 196 608 linear tetrahedral elements used for standard FEM, and the non-matching meshes for DE-FEM with (f) 6, (g) 162, (h) 2058, (i) 20 250, and (j) 178 746 elements.

The error with respect to the exact solution is then measured in the energy norm, *i.e.*,

$$\frac{\left\|\boldsymbol{u}-\boldsymbol{u}^h\right\|_{E(\Omega^h)}}{\|\boldsymbol{u}\|_{E(\Omega^h)}}=\sqrt{\frac{\sum_{e\in\Omega^h}\int_e(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h)^{\intercal}\mathbf{C}(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h)\,\mathrm{d}\Omega}{\sum_{e\in\Omega^h}\int_e\boldsymbol{\varepsilon}^{\intercal}\mathbf{C}\boldsymbol{\varepsilon}\,\mathrm{d}\Omega}}, \tag{2.24}$$

where $\boldsymbol{u}$ is the analytical solution, $\boldsymbol{u}^h$ is the FE approximation by either DE-FEM or standard FEM, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}^h$ are their corresponding strains, and $\mathbf{C}$ is the elasticity tensor. Figure 2.16 shows deformed configurations for modes I, II, and III, obtained by standard FEM (a)-(c) and DE-FEM (d)-(f). For the latter, the deformation field is accurately visualized after post-processing [1].

Convergence results are summarized in Figure 2.17, where we see that convergence rates of DE-FEM are the same as those of standard FEM with the use of matching meshes. It is worth noting that convergence rates are not optimal for either method due to the lack of special techniques for handling the singularity along the crack front.

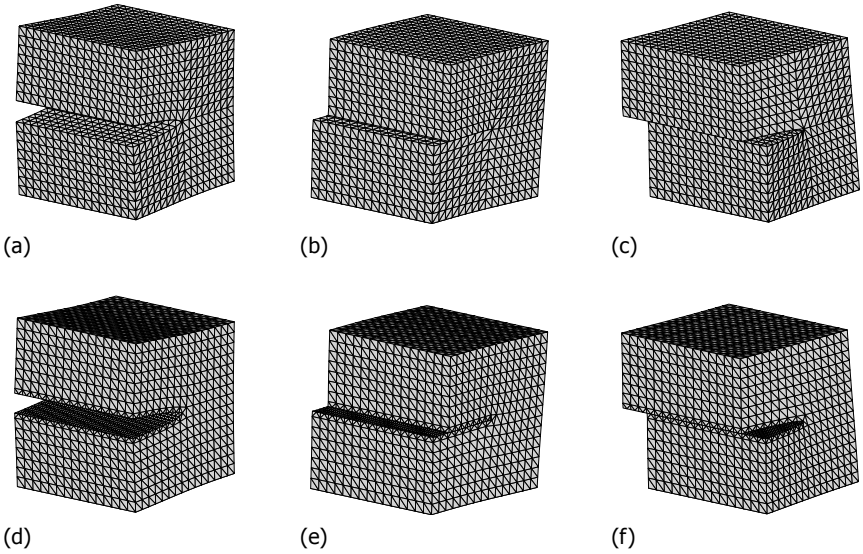



Figure 2.16: Deformed configurations for modes I, II, and III for Standard FEM (a)-(c) and DE-FEM (d)-(f). The latter are obtained after post-processing.

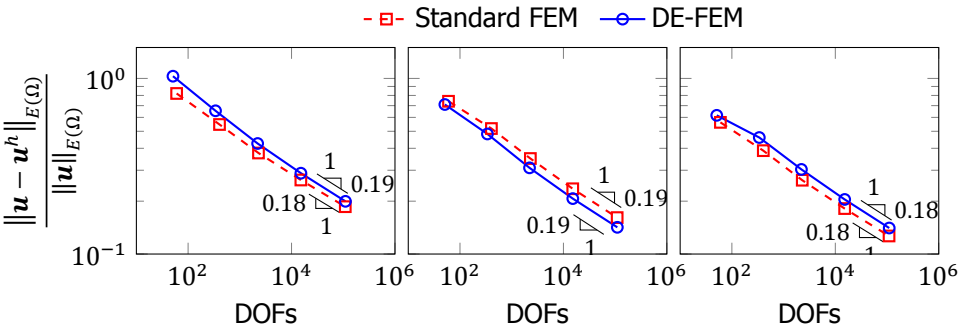



Figure 2.17: Error in energy norm given by Equation (2.24) as a function of the total number of DOFs for modes I (right), II (middle), and III (left). The figures show that both methods have the same convergence rates, although these are not optimal.

To investigate the influence of integration elements' aspect ratios, the element-wise error in the energy norm, given by

$$\left\|\boldsymbol{u}-\boldsymbol{u}^h\right\|_{E(e)} = \sqrt{\int_e \left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h\right)^{\intercal} \mathbf{C}\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h\right) \mathrm{d}e}, \tag{2.25}$$

is evaluated for different locations of the crack on the non-matching mesh shown in Figure 2.15(h). The crack, which moves from the middle of the background element

to the position close to background mesh nodes, cuts original hexahedral units (composed of six tetrahedra) and creates integration elements (see Figures 2.18(a) and (b)). The darkest integration elements shown in the figure are then considered for investigating the influence of aspect ratio $\Upsilon$, which we define following [41] as

$$\Upsilon = \frac{l_{\text{avg}}^3 / V}{8.48}, \tag{2.26}$$

where $l_{\text{avg}}$ is the average edge length, and $V$ is the tetrahedron volume. The aspect ratios considered range from 1.35 to 894.76. It should be mentioned that the aspect ratio of integration element is not always minimum when the crack cuts through the middle of the background mesh.

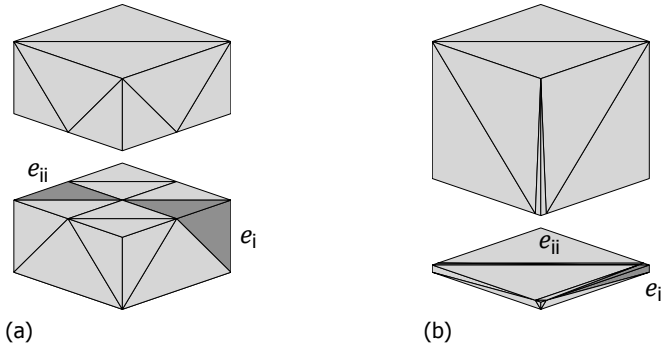



Figure 2.18: Integration elements created when crack cuts a hexahedral unit composed by six tetrahedra: (a) crack through the center; and (b) crack cutting close to background mesh nodes. The darkest integration elements are considered for investigating the influence of aspect ratio $\Upsilon$ in the local error.

Figure 2.19 shows the squared local error in the energy norm defined by Equation (2.25) for the two integration elements $e_{\text{i}}$ and $e_{\text{ii}}$ shown in Figures 2.18(a) and (b). The figure also shows the error measure normalized by the integration elements' volume. Because the local error involves integrating over element's volume, we normalize to single out the effect of aspect ratio. Indeed the solid curves show that increasing $\Upsilon$ also increases the local error, although the effect is more pronounced at low values of aspect ratio. The dashed curves show the combined volume/aspect ratio effect. The increase in local error caused by increasingly bad aspect ratios is counterbalanced by the element's volume, which decreases as the crack approaches the background mesh nodes (Figure 2.18(b)). For the element in the corner, which has the constant aspect ratio $\Upsilon = 1.24$, the local error without normalization shows the same tendency as the above two integration elements due to the decreasing element volume, and the normalized error ranges from $1 \times 10^{-5}$ to $1 \times 10^{-4}$. Consequently, bad aspect ratios in integration elements have not much influence in the global measure defined in Equation (2.24), and reducing the mesh size even further makes the contribution of the local error in integration elements even less significant.

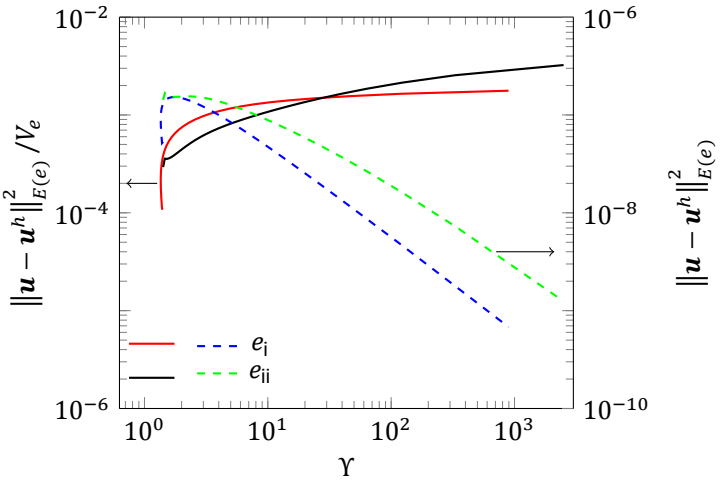



Figure 2.19: Squared local error in the energy norm defined by Equation (2.25) with and without normalizing by its volume, shown by solid and dashed curves, respectively. While the former curve shows the local error increases with increasing values of aspect ratio, the latter shows that the effect of high aspect ratios is counterbalanced by smaller volumes.

### 2.5.3. Stress intensity factors

We now study DE-FEM's capability to extract stress intensity factors (SIFs) for the single edge-crack tension problem of Figure 2.20(a), where for a crack width $a = 2$, we set $a/w = 0.5$, $l/w = 1.5$, and $h/w = 1.75$. Tractions $\bar{\boldsymbol{t}} = \pm\sigma_0 \boldsymbol{e}_3$ are applied at

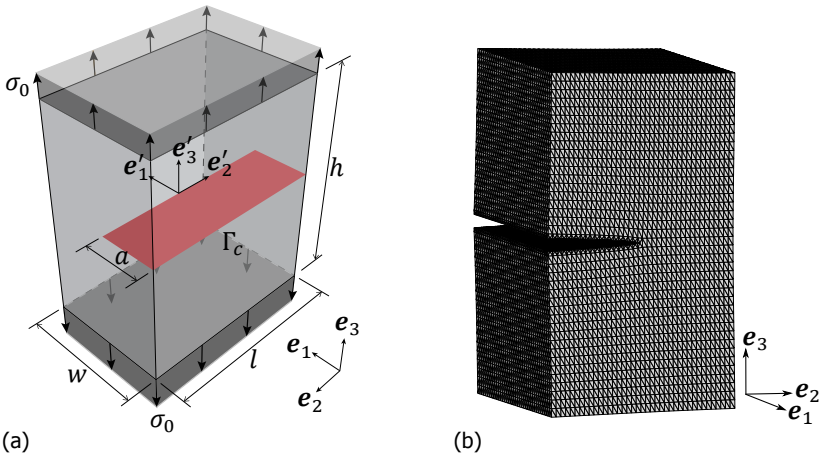



Figure 2.20: Single edge-crack tension example for extracting SIFs: (a) Problem schematic, where a cracked block is loaded by tractions $\bar{\boldsymbol{t}} = \pm\sigma_0 \boldsymbol{e}_3$ acting on top and bottom faces; (b) DE-FEM's deformed configuration.

top and bottom surfaces. A local Cartesian coordinate system $(\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3)$ is located at the center of the block, where $x'_2$ is measured along the crack front. The material properties are $E = 10$, and $\nu = 0.3$. The non-matching mesh used, shown in Figure 2.20(b), is defined over a $39 \times 39 \times 39$ Cartesian grid, resulting in 355 914 linear tetrahedra.

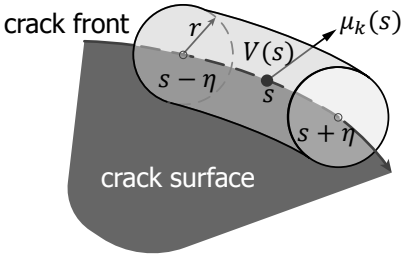



Figure 2.21: The integration domain $V(s)$ enclosing the crack front segment between $s-\eta$ and $s+\eta$ for calculating the interaction energy release rate $\bar{I}(s)$.

We use the domain integral method to calculate the energy release rate and the $M_1$ integral method to extract SIFs [42, 43]. For a point along the crack front, which is parameterized by the coordinate $s$ (see Figure 2.21), the local interaction energy release rate is given by

$$I(s) = \frac{1-\nu^2}{E}\left(2K_{\mathrm{I}}^{(1)}K_{\mathrm{I}}^{(2)} + 2K_{\mathrm{II}}^{(1)}K_{\mathrm{II}}^{(2)}\right) + \frac{1+\nu}{E}\left(2K_{\mathrm{III}}^{(1)}K_{\mathrm{III}}^{(2)}\right), \tag{2.27}$$

where superscripts (1) and (2) indicate the actual and auxiliary fields, respectively, and $K_i^{(1,2)}$, $i \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$ denote their corresponding SIFs. $I(s)$ can be extracted from the total interaction energy release rate $\bar{I}(s)$ under the assumption that it is constant within the integration domain $V(s)$ of length $2\eta$ that encloses a crack front segment. $I(s)$ and $\bar{I}(s)$ can be formulated, respectively, as

$$I(s) = \frac{\bar{I}(s)}{\int_{s-\eta}^{s+\eta} q_k(s')\mu_k(s')\,\mathrm{d}s'}, \tag{2.28}$$

and

$$\bar{I}(s) = \int_{V(s)} \left( (\sigma_{ij}^{(1)}\frac{\partial u_i^{(2)}}{\partial x_k} + \sigma_{ij}^{(2)}\frac{\partial u_i^{(1)}}{\partial x_k})\frac{\partial q_k}{\partial x_j} - \sigma_{ij}^{(1)}\varepsilon_{ij}^{(2)}\frac{\partial q_k}{\partial x_k} \right) \mathrm{d}V, \tag{2.29}$$

where $q_k$ is a continuous function with zero value outside $V(s)$, $\mu_k(s')$ are components of the unit vector generated from the junction of planes at point $s'$ which are perpendicular to the crack front and tangential to the crack plane. For the sake of obtaining the mixed-mode SIFs from Equations (2.27), (2.28), and (2.29), an appropriate auxiliary equilibrium state is chosen. For instance, if mode I is set as the auxiliary field, then $K_{\mathrm{I}}^{(2)} = 1, K_{\mathrm{II}}^{(2)} = K_{\mathrm{III}}^{(2)} = 0$, and the mode I SIF of the actual

field is $K_{\mathrm{I}}^{(1)} = EI(s)/(2(1-\nu^2))$. A similar strategy is used to obtain SIFs for modes II and III. For the integration domain to compute SIFs, we set $\eta = 12/39$ and $r = 16/39$.

Figure 2.22 shows the normalized SIF $K_{\mathrm{I}}/\sigma_0\sqrt{\pi a}$ is a function of the relative coordinate $x_2'/l$ along the crack front, and the corresponding values are given in Table 2.1. Under the plane strain condition, the analytical normalized SIF is 2.8284 [44]. The results obtained by DE-FEM show the same trend as other reference solutions; the SIF values approach the result obtained under the plane strain condition as $x_2'$ reaches the center. Moreover, in accordance with other references, DE-FEM also displays a decreasing $K_I$ as $x_2'$ reaches the boundary of the domain. For the study of Raju and Newman [45], singularity elements (where square-root terms are introduced to approximate the displacement field) were used to capture the singular stress field along the crack front. Generating the required mesh was, however, quite involved. For the study of Sukumar *et al.* [12], which used X/GFEM, special enrichment functions were used to capture the singular stress field. Yet, the required computer implementation is also more involved since, among other things, singular enrichments need special integration procedures [26]. There is always a trade-off between accuracy and complexity, and DE-FEM can obtain SIFs in a fully mesh-independent manner with a simpler computer implementation than that needed for X/GFEM, at the expense of some accuracy lost—1.4% below the value obtained by X/GEM at the center [12]. All these results were obtained on a coarse mesh, so for reference we also added SIFs computed by X/GFEM where the values were obtained using the mesh with local refinement along the crack front [46]. The SIF at the center, which is affected by the finite domain boundary, is higher than the one under the plane strain assumption.

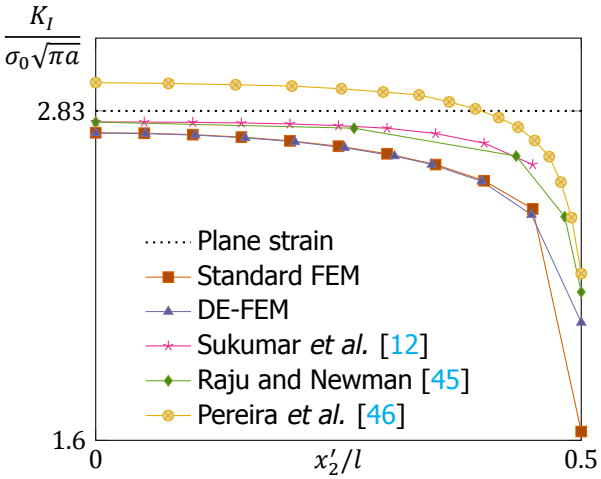



Figure 2.22: Normalized SIF values as a function of relative local coordinate $x_2'/l$. The results obtained by DE-FEM show the same trend as other reference solutions and accuracy that is on a par with that of standard FEM on matching meshes.

Table 2.1: The values of normalized SIF $K_{\mathrm{I}}/\sigma_0\sqrt{\pi a}$ obtained by DE-FEM.

| $x_2'/l$ | 0 | 0.051 | 0.103 | 0.154 | 0.205 | 0.256 | 0.307 | 0.346 | 0.397 | 0.448 | 0.500 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $K_{\mathrm{I}}/\sigma_0\sqrt{\pi a}$ | 2.747 | 2.744 | 2.738 | 2.728 | 2.713 | 2.691 | 2.660 | 2.628 | 2.566 | 2.441 | 2.039 |



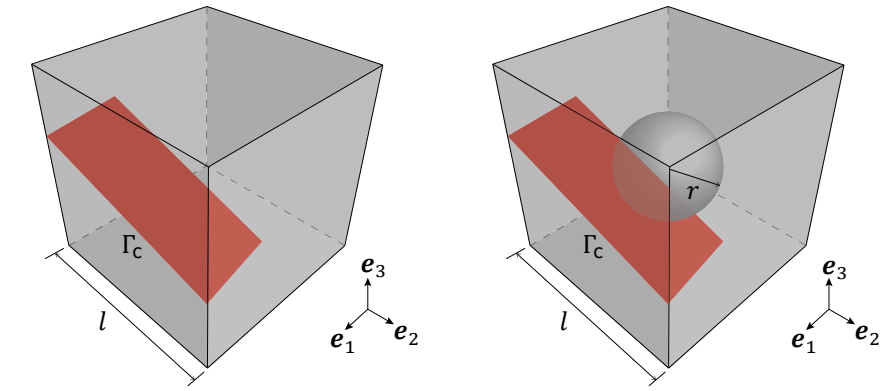
### 2.5.4. Stability

In this section we study how the condition number of the stiffness matrix grows with mesh refinement for both DE-FEM and standard FEM. As mentioned in the introduction, enriched methods like X/GFEM suffer from stability issues when discontinuities get arbitrarily close to nodes of the background mesh, and discontinuity-enriched methods like HIFEM/DE-FEM are not oblivious to this problem. For an integration element, the enriched components of the local stiffness matrix are inversely proportional the element's volume (in fact to the determinant of its Jacobian matrix) [40]. Therefore, vanishingly small integration elements quickly deteriorate the overall condition number of the system matrix. For weak discontinuities alone, *i.e.*, for IGFEM/HIFEM, an optimal function based on a 1-D analysis can be used to scale weak enrichments, resulting in a stable formulation [40]. However, it is also shown in that work that even if weak enrichments are not scaled, using a simple Jacobi-like preconditioner also results in a stable condition number. This is the approach we follow next.

As it is commonly done in SGFEM, we study the condition number of a modified stiffness matrix $\hat{\boldsymbol{K}} = \boldsymbol{DKD}$, where $D_{ij} = \delta_{ij}/\sqrt{K_{ij}}$ and thus $\hat{K}_{ii} = 1$. This modification is tantamount to applying a Jacobi-like preconditioner. The condition number $\varkappa(\hat{\boldsymbol{K}})$ is then obtained as the ratio between the maximum eigenvalue $\lambda_{\max}$ and the minimum (non-zero) eigenvalue $\lambda_{\min}$. We compare this condition number to that of the standard FEM component $\varkappa(\hat{\boldsymbol{K}}_{uu})$ (refer to Equation (2.10)), which grows with respect to mesh size $h$ as $\mathcal{O}\left(h^{-2}\right)$.

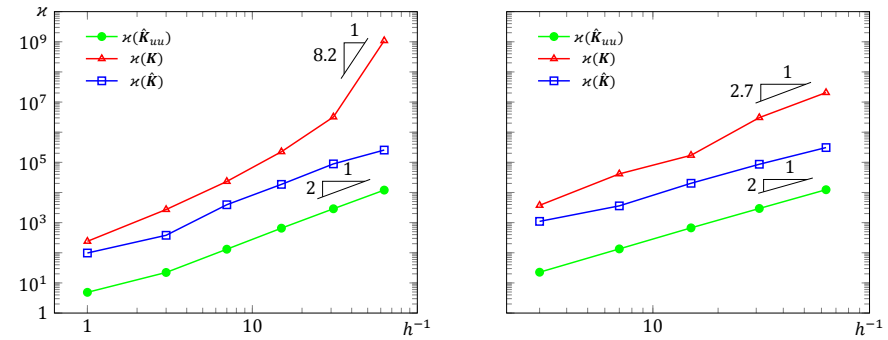
Two problems are used to study the stability. The first one consists of a solid cube containing a slanted crack with an orientation that favors the creation of arbitrarily shaped integration elements (see Figure 2.23(a)). The planar crack's normal is $\boldsymbol{n} = 0.1617\boldsymbol{e}_1 + 0.1819\boldsymbol{e}_2 + 0.9699\boldsymbol{e}_3$. For the second example, we introduce an extra spherical material interface with radius $r = 0.59$ that intersects the planar crack (see Figure 2.23(b)). The structured background meshes considered here are the same as those used in the convergence study.

The results of both cases are reported in Figure 2.24, where the condition number is shown as a function of the reciprocal of the mesh size. Our reference curve is that of the standard FEM component $\varkappa(\hat{\boldsymbol{K}}_{uu})$, which scales as $\mathcal{O}\left(h^{-2}\right)$. The figure shows that the condition number of the raw stiffness matrix $\varkappa(\boldsymbol{K})$, without the use of a preconditioner, is much worse. Using the preconditioner results in the curve $\varkappa(\hat{\boldsymbol{K}})$, which has about an order of magnitude higher condition number than the standard part. However, the rate of growth is roughly the same, and therefore DE-FEM is stable if a simple preconditioner is applied.




(a) (b)

Figure 2.23: Problem schematics for studying DE-FEM's stability: (a) cube of side length $l = 2$ partially cut by a planar crack; (b) spherical material interface with radius $r = 0.59$ intersects the planar crack.




(a) (b)

Figure 2.24: Condition number as a function of $h^{-1}$ for (a) the planar crack shown in Figure 2.23(a); and (b) the intersecting discontinuities shown in Figure 2.23(b) shows that, while the condition number of the raw system matrix $\varkappa(\boldsymbol{K})$ is unstable, applying a simple preconditioner results in $\varkappa(\hat{\boldsymbol{K}})$, which grows roughly at the same rate as that of the standard FEM component $\varkappa(\hat{\boldsymbol{K}}_{uu})$.

### 2.5.5. Bone fracture

We now use DE-FEM to simulate the fracture mechanics of a femur whose geometry was obtained from [47], as shown in Figure 2.25(a). The femur is subjected to a prescribed displacement field $\bar{\boldsymbol{u}} = \pm 1\,\text{mm}\,\boldsymbol{e}_1$ at its ends and a crack $\Gamma_c$ that partially splits the bone is introduced.

Bone can be considered a composite material with cancellous (inner) and cortical (outer) phases with completely different material properties [48]. For simplicity, here we consider them as isotropic materials with elastic constants $E = 10.4\,\text{GPa}$, $\nu = 0.12$ (cancellous bone) and $E = 18.6\,\text{GPa}$, $\nu = 0.3$ (cortical bone) [49, 50]. In

addition, a level set function is introduced to describe their interface. Figure 2.25(b) shows the finite element mesh used, composed of 7355 tetrahedral elements. Although the mesh represents the femur external boundary accurately, the mesh is non-conforming not only to the interface between materials, but also to the crack $\Gamma_c$.

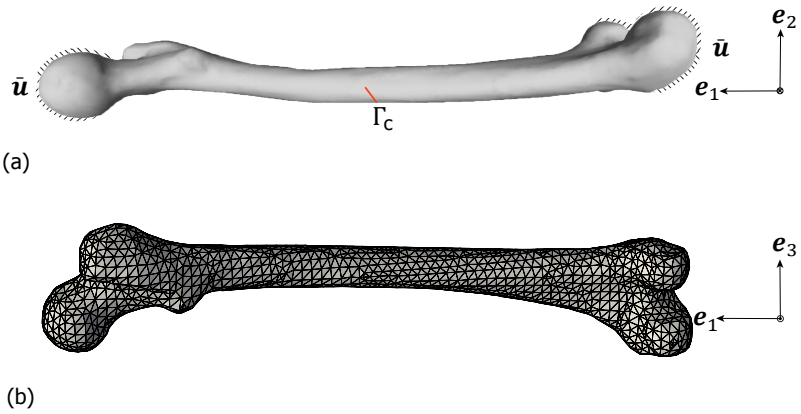



Figure 2.25: (a) The essential boundary condition prescribed on the femur model with the crack $\Gamma_c$; (b) The mesh used matches the femur's external boundary but not the material interface nor the crack.

The results of this simulation are shown in Figure 2.26, where we see that new children (integration) elements are created properly at each side of the interface (Figure 2.26(a)). Figure 2.26(b) displays the resulting von Mises stress field, highlighting the stress concentration near the crack front and the higher stress in cortical bone. This example showcases DE-FEM's versatility not only in modeling both weak and strong discontinuities with a unified formulation, but also in resolving them even if they intersect the same element. This is possible due to the hierarchical implementation discussed in the previous section, which allows for the splitting of elements by several discontinuities seamlessly.

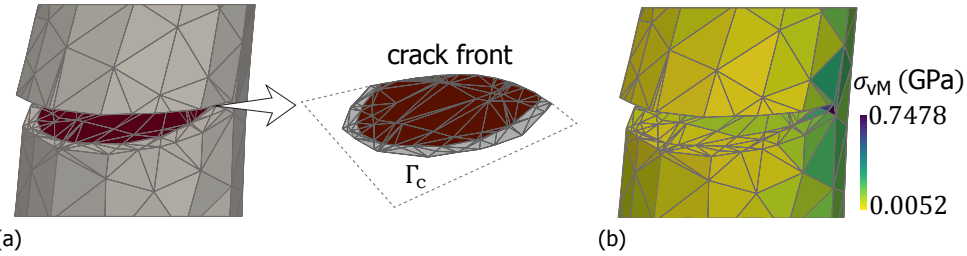



Figure 2.26: The discretized model near the crack extracted from the whole bone where the background mesh is not matching to the interface nor to the crack: (a) The red and grey parts model cancellous and cortical bone, respectively. The resulting integration mesh is matching to both the interface and the crack; (b) von Mises stress field highlighting the higher stress in cortical bone.

## 2.6. Summary and conclusions

In this chapter we introduced DE-FEM for solving problems with both weak and strong discontinuities in 3-D. We discussed our implementation of a geometric engine to handle complex 3-D configurations. By means of constrained Delaunay, the geometric engine handles all possible scenarios that arise when splitting tetrahedral elements with complex interfaces and/or cracks. In addition, a hierarchical data structure was discussed for resolving an arbitrary number of discontinuities within a single mesh element. This data structure is then used construct hierarchically weak and strong enrichment functions.

The *integration mesh* that is created in our DE-FEM implementation serves three purposes. Firstly, the tetrahedra that result from subdividing parent mesh elements are used to construct the enriched approximation. Indeed the enrichment functions are built with the aid of Lagrange shape functions of integration subdomains. Secondly, by using tetrahedra as subdomains we guarantee robustness when handling multiple discontinuities since tetrahedra are the simplest 3-D volumes to split. Lastly, creating integration tetrahedra with sides aligned to discontinuities ensures that the discontinuous functions within these elements are smooth. Other integration schemes that do not require subdividing into integration elements, but which can otherwise recursively perform the numerical quadrature, are also possible for weak discontinuities. However, because of the non-smooth nature of the functions, the computational cost is higher because several levels of recursion are needed along discontinuities [51].

Through a discontinuous patch test we showed that the DE-FEM formulation is able to generate two independent kinematic fields. We also showed that DE-FEM converges at the same rate as the standard FEM with matching meshes and that DE-FEM can be used to extract stress intensity factors. Finally, we simulated the challenging problem of resolving the interplay between weak and strong discontinuities within a complex-shaped femur model.

In Table 2.2 we compare DE-FEM to both standard FEM and to X/GFEM. We see that DE-FEM also decouples the underlying mesh from discontinuities (mesh independency), which is X/GFEM's most attractive feature. However, contrary to X/GFEM, DE-FEM also retains the attractive properties of standard FEM. Because enrichment functions are local to cut elements by construction, there is no need to use the mesh's partition of unity shape functions of the mesh to localize enrichments. Also, since enrichments are exactly zero at nodes of the background mesh, the DOFs associated to these nodes keep their physical meaning. Another consequence of enrichments vanishing at mesh nodes is that it is straightforward to impose Dirichlet boundary conditions. Non-homogeneous essential boundary conditions can be prescribed strongly also at enrichment node locations through a simple multiple point constraint. In fact, there is ongoing work on the use of DE-FEM as an immerse boundary or fictitious domain technique with strong enforcement of Dirichlet conditions [27]. Moreover, the requirements for the quality of the integration mesh are less strict than that of a proper *matching* mesh. Regarding stability, we showed that the condition number of the whole stiffness matrix grows at roughly

the same rate as that of standard FEM. Therefore DE-FEM is stable with the use of a simple preconditioner.

Table 2.2: Characteristics of Standard FEM, X/GFEM, and DE-FEM

| | **Standard FEM** | **X/GFEM** | **DE-FEM** |
|---|---|---|---|
| **Mesh independency** | × | √ | √ |
| **PoU for localizing enrichments** | Not applicable | √ | Local by construction |
| **Physical meaning of standard DOFs** | √ | Shifting required | √ |
| **Non-zero essential boundary conditions** | Direct | Penalty, Lagrange multipliers, *etc.* | Direct |
| **Condition number growth (stability)** | $\mathcal{O}\left(h^{-2}\right)$ | $\mathcal{O}\left(h^{-2}\right)$ with SGFEM [16] | $\mathcal{O}\left(h^{-2}\right)$ |
| **Numerical integration** | Standard | Integration mesh in cut elements | *Conforming* integration mesh |
| **Optimal convergence** | × | √ | TBD |

The simple formulation in DE-FEM leads to a simpler computer implementation when compared to X/GFEM. The current formulation cannot get optimal convergence rates for problems with singularities because the current formulation is devoid of singular enrichments. These could be added if higher accuracy is required—*e.g.*, for extracting stress intensity factors. Nevertheless, adding singular enrichments to the formulation would complicate the computer implementation, not only because of the need for bookkeeping new enriched DOFs (and variable DOFs per mesh node), but also because singular enrichments need to be integrated accurately, for which many more quadrature points are required. Adding DOFs also leads to increasing the cost of solving the linear system of equations. Although we did not use any enrichments to capture singularities, the predictions of SIFs are not far from other more accurate values for the level of mesh size $h$; our predicted result at the center was just 1.4% below the value obtained by X/GFEM.

The geometric engine is the most intricate part of the implementation because of the complexity of computational geometry operations involved and the need for robustness with respect to the placement of discontinuities. This is the crux not only of DE-FEM and other interface-enriched formulations such as IGFEM or HIFEM, but also for X/GFEM when dealing with discontinuities. For DE-FEM we showed the im-

portance of creating a conforming enriched discretization for integration. Although the geometric engine described in this work suffices for the types of problems described, there is wide room for improvement and this will be the subject of subsequent work.



## References


[1] A. M. Aragón and A. Simone, *The discontinuity-enriched finite element method,* International Journal for Numerical Methods in Engineering **112**, 1589 (2017).

[2] D. M. Tracey, *Finite elements for determination of crack tip elastic stress intensity factors,* Engineering Fracture Mechanics **3**, 255 (1971).

[3] R. S. Barsoum, *On the use of isoparametric finite elements in linear fracture mechanics,* International Journal for Numerical Methods in Engineering **10**, 25 (1976).

[4] J. R. Shewchuk, *What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures*, Tech. Rep. (In Proc. of the 11th International Meshing Roundtable, 2002).

[5] Q. Du and D. Wang, *Recent progress in robust and quality delaunay mesh generation,* Journal of Computational and Applied Mathematics **195**, 8 (2006).

[6] R. Rangarajan and A. J. Lew, *Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes,* International Journal for Numerical Methods in Engineering **98**, 236 (2014).

[7] S. Soghrati, F. Xiao, and A. Nagarajan, *A conforming to interface structured adaptive mesh refinement technique for modeling fracture problems,* Computational Mechanics **59**, 667 (2017).

[8] J. T. Oden, C. A. Duarte, and O. C. Zienkiewicz, *A new cloud-based hp finite element method,* Computer Methods in Applied Mechanics and Engineering **153**, 117 (1998).

[9] N. Moës, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing,* International Journal for Numerical Methods in Engineering **46**, 131 (1999).

[10] T. P. Fries and T. Belytschko, *The extended/generalized finite element method: An overview of the method and its applications,* International Journal for Numerical Methods in Engineering **84**, 253.

[11] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydlo, *A generalized finite element method for the simulation of three-dimensional dynamic crack propagation,* Computer Methods in Applied Mechanics and Engineering **190**, 2227 (2001).

[12] N. Sukumar, N. Moës, T. Belytschko, and B. Moran, *Extended finite element method for three-dimensional crack modelling,* International Journal for Numerical Methods in Engineering **48**, 1549 (2000).

[13] T. Belytschko, N. Moës, S. Usui, and C. Parimi, *Arbitrary discontinuities in finite elements,* International Journal for Numerical Methods in Engineering **50**, 993 (2001).

[14] I. Babuška, U. Banerjee, and J. E. Osborn, *Survey of meshless and generalized finite element methods: A unified approach,* Acta Numerica **12**, 1 (2003).

[15] N. Moës, E. Béchet, and M. Tourbier, *Imposing Dirichlet boundary conditions in the extended finite element method,* International Journal for Numerical Methods in Engineering **67**, 1641 (2006).

[16] I. Babuška and U. Banerjee, *Stable generalized finite element method (SGFEM),* Computer Methods in Applied Mechanics and Engineering **201-204**, 91 (2012).

[17] K. Kergrene, I. Babuška, and U. Banerjee, *Stable generalized finite element method and associated iterative schemes; application to interface problems,* Computer Methods in Applied Mechanics and Engineering **305**, 1 (2016).

[18] I. Babuška, U. Banerjee, and K. Kergrene, *Strongly stable generalized finite element method: Application to interface problems,* Computer Methods in Applied Mechanics and Engineering **327**, 58 (2017).

[19] V. Gupta, C. Duarte, I. Babuška, and U. Banerjee, *A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **266**, 23 (2013).

[20] V. Gupta, C. Duarte, I. Babuška, and U. Banerjee, *Stable GFEM (SGFEM): Improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **289**, 355 (2015).

[21] Q. Zhang, I. Babuška, and U. Banerjee, *Robustness in stable generalized finite element methods (SGFEM) applied to poisson problems with crack singularities,* Computer Methods in Applied Mechanics and Engineering **311**, 476 (2016).

[22] J. Chessa, H. Wang, and T. Belytschko, *On the construction of blending elements for local partition of unity enriched finite elements,* International Journal for Numerical Methods in Engineering **57**, 1015 (2003).

[23] T. P. Fries, *A corrected XFEM approximation without problems in blending elements,* International Journal for Numerical Methods in Engineering **75**, 503 (2008).

[24] G. Ventura, R. Gracie, and T. Belytschko, *Fast integration and weight function blending in the extended finite element method,* International journal for numerical methods in engineering **77**, 1 (2009).

[25] N. Sukumar and J. H. Prévost, *Modeling quasi-static crack growth with the extended finite element method part i: Computer implementation,* International Journal of Solids and Structures **40**, 7513 (2003).

[26] K. Park, J. P. Pereira, C. A. Duarte, and G. H. Paulino, *Integration of singular enrichment functions in the generalized/extended finite element method for three-dimensional problems,* International Journal for Numerical Methods in Engineering **78**, 1220 (2008).

[27] S. J. van den Boom, J. Zhang, F. van Keulen, and A. A. M., *A new stable interface-enriched formulation for immersed domains,* International Journal for Numerical Methods in Engineering (Submitted).

[28] S. Soghrati, A. M. Aragón, C. A. Duarte, and P. H. Geubelle, *An interface-enriched generalized fem for problems with discontinuous gradient fields,* International Journal for Numerical Methods in Engineering **89**, 991 (2012).

[29] S. Soghrati, *Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations,* Journal of Computational Physics **275**, 41 (2014).

[30] S. Soghrati and P. H. Geubelle, *A 3D interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries,* Computer Methods in Applied Mechanics and Engineering **217**, 46 (2012).

[31] A. M. Aragón, S. Soghrati, and P. H. Geubelle, *Effect of in-plane deformation on the cohesive failure of heterogeneous adhesives,* Journal of Mechanics and Physics of Solids **61**, 1600 (2013).

[32] S. Soghrati, A. R. Najafi, J. H. Lin, K. M. Hughes, S. R. White, N. R. Sottos, and P. H. Geubelle, *Computational analysis of actively-cooled 3D woven microvascular composites using a stabilized interface-enriched generalized finite element method,* International Journal of Heat and Mass Transfer **65**, 153 (2013).

[33] A. C. Ramos, A. M. Aragón, S. Soghrati, P. H. Geubelle, and J. Molinari, *A new formulation for imposing Dirichlet boundary conditions on non-matching meshes,* International Journal for Numerical Methods in Engineering **103**, 430 (2015).

[34] S. Soghrati and H. Ahmadian, *3d hierarchical interface-enriched finite element method: Implementation and applications,* Journal of Computational Physics **299**, 45 (2015).

[35] M. Safdari, A. R. Najafi, N. R. Sottos, and P. H. Geubelle, *A nurbs-based generalized finite element scheme for 3D simulation of heterogeneous materials,* Journal of Computational Physics **318**, 373 (2016).

[36] S. Soghrati and R. A. Merel, *NURBS enhanced HIFEM: A fully mesh-independent method with zero geometric discretization error,* Finite Elements in Analysis and Design **120**, 68 (2016).

[37] A. M. Aragón and J.-F. Molinari, *A hierarchical detection framework for computational contact mechanics,* Computer Methods in Applied Mechanics and Engineering **268**, 574 (2014).

[38] A. Salzman, N. Moës, and N. Chevaugeon, *On use of the thick level set method in 3D quasi-static crack simulation of quasi-brittle material,* International Journal of Fracture **202**, 21 (2016).

[39] H. Si, *Tetgen, a Delaunay-based quality tetrahedral mesh generator,* ACM Transactions on Mathematical Software (TOMS) **41**, 11:1 (2015).

[40] A. M. Aragón, B. Liang, H. Ahmadian, and S. Soghrati, *The hierarchical interface-enriched finite element method: Stability and implementation,* International Journal for Numerical Methods in Engineering, Submitted (2018).

[41] Q. Zhang and S. Cen, *Multiphysics Modeling: Numerical Methods and Engineering Applications: Tsinghua University Press Computational Mechanics Series* (Elsevier, 2015).

[42] T. Nakamura and D. M. Parks, *Antisymmetrical 3-d stress field near the crack front of a thin elastic plate,* International Journal of Solids and Structures **25**, 1411 (1989).

[43] M. Nagai, T. Ikeda, and N. Miyazaki, *Stress intensity factor analyses of three-dimensional interface cracks using tetrahedral finite elements,* Computational Mechanics **51**, 603 (2013).

[44] H. Tada, P. Paris, and G. Irwin, *The Stress Analysis of Cracks Handbook* (Wiley, 2000).

[45] I. S. Raju and J. C. Newman Jr, *Three dimensional finite-element analysis of finite-thickness fracture specimens*, Tech. Rep. (NASA Langley Research Center; Hampton, VA, United States, 1977).

[46] J. P. Pereira, C. A. Duarte, D. Guoy, and X. Jiao, *hp-generalized fem and crack surface representation for non-planar 3-D cracks,* International Journal for Numerical Methods in Engineering **77**, 601 (2009).

[47] *Bodyparts3d/anatomography,* http://lifesciencedb.jp/bp3d/.

[48] M. E. Launey, M. J. Buehler, and R. O. Ritchie, *On the mechanistic origins of toughness in bone,* Annual Review of Materials Research **40**, 25 (2010).

[49] Y. J. Rho, B. R. Ashman, and H. C. Turner, *Young's modulus of trabecular and cortical bone material: Ultrasonic and microtensile measurements,* Journal of Biomechanics **26**, 111 (1993).

[50] D. C. Wirtz, N. Schiffers, T. Pandorf, K. Radermacher, D. Weichert, and R. Forst, *Critical evaluation of known bone material properties to realize anisotropic FE-simulation of the proximal femur,* Journal of Biomechanics **33**, 1325 (2000).

[51] A. M. Aragón, C. A. Duarte, and P. H. Geubelle, *Generalized finite element enrichment functions for discontinuous gradient fields,* International Journal for Numerical Methods in Engineering **82**, 242 (2010).

# 3

# An Object-oriented Geometric Engine Design for Handling Discontinuities in Unfitted/Immersed/Enriched Methods

*In this chapter, an object-oriented geometric engine is designed for dealing with discontinuities, such as material interfaces and cracks. Compared with the one proposed in the last chapter, the new geometric engine is more general, efficient, and robust. Both explicit and implicit methods, such as geometric entities and level set, are introduced to describe configurations of discontinuities. In order to make the geometric engine efficient, a k-d tree data structure that partitions the background mesh is constructed for detecting cut elements whose neighbors are found by means of a dual graph structure. Moreover, the implementation for creating enriched nodes, integration elements, and physical groups is described in detail, and the corresponding pseudo-code is also provided. The complexity and efficiency of the geometric engine are investigated by solving 2-D and 3-D discontinuous models. Later, numerical examples are proposed to demonstrate the capability of the geometric engine. Topology optimization and intersecting discontinuities problems are handled with enriched finite element methods, where enriched discretizations obtained from the geometric engine are used for the analysis. Furthermore, polycrystalline structures are overlapped with a unfitted mesh, where integration elements created align with the grain boundaries. Finally, we show that the Standford Bunny model discretized by a surface mesh with triangular elements is fully embedded into a 3-D background mesh.*

This chapter has been submitted to *International Journal for Numerical Methods in Engineering.*

## 3.1. Introduction

Commercial softwares such as Abaqus, Ansys, and COMSOL, have become the industry standard for finite element (FE) numerical analysis. They require, however, FE meshes that fit the problems' geometries, *i.e.*, where the edges of finite elements align with the boundary and other discontinuities such as material interfaces and/or cracks. This is a strict requirement: The creation of *fitted* or *geometry-conforming* meshes for complex geometries has been estimated to take about 80% of the total analysis time [1]. Furthermore, creating a good-quality mesh is a challenging endeavour that is prone to failure, especially for 3-D models [2]. Creating fitted meshes is troublesome and/or time-consuming, for instance, for fracture [3–5], structural optimization [6–8], fluid-structure interaction [9, 10], fibre/particle-reinforced composites [11], and structures with irregular boundaries [12, 13]. For these problems, transferring the complexity of mesh generation to the finite element formulation has been shown advantageous. Unfitted/immersed/enriched FEMs allow for a complete decoupling between the problem geometry and the discontinuities. A shared aspect among unfitted/immersed/enriched FEMs that use (usually structured) unfitted discretizations is that interactions between a background mesh and the problem's geometry have to be performed. This work presents a general object-oriented *geometric engine* that can be used for mesh interactions with both implicit and explicit representations of discontinuities in all dimensions.

Even though the proposed work is applicable for unfitted/immersed/enriched FEMs, we place emphasis on the latter family of methods, for which the geometric engine was originally designed. Enriched FEMs have been vastly explored to treat discontinuities such as cracks and material interfaces. In the eXtended/Generalized Finite Element Method (X/GFEM) [14, 15], which originates from the partition of unity method [16, 17], the standard finite element approximation is augmented by enrichment functions that reproduce a discontinuity in the primal field (*e.g.* for cracks) or its gradient (*e.g.* for material interfaces). However, although X/GFEM provides great flexibility in solving discontinuous problems with a *fixed* background mesh, the method is not without issues, including loss of accuracy in blending elements for certain choice of enrichment function [18, 19], the need for special formulations for prescribing non-homogenous essential (Dirichlet) boundary conditions (BCs) [20, 21], and lack of stability—*i.e.*, the condition number of the stiffness matrix may grow much faster than that of standard FEM with fitted discretizations [22–24]. There are, however, less known enriched finite element formulations that, while retaining the mesh-geometry decoupling, are devoid of the aforementioned issues. The Interface-enriched Generalized Finite Element Method (IGFEM) [25], for instance, is a simpler enriched formulation that can be derived from X/GFEM [26]. IGFEM adds enriched degrees of freedom (DOFs) to nodes created along discontinuities—instead of associating them to nodes of the original FE mesh as done in X/GFEM. IGFEM recovers the standard finite element space and thus it can be seen as a method between X/GFEM and FEM. IGFEM is intrinsically stable by a proper scaling of enrichment functions or the use of a simple diagonal preconditioner [27, 28], and nonzero essential BCs can be prescribed strongly [29]. Furthermore, multiple discontinuities within a single finite element

can be handled via a hierarchical implementation [28]. The method can seamlessly resolve boundaries so it can be used to solve immersed boundary (fictitious domain) problems—and with smooth reactive tractions in Dirichlet boundaries [29, 30]. In addition, IGFEM has been used to prescribe Bloch-Floquet periodic boundary conditions in the analysis of phononic crystals [31], and for coupling non-conforming meshes and highly-nonlinear contact problems [32]. In the context of topology optimization, IGFEM has been used for compliance minimization [26], and for tailoring the fracture resistance in brittle structures [33]. Finally, IGFEM has also been generalized for the unified treatment of both weak and strong discontinuities in the Discontinuity Enriched Finite Element Method (DE-FEM) [34–36]. These enriched FEMs have been shown to outperform standard FEM for solving weak and/or strong discontinuous problems, including



- **Multi-phase materials** Inclusions embedded in a matrix material with distinct properties (see Figure 3.1(a)) are prevalent in natural materials such as bone [37] and in artificial materials such as fibre-reinforced composites [38]. Multi-phase materials can also exhibit a microstructure where there is no apparent matrix phase, as is the case with polycrystalline microstructures (see Figure 3.1(b)). These materials, which are assemblies of small grains (crystals) with different material properties, include polycrystalline ceramics, metals, and alloys. Because of their common use in industrial applications, multi-phase materials require accurate numerical characterization. Material interfaces, which may have a critical effect on the macroscopic material behavior, have been modeled by assuming perfect bonding between interfaces ($C^0$–continuity) [39–41] and progressive degradation of the interface by means of cohesive models ($C^{-1}$–continuity) [42, 43].

- **Solidification** Boundaries between phases can evolve as a consequence of temperature changes, *e.g.*, liquid to solid (the molten metal back into the solid state during metal casting). This solidification phenomenon plays an important role in material processing in many industries because it could lead to important variations in material properties such as strength, thermal stability, and ductility [44, 45]. It is therefore important to understand this phase transformation process so it can be precisely controlled [46]. Due to the complex nature of the problem, where the location of phase interfaces varies with time, only a limited number of analytical solutions are available [47]. Numerical techniques are therefore widely used to model the discontinuous temperature gradient field at phase interfaces [48–52].

- **Two-phase flows** A flow consisting of different phases (see Figure 3.1(c)) is commonplace in the industrial applications, such as oil and gas in pipelines, and water and steam in nuclear reactor cooling systems [53]. Since these phases have different properties, the interface between them can result in discontinuous velocity/pressure fields (under a slip condition) and their corresponding gradient fields. This characteristic makes very challenging to solve two-phase flow problems in an analytical way [54, 55]. Although most two-phase flow formulations obtained are based on experimental data [56], nu-

merical simulation (based on enriched FEMs) also provides a viable alternative to describe the flow behavior [57–60]. Two important aspects should be carefully considered for obtaining an accurate approximation: First, a robust representation is necessary to describe the geometric configuration of interfaces, which could be altered significantly during the analysis. More importantly, appropriate enrichment functions should be constructed to account for the discontinuities in the primal/gradient fields along phase interfaces.



- **Fluid-structure interaction** Unlike two-phase flows, fluid-structure interaction (FSI) is a problem that couples fluid dynamics and solid mechanics (loosely referred as multiphysics) [61]. There are multiple applications of FSI, such as dams [62], aircrafts [63], and even the cardiovascular system [64]. For solving the FSI problem, capturing accurately the interaction between fluid and solid, *i.e.*, both the fluid velocity and pressure at fluid-solid interfaces and the structure deformation, is a challenging endeavour for which enriched FEMs have proven to be effective [65–67].

- **Fracture** Cracks (see Figure 3.1(d)), which could nucleate along the boundary or inside the structure, may have a detrimental effect on the integrity of a structure and could even result in its failure [68, 69]. Therefore, it is of utmost important to properly characterize their effect, for which a plethora of works can be found on enriched FEMs alone. Cracks are usually modeled as strong discontinuities that introduce the kinematics of a discontinuous primal displacement field [15, 34]. Enriched FEM has been used to obtain stress intensity factors (SIFs) in stationary cracks [35, 70], and to model crack propagation by means of cohesive zone models [71, 72] or by comparing the stress intensity to critical values [14, 15].

- **Dislocations** These are linear defects that generally occur in crystalline materials, *e.g.*, metals, diamonds, rocks, or ceramics [73]. Dislocations could result in phase transformations and grain growth, and even change material properties [74]. The geometrical configuration of a dislocation is represented by a glide plane with unit normal vector $\boldsymbol{n}$ (see Figure 3.1(e)), where either side of the glide surface shears with respect to another. Unlike cracks, the bare presence of dislocations can result in stresses in the material, even without applying external tractions [75]. Similarly to cracks, dislocations are treated as strong discontinuities, and the jump in the displacement field is described by the Burgers vector $\boldsymbol{b}$ that is tangent to the glide plane [73]. Therefore, similar considerations to fracture can be made when modeling dislocations [76, 77].

- **Shear bands** These are narrow zones with intense localized shearing (see Figure 3.1(f)) that are usually observed in soil [78], rock [79], ceramics [80], and metal [81]. Created by strain localization, shear bands could lead to extreme deformations, instabilities, and even failure [82]. The influence of shear hands on material behavior has been widely investigated via theoretical and experimental studies [79, 83]. Within the realm of enriched FEMs, a shear band can be modeled as a strong discontinuity when the width of the band

relative to the medium is small [84, 85], or as two weak discontinuities otherwise [86, 87].

A common feature on the use of enriched FEMs to solve all aforementioned discontinuous problems is that a simple (usually structured) finite element mesh is used. Yet, this flexibility does not come for free; usually, enrichment functions need to be created, which relies on the known location of discontinuities. In addition, the numerical quadrature of these discontinuous enrichment functions usually requires intersection tests between discontinuities and mesh elements, and the further subdivision of the latter into so-called *integration elements*. For some problems it is also required to keep track of the evolution of such discontinuities. In enriched FEMs all this functionality is handled by a *geometric engine*.



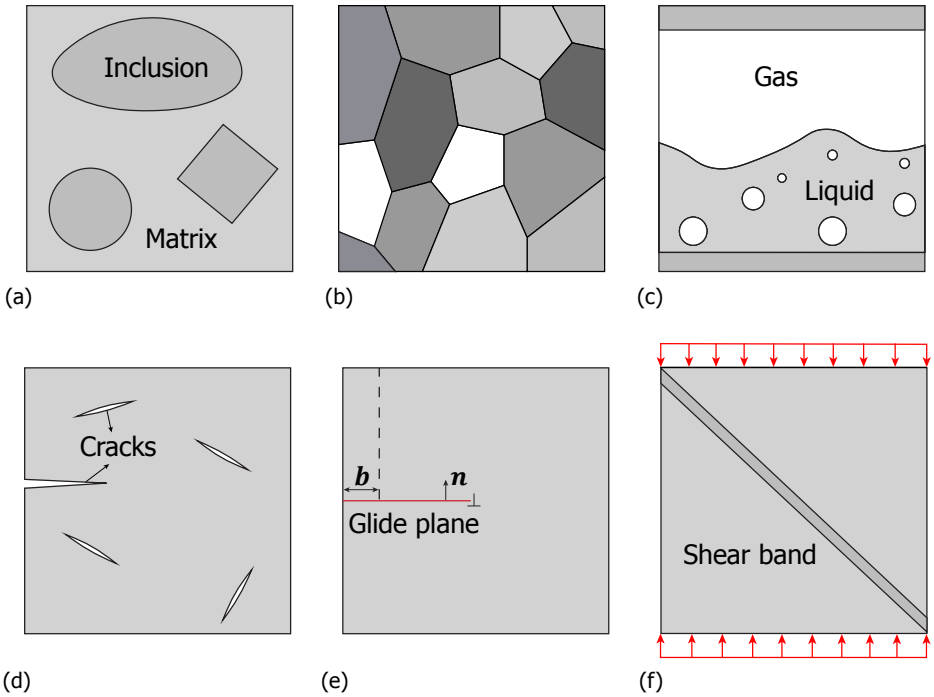



Figure 3.1: (a) Material inclusions embedded in a matrix with distinct material properties; (b) Polycrystalline materials which are assemblies of different small grains (crystals); (c) Two-phase flow model where liquid and gas share an interface and interact with each other; (d) Cracks nucleate along the boundary and inside the structure; (e) Dislocations lead to a discontinuous primal field where the jump is described by the Burgers vector $\boldsymbol{b}$; (f) A shear band occurs under intense localized shearing.

An appropriate geometric engine should be robust and work for an arbitrary number of discontinuity-mesh configurations. Still, designing a robust engine is extremely challenging, simply because it is not straightforward to write computational geometry functions that are robust when dealing with floating-point arithmetic. One

may opt to use external libraries at the expense of increasing the software dependency. For instance, open-source libraries such as the Computational Geometry Algorithms Library (CGAL) [88] have been used. Ren and Younis [89] used CGAL to describe and update the geometric description of cracks under the hydraulic fracture propagation. Rather than using a single library, Massing *et al.* [90] developed a geometric package that uses data structures and algorithms from CGAL and GTS [91], showing good performance in detecting intersections between two unfitted and overlapping meshes and integration over cut cells. These computational geometry libraries, however, make software's learning curve steeper and may not be necessary for some implementations (overkill). Therefore, implementing an in-house geometric engine is also a viable alternative and a popular one in the enriched FEM community. In the context of X/GFEM for fracture problems, Sukumar *et al.* [70, 92] described in detail the geometric operations for detecting elements that intersect with cracks and the data structures used to store relevant geometric information. Instead of using level set functions to represent cracks, Pereira *et al.* [93] adopted a mesh with triangular elements to describe a crack surface explicitly in 3-D, and briefly described the strategy used for creating integration elements. Soghrati *et al.* [94] listed all possible situations when a tetrahedral element is completely split by a material interface, and explained the scheme for performing tetrahedralization of the resulting subdomains. For speeding up geometric operations, Březina and Exner [95] proposed a set of the algorithms aiming to calculate intersections between 3-D background meshes and lower-dimensional simplices, whereby the number of calculations is minimized by reusing information obtained from neighboring elements. Yang *et al.* [96] proposed an approach for reconstructing an explicit description of heterogeneous materials based on scanning electron microscopy images, where Non-Uniform Rational B-Splines (NURBS) are used to represent the morphology of fibers and bounding boxes are used to check for intersections between newly-added and existing inclusions. Recently, Zhang *et al.* [35] discussed a basic geometric engine that is able to deal with multiple discontinuities cutting a single element hierarchically, whereby the entire hierarchy that results from splitting completely or partially a tetrahedron is stored in an ordered tree data structure. All these works, however, just provide a sufficient ad hoc solution for their particular problem. As a result, these ideas cannot be easily generalized. In addition, although effective, most works do not discuss efficiency of the computational geometry routines. Although bits and pieces on implementing a robust geometric engine to be used with unfitted/immersed/enriched FEMs can be found in the literature, a detailed discussion is still missing.

In this work we describe the objected-oriented design of a geometric engine that can be used to solve problems with discontinuities by means of enriched FEMs. Although discussed in the context of interface- and discontinuity-enriched FEMs, the engine is general and thus could also be used together with any unfitted FEM without much modification. The engine's main functions are: *i*) Perform intersection tests between discontinuities and finite elements; *ii*) Create enriched nodes that are collocated at the intersection between discontinuities and edges of background elements; *iii*) Create integration subdomains considering multiple discontinuities in-

tersecting a single element; *iv*) Store such hierarchies of integration elements in an ordered tree data structure; and *v*) Create physical groups that are then used to assign correct material properties or to refer to discontinuities. The design of the engine goes beyond basic requirements in many areas, including:

- **Generality** Discontinuities can be represented both implicitly (*e.g.*, via level set functions) and/or explicitly (*e.g.*, line segments, polygons or even lower-dimensional meshes). Integration subdomains of the background mesh in all dimensions can be created by rules or by more advanced techniques such as Delaunay triangulation (tetrahedralization) [97, 98];
- **Robustness** The use of tolerances is mitigated wherever possible, as tests based on them are prone to fail;
- **Efficiency** Given a coordinate, a space partitioning method (concretely a $k$-d tree) is used to find the background mesh element that contains it. In addition, a dual graph data structure of the background mesh is used to efficiently discover neighbors of cut elements (and to avoid intersection checks). Data on subdomains of cut elements' edges (or faces) is reused for generating subcells of the neighboring elements. An ordered tree data structure is used to store hierarchical element information and to efficiently iterate over quadrature elements (leaves of the tree). Finally, we exploit a flood-fill algorithm that works in conjunction with a dual graph data structure to determine which (background and integration) elements belong to a corresponding physical domain.

The design of the geometric engine is thoroughly explained with pseudo-code, and its computational complexity is investigated with several 2-D and 3-D examples of material interfaces and cracks. The capability of the proposed engine is illustrated by means of several examples. First, the topology of time-dependent discontinuities represented as level set is optimized for the minimal structural compliance in 3-D. Next, a crack junction problem, where cracks intersect at the same location, is solved with using DE-FEM. Later, a discontinuous model that includes intersected cracks and material interfaces is investigated. Furthermore, we overlap the unfitted mesh and polycrystalline structures described by polygons or polyhedra, where integration elements created align with the grain boundaries. Finally, the Standford Bunny model discretized by a surface mesh with triangular elements is fully immersed into a 3-D background mesh, where a new discretization with more elements are created along the exterior of the Standford Bunny.

## 3.2. Geometric engine

The proposed object-oriented design of the geometric engine is composed of several modules, as schematically illustrated in Figure 3.2:

*i*) `Discontinuities` $\mathcal{D}$ is a collection of discontinuities, which can be represented implicitly (*e.g.,* by means of a level set function) or explicitly (*e.g.*, by means of geometric primitives) (see Chapter 3.2.1);

*ii*) `SpacePartitioner` **S** is a $k$-d tree data structure for partitioning the space occupied by the background mesh, enabling fast search of background mesh element(s) containing a given point along a discontinuity (see Chapter 3.2.2);

*iii*) `DualGraph` **G** is a graph data structure of the background mesh for fast neighbor element search (see Chapter 3.2.3);

*iv*) `Interactor` is used for processing the interactions between the background mesh and discontinuities (see Chapter 3.2.4);

*v*) `Intersector` finds intersection points between discontinuities and the sides of background mesh elements (see Chapter 3.2.5);

*vi*) `ElementCreator` creates lower-dimensional and integration elements (see Chapter 3.2.6);

*vii*) `Mesh` **M** contains data structures that store data regarding finite element nodes $\mathcal{N}$, elements $\mathcal{E}$, and physical groups $\mathcal{P}$ (where geometrical entities are combined into more meaningful groups, *e.g.*, mathematical, functional or material properties [99]). **M** is updated via adding enriched nodes and new created elements, and creating new and modifying original physical groups (see Chapter 3.2.7).

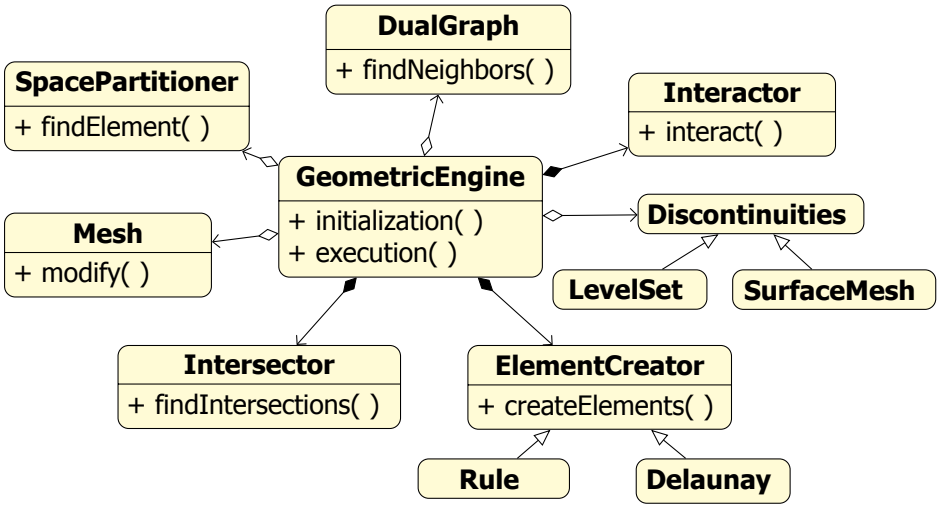



Figure 3.2: Diagram of modules composing the geometric engine represented the unified modeling language (UML) notation.

In addition, an associative container **I** keeps information about enriched nodes and subdomains of cut element edges and/or faces, and aids in creating integration elements of any cut element sharing the same cut edge/face. A hierarchical ordered tree data structure **T** is used for storing cut mesh elements and their children. This tree enables fast traversal of integration elements, which is required for computing their local contributions to system matrices and vectors. Pseudo-code for the behavior of this geometric engine is given in Algorithm 3.1.

**Algorithm 3.1** Geometric Engine: Pre-process a background mesh for the use in enriched finite element methods, such as DE-FEM

**Input:** A background mesh $\mathbf{M} := \{\mathcal{N}, \mathcal{E}, \mathcal{P}\}$ (node, element and physical group sets), a set $\mathcal{D}$ of discontinuities, and a set $\mathcal{P}_{\text{new}}$ of new physical groups

```
function initialization(M)
    S ← SpacePartitioner(M) – Create a k-d tree data structure
    G ← DualGraph(M) – Create a dual graph based on the background mesh
    return S, G
end function

function execution(D, P_new)
    I, T ← ∅, ∅ – Initialize enrichment associative container and tree data structure
    for d_i ∈ D do – Loop over discontinuities
        {I, T} ← Interactor.interact(G, S, d_i, I, T)
                – Interact the background mesh with the discontinuity
    M ← M.modify(D, P_new, I, T)
                – Modify mesh with the enriched nodes, elements and physical groups
    return M
end function
```

**Output:** Modified mesh $\mathbf{M} := \{\mathcal{N}, \mathcal{E}, \mathcal{P}\}$

### 3.2.1. Discontinuities

Both implicit and explicit descriptions are allowed for representing the geometry of discontinuities. While the former can be achieved, for instance, by means of a level set function, the latter is attained by using geometric entities in the form of a finite element mesh—even for fundamental geometric primitives. Since these two representation methods are fundamentally different, a separate implementation is required to deliver the same functionality, namely for determining intersections between the background mesh and discontinuities, and for detecting background and integration elements inside or outside discontinuities.

#### Level set

Level set provides a flexible way to model moving/evolving boundaries [100]. They have been widely used in many fields, such as multi-phase flows [101], image processing [102], and topology optimization [103, 104]. A level set function $\phi(\boldsymbol{x})$ describes boundaries in an implicit form as the iso-contour of a smooth function, for instance the zeroth level contour. Then, $\phi(\boldsymbol{x})$ defined in domain $D$ is expressed as

$$\begin{cases} \phi(\boldsymbol{x}) = 0 \Leftrightarrow \boldsymbol{x} \in \partial\Omega, \\ \phi(\boldsymbol{x}) < 0 \Leftrightarrow \boldsymbol{x} \in \Omega, \\ \phi(\boldsymbol{x}) > 0 \Leftrightarrow \boldsymbol{x} \in (D \backslash \overline{\Omega}), \end{cases} \tag{3.1}$$

where $x$ is the Cartesian coordinate, $\Omega$ represents a subset of $D$ with closure $\overline{\Omega}$ (see Figure 3.3(a)). Level set can be used to represent a straight boundary, a circular material interface (shown in Figure 3.3(b)) and even the structural topology with more complex geometric configurations [105]. In the proposed geometric engine, level sets are used to describe both static material interfaces in multi-phase materials, and the evolving boundaries in topology optimization. For the latter, radial basis functions (RBFs) are used to discretize the level set function [106, 107], and coefficients multiplying RBFs are considered as design variables that are updated during the optimization process.

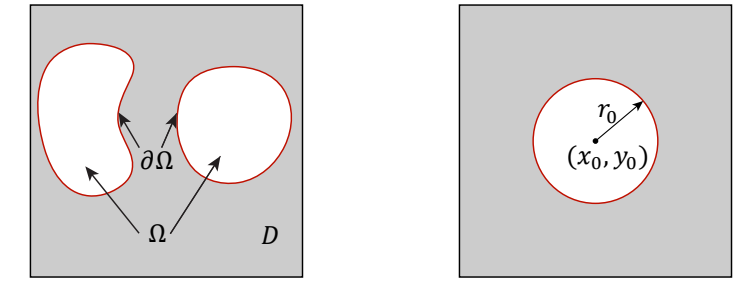



Figure 3.3: (a) Level set used to describe a domain $D$ with an embedding void domain $\Omega$, and the interface $\partial\Omega$ (marked with a red curve); (b) Level set function $\phi(x,y) = \sqrt{(x-x_0)^2+(y-y_0)^2} - r_0$ used to describe a circular interface in 2-D.

### Surface mesh

Explicit representations based on geometric entities, such as segments, triangles, rectangles, and polygons, are used to describe both weak and strong discontinuities in form of a surface mesh with lower dimension compared to that of the background mesh. As shown in Figure 3.4(a), a 1-D mesh consisting of line elements (marked

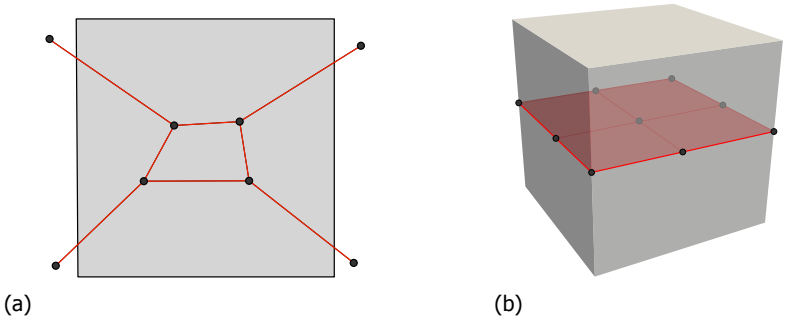



Figure 3.4: (a) A 1-D mesh with eight line elements and eight nodes that intersects within a square domain; (b) A 2-D mesh with four rectangular elements and nine nodes that splits a cube completely can be used to describe a material interface or a crack.

with red segments) is used to describe eight cracks intersecting a 2-D square domain. Figure 3.4(b) shows the representations of discontinuities in 3-D, whereby a surface mesh with four rectangular elements that can either represent a crack or a material interface, splits a cubic domain completely.

### 3.2.2. Space patitioner

`SpacePatitioner` constructs a $k$-d tree data structure $\mathbf{S}$ for partitioning the $k$-dimensional space occupied by the background mesh $\mathbf{M}$. As with a general $k$-d tree for locating points in space, the $k$ axes are cycled as the tree is constructed. Each non-leaf node in the tree is defined as a plane perpendicular to the specific axis, dividing the space into two parts at either side of the plane. It is worth noting that elements intersected with the plane are assigned to both parts. The coordinate associated with non-leaf node is calculated as the average of elemental centers along the corresponding axis. The tree construction is finalized when the element set at either side of the splitting plane remains the same, and then these elements are stored in the tree as a leaf node. This $k$-d tree is used for finding a background element enclosing a given point, operation that is carried out by the function `findElement`. Given a point, we traverse the tree from the root and we use the point coordinates to guide the search until a leaf node is found containing potential enclosing elements. Then geometric predicates are performed to check which element (or elements in case the point lies at an edge shared by two elements) contains the point.

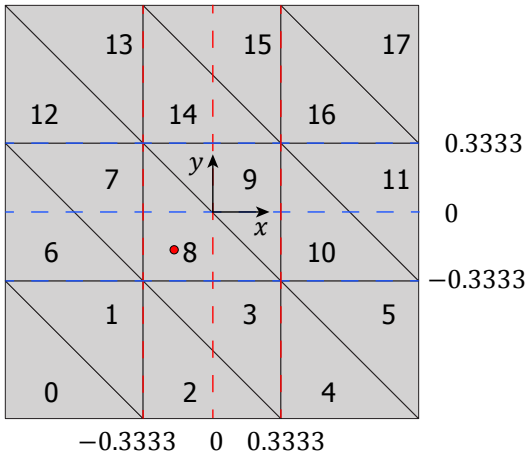



Figure 3.5: The background mesh consisting of $3 \times 3 \times 2$ triangular elements shown with numbers, where a plane perpendicular to an axis is represented as a dashed line.

As an example, we show in Figure 3.5 the space partitioning of a square $2 \times 2$ domain discretized by a background mesh with $3 \times 3 \times 2$ triangular elements; the range for both $x$ and $y$ axes is $[-1, 1]$. For the corresponding $k$-d tree data structure shown in Figure 3.6, the root node divides the space with the plane $x = 0$. This plane divides the background mesh into two groups of elements

$\{0, 1, 2, 3, 6, 7, 8, 9, 12, 13, 14, 15\}$ and $\{2, 3, 4, 5, 8, 9, 10, 11, 14, 15, 16, 17\}$ to the left and right sides, respectively. Notice there are several elements that belong to both groups since they are cut by the plane, namely $\{2, 3, 8, 9, 14, 15\}$. For the next level in the hierarchy the plane $y = 0$ is used, which further subdivides elements in the previous level to either side. For example, the left subtree of the root is split into $\{0, 1, 2, 3, 6, 7, 8, 9\}$ and $\{6, 7, 8, 9, 12, 13, 14, 15\}$. For the former element group, in the next hierarchical level a plane $x = -0.3333$ is obtained as the elements' centroid. Then this element group is divided into $\{0, 1, 6, 7\}$ and $\{2, 3, 8, 9\}$. For the next hierarchical level the $y = -0.3333$ is computing, further subdividing elements into $\{0, 1\}$ and $\{6, 7\}$, which are finally stored as leaves in the three. The subdivision of other element groups follows the same procedure. Finally, we obtain a $k$-d tree with five levels, with each leaf node containing two elements. In order to detect a point with coordinates $(-0.25, -0.25)$ (marked with a red circle in Figure 3.5, we follow the red path shown in the tree (see Figure 3.6), till arriving to the element set $\{8, 9\}$, after which we find the point is contained in the 8th element using simple predicates.

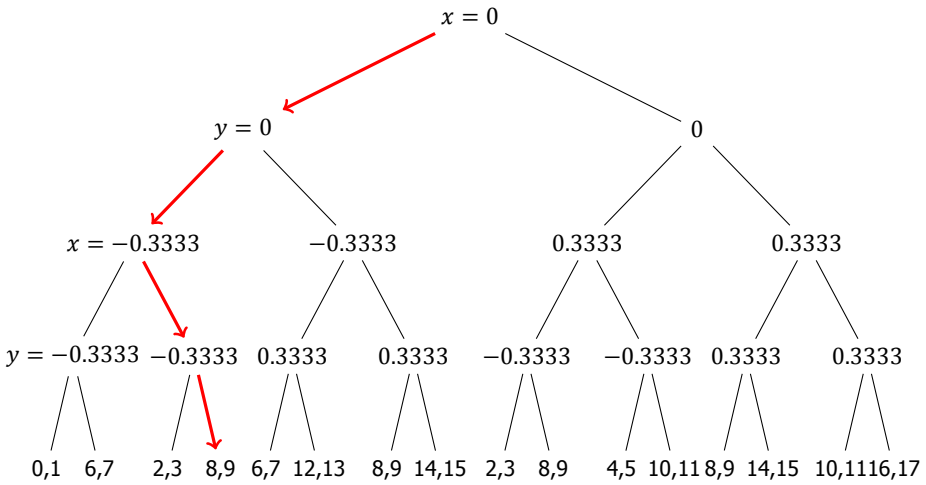



Figure 3.6: The corresponding $k$-d tree data structure, where the red lines with arrows show the path to find the elements enclosing the point $(-0.25, -0.25)$.

### 3.2.3. Dual graph

`DualGraph` is an object that represents a dual graph data structure $\mathbf{G} := \{\mathcal{V}, \mathcal{S}\}$ (composed of sets of vertices $\mathcal{V}$ and edges $\mathcal{S}$) of a given background mesh $\mathbf{M}$. In this graph, vertices $v_i, v_j$ represent the $i$th and $j$th finite elements, respectively, and they are connected with an edge if the elements share an edge. For constructing this graph structure, the background elements are iterated and an auxiliary data structure is used to store elements' edges that are later checked for overlap with other elements. Figure 3.7 shows a background mesh with $3 \times 3 \times 2$ triangular

elements and its corresponding dual graph structure, which clearly displays with edges adjoining elements. This undirected graph is then used to find neighbors of any given target element via the function `findNeighbors`.

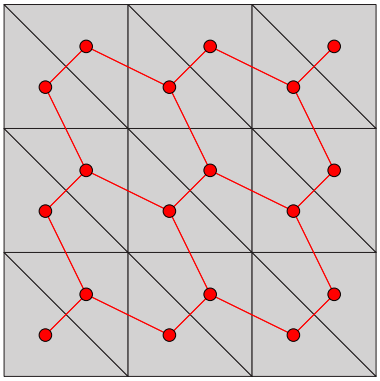


Figure 3.7: The background mesh consisting of $3 \times 3 \times 2$ triangular elements with the corresponding graph structure (shown with red vertices and edges).

### 3.2.4. Interactor

`Interactor` takes care of processing intersections between discontinuities and the background mesh. `Intersector` is used to find intersections between discontinuities and elements, forwarding that information to `ElementCreator` to create children elements, and for storing the newly created elements in the ordered tree data structure **T**. For bookkeeping purposes, an enrichment associative container **I** is used, where the keys are cut element edges and/or faces. Their corresponding values include enriched nodes on the keys and newly created edges and/or faces. Keeping these data can facilitate the creation of integration elements. For instance, if an element side is already split into several subparts, we can directly get this information from **I** for creating children of background elements sharing this cut edge or face. Although it is possible to create integration elements of a cut background element immediately after finding it intersects a discontinuity, we defer this operation to a later time. This is done because a single background element may be cut by multiple discontinuities, and the most effective way to create integration elements requires *all* enriched nodes *a priori*. Therefore, we first process all discontinuity-element intersections with the aid of a queue **Q**, and we store cut elements in a set $\mathcal{W}$ so that integration elements are created at a later stage.

The pseudo-code of function `interact` in `Interactor` is given in Algorithm 3.2. Firstly, looping over a list of points that describe a given discontinuity $d_i$, we detect elements $e$ that intersect with this discontinuity by calling function `findElement` in `SpacePartitioner` **S**, and store each cut element into the queue **Q** and the set $\mathcal{W}$. Secondly, the queue **Q** is iterated over cut elements $e$, and `Intersector` is called to calculate the intersections of the cut element with the given disconti-

nuity $d_i$. Afterwards, enriched nodes are created at these intersections and stored in the enrichment associative container $\mathbf{I}$ together with their corresponding cut element's edges/faces. The function `findNeighbors` in `DualGraph` $\mathbf{G}$ is then used to find this cut element's neighbors. If these neighboring elements are not yet in the set $\mathcal{W}$ or are not already processed, they are added subsequently to the queue and set. We then continue to process stored components in the queue until $\mathbf{Q}$ becomes empty. As all intersections between background elements and the discontinuity $d_i$ are detected, the function `createElements` in `ElementCreator` is called to create children elements, which include both integration elements and lower-dimensional elements along the discontinuity. Meanwhile, these elements $e_h$ are stored into the tree data structure $\mathbf{T}$, where they are set as children of the parent cut background elements. This tree data structure $\mathbf{T}$ will be later used to update the background mesh data.



**Algorithm 3.2** Interact a background mesh with a discontinuity

**Input:** A dual graph $\mathbf{G}$, a $k$-d tree data structure $\mathbf{S}$, the $i$th discontinuity $d_i$, a tree data structure $\mathbf{T}$, and an enrichment associative container $\mathbf{I}$

```
function interact(G, S, d_i, I, T)
    e ← S.findElement(d_i.points)
                – Obtain elements intersected with the discontinuity
    Q, W ← ∅, ∅ – Initialize a queue and a set
    Q, W ← addToQueue(e), addToSet(e)
                – Add cut elements to the queue and set
    do
       e_i ← getQueue(Q) – Get a cut element and remove it from queue
       x_i, I ← Intersector.findIntersections(e_i, d_i, I)
                – Obtain intersections and update enrichment associative container
       if x_i ≠ ∅ then
          e_n ← G.findNeighbors(e_i) – Obtain the neighboring elements
          if e_n ∉ W then
             Q, W ← addToQueue(e_n), addToSet(e_n)
                – Add neighboring elements to the queue and set
    while Q ≠ ∅
    for e_i ∈ W do
       e_h ← ElementCreator.createElements(e_i, d_i, I)
                – Create children elements
       T ← addToTree(e_h, e_i, T) – Add children elements to the tree data structure
    return I, T
end function
```

**Output:** Enrichment associative container $\mathbf{I}$ and tree data structure $\mathbf{T}$

### 3.2.5. Intersector

`Intersector` is the object used to find intersections between element edges (faces in 3-D) and a given discontinuity. Also, if applicable, it also detects points that belong to discontinuities inside/outside background elements, *e.g.*, those of a surface mesh. The corresponding pseudo-code of the function `findIntersections` is given in Algorithm 3.3, where enriched nodes $\boldsymbol{x}_i$ are created and stored in the enrichment map $\mathbf{I}$. Depending on the dimension of the background mesh, different approaches are implemented to detect element-discontinuity intersections. The procedure used is determined by the type of discontinuity since finding intersections for explicit representations follows a different approach than that used for implicit representations. For simplicity, we assume that the background mesh consists of triangular (tetrahedral) elements in 2-D (3-D).



**Algorithm 3.3** Find intersections between an element and a discontinuity

**Input:** $i$th background element $e_i$, $i$th discontinuity $d_i$, and an enrichment associative container $\mathbf{I}$

```
1: function findIntersections(e_i, d_i, I)
2:     dim ← e_i.dimension – Get the element dimension
3:     do
4:         x_i, I ← d_i.findIntersections(e_i, I) – Obtain the intersections
5:         dim ← dim−1 – Check the lower-dimensional geometric entity
6:     while dim ≠ 0
7:     return x_i, I
8: end function
```

**Output:** Intersections $\boldsymbol{x}_i$ and enrichment associative container $\mathbf{I}$

For a level set implicit representation, the approach is quite straightforward as it remains the same regardless of the mesh dimension. Firstly, we iterate over every edge of the background element and calculate their corresponding nodes' level set values. If the level set signs are not the same, the edge is crossed by a discontinuity at the zero level set value. The location of intersection point is then simply determined by interpolation.

For a crack described by a surface mesh, the corresponding procedure is based on algorithms for finding segment-segment (in 2-D) or segment-plane (in 3-D) intersections. To illustrate this type of intersection, consider in Figure 3.8 a discontinuity represented explicitly via a triangular (t3) element intersects a tetrahedral (T4) element. In this situation, the tetrahedral element is considered as a 3-D geometric entity that is composed of a collection of four triangular faces (2-D geometric entities) and six edges (1-D geometric entities). Therefore, several computational geometry tests are conducted to find the intersections. Firstly, we detect whether any of the discontinuity nodes lie inside the tetrahedron (see a blue node in Figure 3.8). Then we create the corresponding enriched node at the same location and add it to the enrichment associative container $\mathbf{I}$. Secondly, we iterate over the four triangular

faces and determine whether they are intersected by any of the three edges of the triangle. Enriched nodes are then created at intersection locations found (see red nodes in Figure 3.8) and stored into the enrichment associative container $\mathbf{I}$ with linking to the corresponding triangular face; This data will be used when computing intersections in the neighboring element sharing the same face. Finally, intersections between the six edges and the discontinuity's triangular surface elements are determined (see a white node in Figure 3.8). As before, new enriched nodes created are added to the enrichment associative container $\mathbf{I}$ and associated with the corresponding tetrahedral edges. Obviously, it is more involved to handle explicit discontinuities than implicit ones since more operations are required. Noteworthy, this procedure works for any (surface and background) element type, for instance, quadrilateral (Q4) or hexahedral (H8) elements.



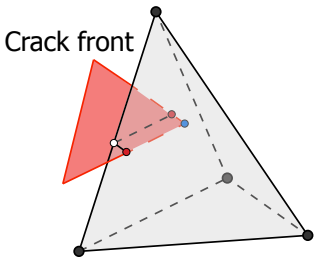



Figure 3.8: A crack represented by a triangular surface element (marked with red) intersects a tetrahedral element. Intersection tests are conducted to determine nodes lying inside the tetrahedron (marked with a blue circle) and along its boundaries (see red nodes located at two surfaces and a white node located within an edge).

### 3.2.6. Element creator

`ElementCreator` uses two different strategies, *i.e.*, lower-dimensional and integration elements are created based on specific rules or via a certain algorithm. For the former, represented by the object `Rule`, considers all possible cut situations and sets them as templates for generating lower-dimensional and integration elements. The latter could use, for instance, Delaunay triangulation (tetrahedralization) [97, 98], and is represented accordingly by a `Delaunay` object.

The procedure for creating new elements is elucidated based on Delaunay's algorithm, which is actually the only technique to handle complex situations where elements are not fully split. Since creating 2-D triangular subdomains is included in the procedure that constructs a discretization in 3-D, we focus on the description of the latter. Algorithms 3.4 and 3.5 show pseudo-code for the algorithm used to create lower-dimensional and integration elements. For clarity, we denote a cut background element as the *parent element* and integration elements that belong to it as *children* or *integration* elements. Figure 3.9(a) shows a tetrahedral element intersected by a discontinuity represented as a surface mesh composed of three quadrilateral (Q4) elements (marked with different colors). We first create lower-dimensional elements along the discontinuity, so surface elements describ-

ing the discontinuity are split into smaller subdomains in the function `createDiscontinuityElements`. Since enriched nodes along the discontinuity had already been stored by `Intersector` in the enrichment associative container $\mathbf{I}$, it is now straightforward to create these lower-dimensional elements. In addition, these elements are set as internal triangular faces for creating *children* elements using the function `createIntegrationElements`. If these lower-dimensional elements are not considered when creating *integration* elements, an incorrect situation could emerge, where internal faces do not match the original surface elements (see Figure 3.9(a)). Later, triangular faces of this tetrahedron that are cut by surface elements are handled, where the corresponding subfaces are created by using Delaunay triangulation. Noteworthy, it is necessary to trace intersected segments created between each surface element and the triangular faces, which are shown as red segments in Figure 3.9(b). After obtaining children faces, we store them into the enrichment associative container $\mathbf{I}$. For the neighboring element sharing the same cut face, these segments will be retrieved and not recreated. After all cut triangular faces are handled, their corresponding subfaces, together with non-cut faces, are used to create children tetrahedral elements by calling constrained Delaunay tetrahedralization.



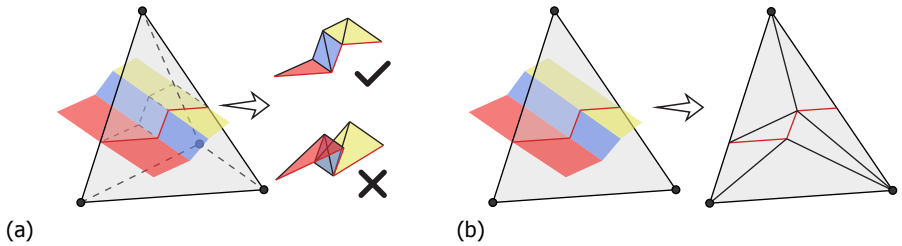



Figure 3.9: An explicit discontinuity composed of three quadrilateral elements intersects a tetrahedral element. (a) Lower-dimensional elements created along the discontinuity are set as internal faces when creating integration elements (top), and wrong internal faces (that do not match the original surface elements) could be created without considering lower-dimensional elements (bottom); (b) When creating the children elements of a triangular face, intersections between the face and this surface mesh (marked with red segments) are considered as constrained segments.

**Algorithm 3.4** Create children elements of a background element

**Input:** A parent element $e_p$, $i$th discontinuity $d_i$, and an enrichment associative container $\mathbf{I}$

```
1: function createElements(e_p, d_i, I)
2:     e_{d_i} ← createDiscontinuityElements(e_p, d_i)
3:                 – Create lower-dimensional elements along the discontinuity
4:     e_c ← createIntegrationElements(e_p, e_{d_i}, d_i, I)
5:                 – Create children elements of the background element
6:     return e_{d_i}, e_c
7: end function
```

**Output:** Lower-dimensional elements along the discontinuity $e_{d_i}$ and volumetric children elements $e_c$

**Algorithm 3.5** Create volumetric integration elements

**Input:** A parent element $e_p$, lower dimensional elements $e_{d_i}$, $i$th discontinuity $d_i$, and an enrichment associative container $\mathbf{I}$

```
function createIntegrationElements(e_p, e_{d_i}, d_i, I)
    dim ← getDimension(e_p) – Get dimension of the parent element
    X_p ← getNodes(e_p) – Get original nodes of the parent element
    x_p ← getEnrichedNodes(e_p, I) – Get enriched nodes of the parent element
    if dim = 1 then
        l ← createSegments(X_p, x_p) – Create segments of the edge
        e_c ← createElements(l) – Create children elements
    if dim = 2 then
        edges ← getEdges(e_p) – Get edges of the parent element
        for edge ∈ edges do
            newEdges ← createIntegrationElements(edge, e_{d_i}, d_i, I)
                – Create children of each edge
        t ← Triangle.MeshInfo() – Initialize Triangle package
        t ← t.set_points(X_p, x_p) – Pass all nodes to Triangle
        t ← t.set_facets(newEdges) – Pass all edges to Triangle
        e_c ← Triangle.build(t) – Create children elements
    if dim = 3 then
        faces ← getFaces(e_p) – Get faces of the parent element
        for face ∈ faces do
            newFaces ← createIntegrationElements(face, e_{d_i}, d_i, I)
                – Create children of each face
        t ← Tetgen.MeshInfo() – Initialize Tetgen package
        t ← t.set_points(X_p, x_p) – Pass all nodes to Tetgen
        t ← t.set_facets(newFaces) – Pass all faces to Tetgen
        e_c ← Tetgen.build(t) – Create children elements
    return e_c
end function
```

**Output:** Volumetric children elements $e_c$

### 3.2.7. Mesh

The last step of the geometric engine is transforming the background mesh into the enriched discretization. This consists of three main functions as shown in Algorithm 3.6: `modifyNodes`, `modifyElements`, and `modifyPhysicalGroups`. The purpose of functions `modifyNodes` and `modifyElements` is straightforward, since they update the mesh data structures adding newly created enriched nodes and integration elements. When dealing with multi-phase problems, it is required to specify *physical groups* to different parts of the mesh that correspond to different

materials. The function `modifyPhysicalGroups` creates new physical domains associated with discontinuities and changes the original physical domains (see Algorithm 3.7). Geometric groups, which include elements inside or at either side of a given discontinuity, are a prerequisite for the creation of physical groups. Once geometric groups are defined, boolean operations are performed on them for creating the physical groups, which could have either the same dimension as that of the background mesh or a lower dimension—*e.g.*, lower-dimensional elements created along discontinuities. If there is no need to create new physical groups, for instance, when modeling a fracture problem with a single material phase, all integration and background elements are then assigned to the original physical group of the background mesh.



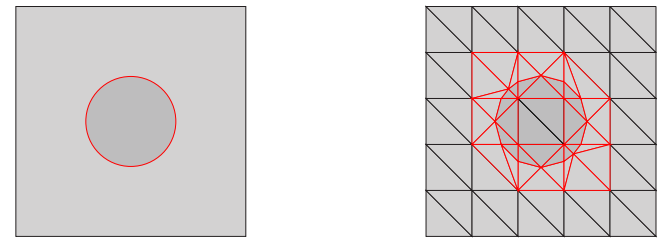
**Algorithm 3.6** Modify the mesh data structure

**Input:** A discontinuity set $\mathcal{D}$, a new physical group set $\mathcal{P}_{\text{new}}$, a tree data structure $\mathbf{T}$, and an enrichment associative container $\mathbf{I}$

```
function modify(D, P_new, I, T)
    N ← modifyNodes(I)  – Add enriched nodes to the mesh data structure
    E ← modifyElements(T)
                – Add newly created elements and mask the cut background elements
    P ← modifyPhysicalGroups(D, P_new, T)
                – Add new physical groups and modify original ones
    return N, E, P
end function
```

**Output:** Modified mesh $\mathbf{M} := \{\mathcal{N}, \mathcal{E}, \mathcal{P}\}$

To explain this last step of the geometric engine, consider in Figure 3.10(a) a 2-D square matrix containing a circular inclusion (marked with darker color) of a different material, which is described by a level set function. The model is discretized by a mesh composed of $5 \times 5 \times 2$ triangular elements. Then, 42 integration elements



(a) (b)

Figure 3.10: (a) A circular inclusion (marked with darker color and red interface) is embedded into a square matrix; (b) A 2-D background mesh with $5 \times 5 \times 2$ triangular elements intersected by the circular inclusion creates integration elements (marked with red triangles) that are assigned to a new physical group, together with uncut elements that lie fully inside the inclusion (marked with darker color).

and 14 lower-dimensional line elements along the material interface are created via cutting 14 background elements (see Figure 3.10(b)).

**Algorithm 3.7** Create and modify physical groups

**Input:** A discontinuity set $\mathcal{D}$, a new physical group set $\mathcal{P}_{\text{new}}$, and a tree data structure $\mathbf{T}$

```
function modifyPhysicalGroups(D, P_new, T)
    D_new ← selectDiscontinuities(D, P_new)
                – Split original discontinuities if needed
    G_group ← propagateGeoGroups(D_new, P_new, T)
                – Create geometric groups for discontinuities in D_new if needed
    P ← assignPgroups(G_group, P_new)
                – Create new physical groups and add them to M if needed
    return P
end function

function propagateGeoGroups(D_new, P_new, T)
    dims ← list(dimensions(P_new)) – Get dimensions of the new physical groups
    for d_i ∈ D_new do – Loop over discontinuities to create geometric groups
        for dim ∈ dims do
            if dim == M.dim then – For volumetric geometric groups
                e_p ← intersectDiscontinuity(d_i)
                    – Find a background element intersecting d_i
                e_c ← getTreeLeaves(e_p, T) – Get children elements
                e_in ← findElementsInside(d_i, e_c)
                    – Find all elements inside d_i by going through neighbors of e_c
                G_group ← addToGroup(e_in) – Add elements to this geometric group
            if dim < M.dim then – For lower-dimensional geometric groups
                e_{d_i} ← getDiscontinuityElements(d_i)
                    – Get lower dimensional elements along the discontinuity
                G_group ← addToGroup(e_{d_i}) – Add elements to this geometric group
    return G_group
end function

function assignPgroups(G_group, P_new)
    for p_i ∈ P_new do
        p_i ← boolean(G_group) – Create new physical groups with boolean operations
        P ← update(p_i) – Update physical groups
    return P
end function
```

**Output:** Updated physical group $\mathcal{P}$

For assigning these elements to their corresponding geometric groups, we start with any integration element and determine its group by checking its nodes' level set values. Later, neighboring integration elements are also assigned to their corresponding geometric group based on the same strategy. As only one discontinuity is considered in this example, the inclusion's geometric group coincides with a physical group associated with different material properties. As shown in Figure 3.10(b), 18 integration and 2 original elements are added to this new physical group—and are removed from the their original physical group.

In the case of handling models that contain multiple interfaces and/or cracks, the function `selectDiscontinuities` is used to select the discontinuities associated with new physical groups $\mathcal{P}_{\text{new}}$. For instance, when solving a problem with a crack and an inclusion, only the discontinuity representing the material interface is required to create its corresponding physical group. However, when modeling a problem with multiple phases, geometric groups could be combined into a single physical group with a single material property. The geometric engine makes this possible by means of boolean operations performed on geometric groups associated with different discontinuities. For instance, Figure 3.11 shows two overlapping material interfaces represented implicitly. In order to collect elements shared by them into a single physical group, as shown in Figure 3.11(a), the boolean operation `Intersection` is used to choose elements from the corresponding geometric groups. Other operations, such as `Subtraction` or `Union`, are also available to create complex physical groups (see Figures 3.11(b) and (c), respectively).

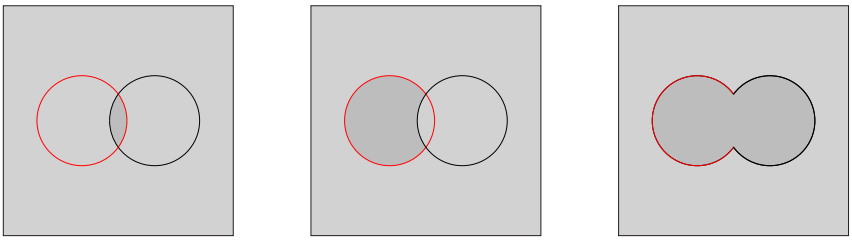


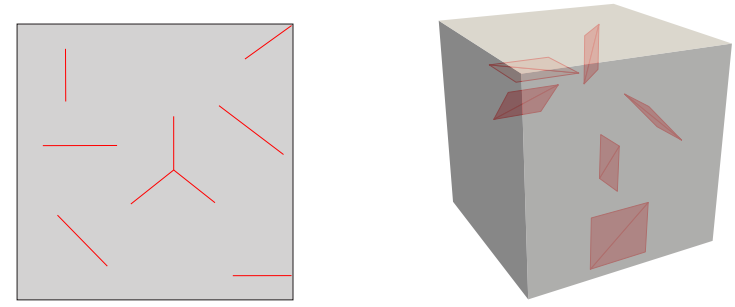

Figure 3.11: Two overlapping circular material interfaces, where different parts are assigned to a new physical group: (a) Intersection between discontinuities; (b) Subtraction of the right discontinuity; (c) Union of discontinuities.

## 3.3. Complexity

The complexity of the geometric engine depends on that of each individual component. It is well known that finding the nearest neighbor in a $k$-d tree for any given point has time complexity $\mathcal{O}(\log|\mathcal{E}|)$ on average, where $|\mathcal{E}|$ is the number of elements in the background mesh. `DualGraph` takes constant time (*i.e.*, $\mathcal{O}(1)$) to obtain neighbors of the target element, where a maximum number of neighbors equals to the number of element sides. The time complexity of `Interactor` de-

pends on the number of background elements intersected by discontinuities. Every cut element is tested for intersection against all discontinuities for constructing new children elements; this is done by iterating over all element edges and faces, which asymptotically has constant time complexity $\mathcal{O}(1)$. Considering Delaunay's algorithm for the creation of children elements, the time complexity is $\mathcal{O}(|\mathcal{N}_e| \log |\mathcal{N}_e|)$ in 2-D [108] and $\mathcal{O}(|\mathcal{N}_e|)$ in 3-D [109], respectively, where $|\mathcal{N}_e|$ is the number of original and enriched nodes in a cut background element. Noteworthy, as $|\mathcal{N}_e|$ is relatively small, asymptotically `ElementCreator` has constant time complexity both in 2-D and 3-D. Since discontinuities have one dimension less than that of the background mesh—*i.e.*, they are lower-dimensional manifolds—as problems increase in size, the number of intersected (and integration) elements becomes less significant compared to the total number of mesh elements. We can therefore estimate the total complexity of the `Interactor` as $\mathcal{O}(|\mathcal{E}|^{1/2})$ and $\mathcal{O}(|\mathcal{E}|^{2/3})$ in two and three dimensions, respectively.

2-D and 3-D models with discontinuities are used to corroborate the time complexity of the geometric engine. For 2-D, a square domain contains 9 discontinuities (represented by 9 line elements) with arbitrary orientations (see Figure 3.12(a)). This computational domain is then discretized with increasingly finer meshes with 882, 3362, 13 122, 51 842, 206 082, 821 762 and 3 281 922 constant strain triangles. The cubic 3-D model contains 6 discontinuities described by 12 triangular elements (see Figure 3.12(b)), and is discretized by meshes containing 750, 4374, 29 478, 215 622 and 1 647 750 linear tetrahedra. In order to show the efficiency of the geometric engine, we first compare the computational complexity for finding cut elements via a brute-fore approach that loops over *all* background elements and uses `SpacePartitioner` to locate elements that contain given points. The computational time $\varkappa$ has been normalized by the maximum value obtained by all methods. From Figures 3.13(a) and (b), it can be seen that, asymptotically, the brute-fore approach and `SpacePartitioner` associated with the problem size scales as $\mathcal{O}(|\mathcal{E}|)$ and $\mathcal{O}(\log |\mathcal{E}|)$, respectively.



(a) (b)

Figure 3.12: (a) A 2-D model including 9 cracks represented by line elements; (b) A 3-D model with 6 discontinuities described as triangular elements.

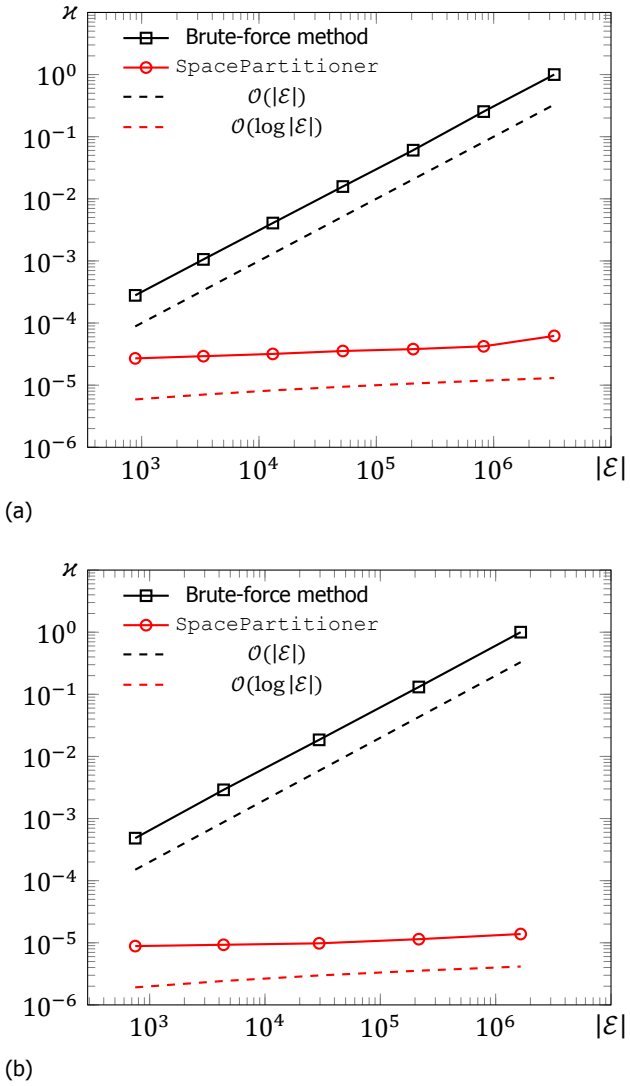
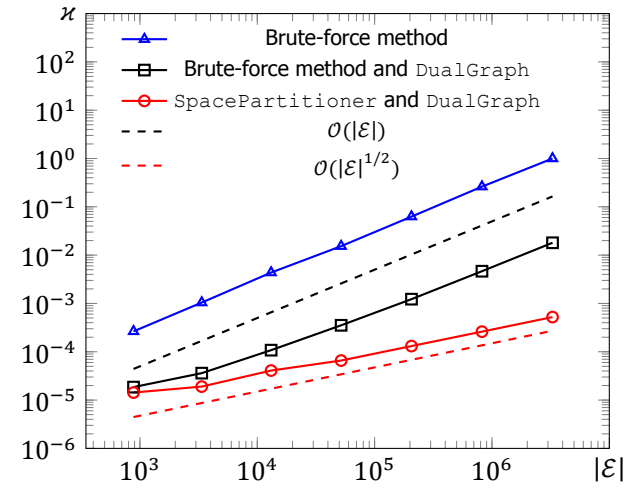


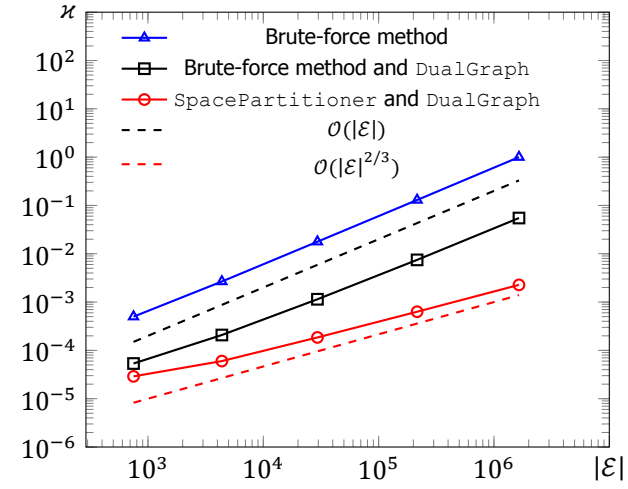

Figure 3.13: Complexity for finding cut elements with given points of discontinuities via Brute-force method and `SpacePartitioner` in (a) 2-D and (b) 3-D.

The computational time for interacting discontinuities with the background mesh is studied via three different strategies: *i*) A brute-force approach that iterates over background elements; *ii*) The brute-force approach to find cut elements but calls `Dualgraph` to find neighbors of cut elements; *iii*) Adopting both `SpacePartitioner` and `Dualgraph`. The normalized computational time $\varkappa$ for 2-D and 3-D examples with respect to the number of background elements $|\mathcal{E}|$ is shown in Figures 3.14(a) and (b), respectively. For both examples, the first strategy has com-

plexity $\mathcal{O}(|\mathcal{E}|)$ and is therefore the most time-consuming; the second strategy has the same complexity, but takes less time with the help of `Dualgraph`; the last approach is the least expensive one, scaling as predicted above as $\mathcal{O}(|\mathcal{E}|^{1/2})$ in 2-D and $\mathcal{O}(|\mathcal{E}|^{2/3})$ in 3-D.




(a)




(b)

Figure 3.14: Complexity for interacting discontinuities and the background mesh via looping over elements, Brute-fore method with `DualGraph`, and `SpacePartitioner` with `DualGraph` in (a) 2-D and (b) 3-D.

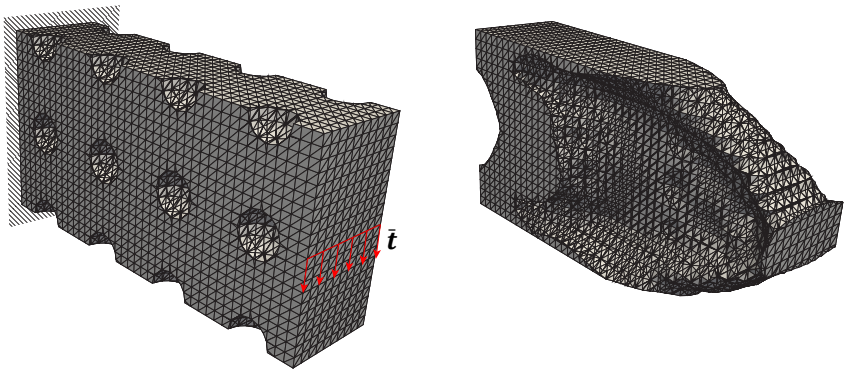
## 3.4. Numerical examples

In this section we demonstrate the capability of the proposed geometric engine with several 2-D and 3-D discontinuous problems. Two examples focus on creating a new discretization without conducting analysis, whereas the rest use IGFEM and DE-FEM to solve weakly and strongly discontinuous problems, respectively. Since no units are given, geometrical parameters, Young's moduli, and magnitudes of applied tractions can be taken under any consistent unit system.



### 3.4.1. Topology optimization

The first example considers evolving material interfaces described by a level set function in the context of topology optimization. Figure 3.15(a) shows the schematic of a 3-D cantilever beam problem with the corresponding initial hole seed design used; the back surface is clamped and a downward line traction $\bar{\boldsymbol{t}}$ is applied along the middle of the front surface. The cantilever beam is discretized by a structured




(a) (b)

Figure 3.15: 3-D Cantilever beam topology optimization problem: (a) Schematic where the back surface is clamped and a downward line traction $\bar{\boldsymbol{t}}$ is applied along the middle of the front surface. The figure also shows the initial hole seed design; (b) Optimized design for minimal structural compliance after 200 iterations.

background mesh with $40 \times 20 \times 10 \times 6$ tetrahedral elements. Young's moduli $E_s = 1$ and $E_v = 10^{-6}$ are assigned for solid and void domains, respectively. Both of them have the same Poisson's ratio 0.3. Although it is possible to directly use nodal level set values as design variables—thus the set function is discretized by means of standard finite element shape functions—we use instead compact radial basis functions (RBFs) [106] to parameterize the level set function. This choice, whereby the design space is decoupled from the discretization, shows benefits comparable to a filter in density-based topology optimization [26]. The design space with the same dimension as that of the cantilever beam is discretized using a grid of $21 \times 11 \times 6$ RBFs, with a supported radius of $r_s = \sqrt{2} \cdot a$, where $a$ is the grid size. The cantilever beam is optimized for minimum structural compliance (*i.e.*, maximum

stiffness) with solid material constrained at $55\%$ of the total volume of the design domain. Figure 3.15(b) shows the optimized design with satisfied volume constraint after 200 iterations. This example shows the geometric engine's ability to handle moving level set—thus implicit—weak discontinuities in 3-D. For more information on level set-based topology optimization using IGFEM we refer to van den Boom *et al.* [26].

### 3.4.2. Intersecting discontinuities

#### Fracture junction

In this example DE-FEM is used to resolve multiple intersecting cracks. As shown in Figure 3.16(a), a star-shaped fracture is placed at the center of a finite plate with dimensions $1.25 \times 1.25$, where unit magnitude bi-axial tractions $\bar{\boldsymbol{t}}$ are prescribed normal to the boundary. Cracks, represented explicitly as line elements of

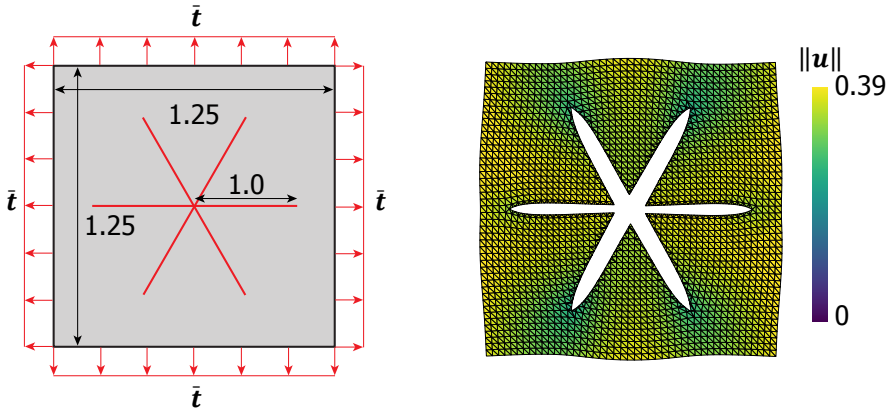



Figure 3.16: (a) Schematic diagrams for star-shaped cracks in a finite plate; (b) The displacement $\boldsymbol{u}$ distribution with the deformed model discretized by the background mesh consisting of $45 \times 45 \times 2$ triangular elements.

unit length, intersect at the center of the domain. The background mesh used has $45 \times 45 \times 2$ triangular elements. After detecting intersections between cracks and background elements and creating integration elements, a new discretization with 712 integration elements is obtained. For modeling crack junctions using DE-FEM, the number of strong nodes (for resolving the jump in displacement) added in the junction depends on the number of intersecting cracks. In other words, we need to create a displacement jump between each two intersecting cracks, and if $n$ cracks intersect, we need at least $n-1$ strong enriched nodes, since the last displacement jump can be formed by a linear combination of all other displacement jumps. In our example, since six line segments intersect, five strong nodes are generated. The resulting displacement field is shown in Figure 3.16(b). More details on the use of DE-FEM for resolving multiple intersecting discontinuities can be found in Liu *et al.* [43].

### Resolving both weak and strong discontinuities

Figure 3.17(a) shows a $2 \times 2$ plate that contains a material interface (marked with blue) and a crack (marked with red). While the former is represented using a level set function, for the latter we use a one-dimensional surface mesh. An elastic material with Young's modulus $E_1 = 1$ and Poisson's ratio $\nu_1 = 0$ is assigned to the domain at the left side of the interface, and $E_2 = 2$ and $\nu_2 = 0$ is assigned to the rest of the plate. While the left boundary is fully constrained, only the vertical displacement is restricted along top and bottom edges. Horizontal tractions $\bar{\boldsymbol{t}}_1 = \boldsymbol{e}_1$ and $\bar{\boldsymbol{t}}_2 = 2\boldsymbol{e}_1$ are imposed on the right edge below and above the crack, respectively. For this example, which aims at showing the hierarchical data structure used to store the information of cut elements and their corresponding subdomains, a simple background mesh with $5 \times 5 \times 2$ triangular elements is used. The result obtained via DE-FEM is shown in Figure 3.17(b), where it can be seen that two independent kinematic fields with constant state of stress $\boldsymbol{\sigma}$ are recovered at either side of the crack. The resulting discretization after processing the interaction between both discontinuities with the non-fitted mesh is given in Figure 3.18, where a representation of the tree data structure shows the three hierarchical levels created to store integration elements. For instance, the background element (hatched in Figure 3.18) intersects with both the material interface and the crack. The material interface first splits this element (added to the first level of the tree data structure) into three children elements, which are created at the second level of the hierarchy. Later, these elements intersect with the crack, and their corresponding integration elements are stored at the last level (leaves of the tree). More details about this example can be found in van den Boom *et al.* [29].

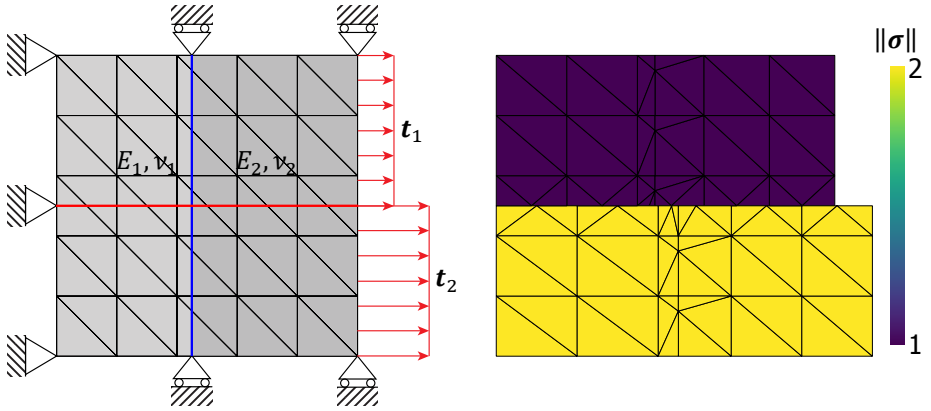



Figure 3.17: (a) A square domain with dimensions $2 \times 2$ includes both a material interface (marked with blue) and a crack (marked with red); horizontal tractions $\|\boldsymbol{t}_1\| = 1$ and $\|\boldsymbol{t}_2\| = 2$ are imposed on the right boundary below and above the crack, respectively. (b) Two independent kinematic fields with constant state of stress are generated in either side of the crack.

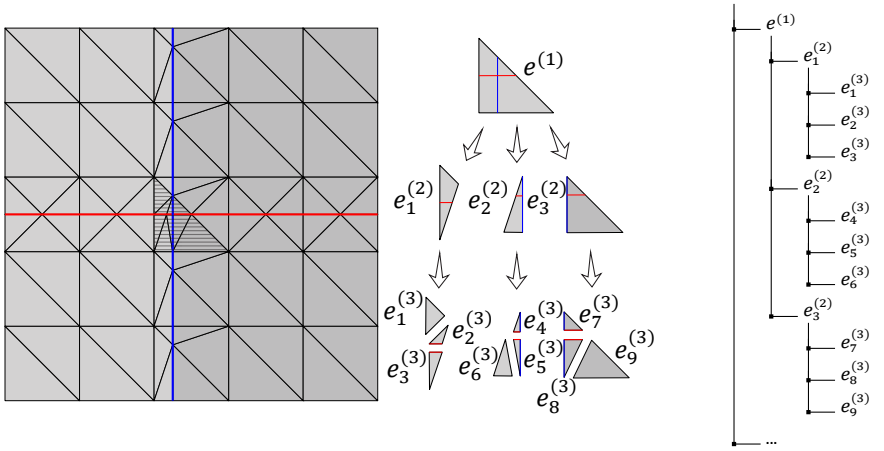
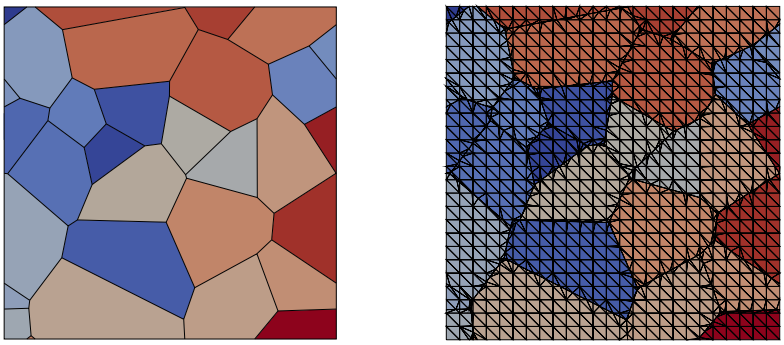



Figure 3.18: The new discretization created after interacting discontinuities with the background mesh, where a cut element (hatched) intersects with both material interface (marked with blue) and crack (marked with red); the corresponding children elements are created and stored into a hierarchical tree data structure with three levels.

### 3.4.3. Polycrystalline materials

In this example, subdividing the computational domain into grains is aggravated by the correct assignment of material properties. Here grains are represented explicitly with polygons in 2-D and polyhedra in 3-D. Polygons are represented by segments forming a closed surface. This is achieved by defining a proper nodal connectivity—similarly to the way we store the node connectivity of finite elements. Similarly, surface polygons are required to describe the corresponding polyhedra.

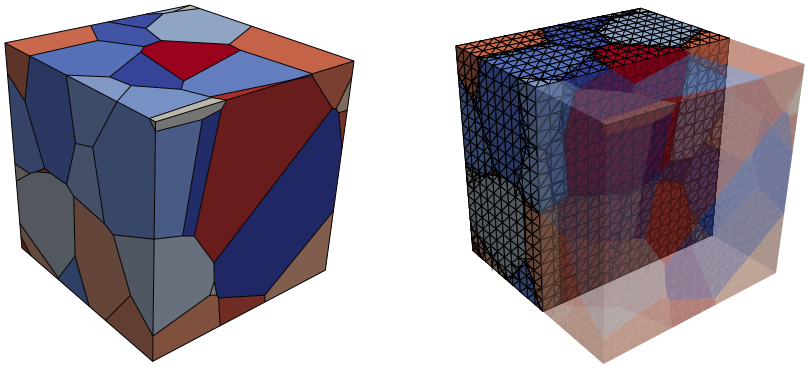


(a) (b)

Figure 3.19: (a) A 2-D polycrystalline model and (b) the corresponding finite element discretization after interacting with the geometric engine. The figure shows that the geometric engine is able to properly construct integration elements and that the right material properties are assigned (color-coded regions).

We showcase the geometric engine's capability to cope with polycrystalline models in 2-D and 3-D. The 2-D model shown in Figure 3.19(a) is composed of 30 polygonal grains with different material properties (every color corresponds to a different material), and a structured mesh of $25 \times 25 \times 2$ triangular elements is used. As apparent from Figure 3.19(b), the geometric engine is able to correctly assign the appropriate material properties to each grain. Obtaining the same results in 3-D is more challenging. Figure 3.20(a) shows 157 polyhedra used to describe the 3-D polycrystalline model, on a background mesh composed of $17 \times 17 \times 17 \times 6$ tetrahedral elements. The result of interacting the polycrystalline model with the background mesh in Figure 3.20(b) shows the proper creation of integration elements and assignment of material properties.





(a) (b)

Figure 3.20: (a) A 3-D polycrystalline model and (b) the corresponding finite element discretization after interacting with the geometric engine. The figure shows that the geometric engine is able to properly construct integration elements and that the right material properties are assigned (color-coded regions).

### 3.4.4. Immersing lower-dimensional manifolds

Here we demonstrate the geometric engine's ability to immerse a 2-D surface mesh (composed of triangular elements) into a 3-D tetrahedral mesh. To this end we use the well-known Stanford bunny in the computer graphics literature [110], which is discretized by a coarse surface mesh containing 1006 triangular elements (see Figure 3.21(a)). This surface mesh is then fully immersed into a structured mesh composed of $20 \times 20 \times 20 \times 6$ tetrahedral elements, as shown in Figure 3.21(b). A total of 3768 background elements are cut by the surface mesh (see Figure 3.21(c)), where 57 172 integration elements are created. The resulting new volumetric discretization of the Stanford bunny is composed of 2844 background and 28 066 integration elements, and the original surface mesh is split into 20 319 triangular elements (see Figure 3.21(d)).

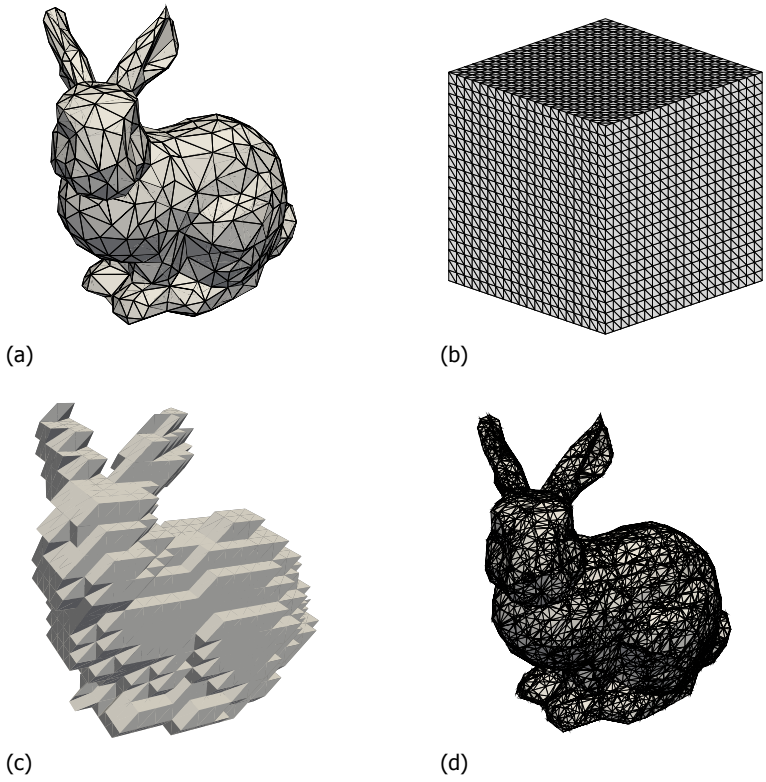



Figure 3.21: (a) The Stanford bunny surface mesh discretized by 1006 triangular elements; (b) The background mesh consisting of $20 \times 20 \times 20 \times 6$ tetrahedral elements; (c) 3768 tetrahedral elements intersected by the surface mesh; (d) The new volumetric discretization of the Stanford bunny model with 2844 background and 28 066 integration elements.

## 3.5. Summary and conclusions

In this work we proposed an object-oriented geometric engine especially designed for unfitted/immersed/enriched FEMs, which aims to interact the background mesh with discontinuities for generating a new discretization. Consider the representation of discontinuities, both implicit and explicit methods are implemented to describe their corresponding geometric configurations. The geometric engine design contains several main modules. `SpacePartitioner` creates a $k$-d tree data structure for partitioning the background mesh, which is used to identify elements enclosing a given point. `DualGraph` builds a graph structure targeting for finding neighbors of any given element. These two components are called in the rest for facilitating the geometric operations. `Interactor` focuses on creating enriched nodes at intersections between discontinuities and the background mesh via calling `Intersector` and integration elements within `ElementCreator` that are stored in a hierarchical data structure used for the assembly. Later, enriched nodes and integration ele-

ments are added to the original mesh data in `Mesh`, where new physical groups associated with discontinuities are created, for instance, when solving multi-phase problems. Together with the detailed explanation, the corresponding pseudo-code of each module was also provided. We investigated the computational complexity of the geometric engine through several 2-D and 3-D models including weak or strong discontinuity, where the behavior of the computational time associated with the number of background elements $|\mathcal{E}|$ is asymptotically approaching $\mathcal{O}\,|\mathcal{E}|^{1/2}$) (in 2-D) and $\mathcal{O}(|\mathcal{E}|^{2/3})$ (in 3-D)

Furthermore, several challenging numerical examples were used to investigate the performance of the geometric engine under IGFEM and DE-FEM. First, a 3-D topology optimization example for minimizing the structural compliance was proposed, where the level set function is used to describe the structural boundary and the structural analysis is performed by IGFEM. Next, a fracture junction problem was solved by DE-FEM, where several cracks intersect with each other within a single background element. Later, an example including intersected weak and strong discontinuities was proposed to show the hierarchical data structure that stores cut background elements and integration elements. Moreover, we incorporated polycrystalline materials that are represented as polygons in 2-D and polyhedra in 3-D into the background mesh, where new physical groups associated with the corresponding grains are created for assigning different material properties. Finally, the Stanford bunny model with complex geometry configurations, which is discretized as a surface mesh with 1006 triangular elements, is immersed into a 3-D background mesh, where the new discretization conforming to the surface of the Stanford bunny is generated.

A summary of the main findings of this work is listed in the following.

- Delaunay triangulation (tetrahedralization) behaves well for creating the corresponding integration elements in 2-D and 3-D. However, if only considering the background mesh with triangular or tetrahedral elements, it is also practical to set specific routines to construct integration elements. For instance, there are only a few scenarios when a triangle is split with a line segment completely. Although more cases should be taken into account for splitting a tetrahedron, it is still durable to implement the corresponding scheme to perform tetrahedralization. Although this alternative could increase the difficulty in the implementation and require more efforts to take all possible situations into account, the resulting geometric engine does not require any extra package based on Delaunay algorithm.

- Discontinuities can be handled in either hierarchical or sequential way within the geometric engine. For instance, when only one surface mesh is used to represent several cracks, where each surface element is considered as a crack, these cracks are handled sequentially as they are defined as one discontinuity. However, these cracks can be defined in the same amount of surface mesh that only contains one element, where each crack (surface mesh) is one discontinuity that is processed in a hierarchical way. Generally, the latter could create more integration/children elements, as children elements associ-

ated with previous discontinuities could be split by the rest cracks. Therefore, in terms of efficiency, the former method is recommended when handling discontinuous models with multiple cracks.

- As geometric operations between different geometric entities are performed to detect intersections and create integration elements, it is critical to set an appropriate value of tolerance/precision to keep the robustness of the proposed geometric engine. For example, intersections between background element edges and surface elements are detected via using the geometric algorithm based on their corresponding coordinates. The recommendation is to choose the value of tolerance related to the background/surface mesh size.

- Considering efficiency, when handling any cut background element, it is important to store the information about subdomains of cut element edges and/or faces, which can save the computational cost and avoid the precision issue related to check the location of intersections. On the one hand, this data can be directly used to create children of any background elements sharing these edges/faces. On the other hand, it is useful for handing discontinuities in a hierarchical way, as any subedges and/or subfaces created by the handled discontinuities could be split by the following ones.

## References


[1] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, 1st ed. (John Wiley & Sons, Ltd, 2009).

[2] Q. Du and D. Wang, *Recent progress in robust and quality delaunay mesh generation,* Journal of Computational and Applied Mathematics **195**, 8 (2006).

[3] D. Swenson and A. Ingraffea, *Modeling mixed-mode dynamic crack propagation nsing finite elements: theory and applications,* Computational Mechanics **3**, 381 (1988).

[4] P.-O. Bouchard, F. Bay, Y. Chastel, and I. Tovena, *Crack propagation modelling using an advanced remeshing technique,* Computer methods in applied mechanics and engineering **189**, 723 (2000).

[5] P.-O. Bouchard, F. Bay, and Y. Chastel, *Numerical modelling of crack propagation: automatic remeshing and comparison of different criteria,* Computer methods in applied mechanics and engineering **192**, 3887 (2003).

[6] G. Bugeda and J. Oliver, *A general methodology for structural shape optimization problems using automatic adaptive remeshing,* International Journal for Numerical Methods in Engineering **36**, 3161 (1993).

[7] Z. Liu and J. G. Korvink, *Adaptive moving mesh level set method for structure topology optimization,* Engineering Optimization **40**, 529 (2008).

[8] M. A. S. de Troya and D. A. Tortorelli, *Three-dimensional adaptive mesh refinement in stress-constrained topology optimization,* Structural and Multidisciplinary Optimization **62**, 2467 (2020).

[9] P. Saksono, W. Dettmer, and D. Perić, *An adaptive remeshing strategy for flows with moving boundaries and fluid–structure interaction,* International Journal for Numerical Methods in Engineering **71**, 1009 (2007).

[10] A. Masud, M. Bhanabhagvanwala, and R. A. Khurram, *An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction,* Computers & fluids **36**, 77 (2007).

[11] R. Vankadhar, D. Dandotiya, and B. Pal, *The study of elastic modulus and thermal conductivity of different fiber cross-section of fiber-reinforced composite in ansys, IOP Conference Series: Materials Science and Engineering*, **1013**, 012015 (2021).

[12] K. Riemselagh, J. Vierendeels, and E. Dick, *Two-dimensional incompressible navier-stokes calculations in complex-shaped moving domains,* Journal of engineering mathematics **34**, 57 (1998).

[13] L. Wu and D. Bogy, *Unstructured triangular mesh generation techniques and a finite volume numerical scheme for slider air bearing simulation with complex shaped rails,* IEEE transactions on magnetics **35**, 2421 (1999).

[14] N. Moës, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing,* International Journal for Numerical Methods in Engineering **46**, 131 (1999).

[15] C. Duarte, O. Hamzeh, T. Liszka, and W. Tworzydlo, *A generalized finite element method for the simulation of three-dimensional dynamic crack propagation,* Computer methods in applied mechanics and engineering **190**, 2227 (2001).

[16] J. Melenk and I. Babuška, *The partition of unity finite element method: Basic theory and applications,* Comput. Methods Appl. Mech. Eng. **139**, 289 (1996).

[17] I. Babuška and J. Melenk, *The partition of unity method,* Int. J. Numer. Methods Eng. **40**, 727 (1997).

[18] J. Chessa, H. Wang, and T. Belytschko, *On the construction of blending elements for local partition of unity enriched finite elements,* International Journal for Numerical Methods in Engineering **57**, 1015 (2003).

[19] T. P. Fries, *A corrected xfem approximation without problems in blending elements,* International Journal for Numerical Methods in Engineering **75**, 503 (2008).

[20] I. Babuška, U. Banerjee, and J. E. Osborn, *Survey of meshless and generalized finite element methods: A unified approach,* Acta Numerica **12**, 1 (2003).

[21] N. Moës, E. Béchet, and M. Tourbier, *Imposing Dirichlet boundary conditions in the extended finite element method,* International Journal for Numerical Methods in Engineering **67**, 1641 (2006).

[22] I. Babuška and U. Banerjee, *Stable Generalized Finite Element method (SGFEM),* Computer Methods in Applied Mechanics and Engineering **201–204**, 91 (2012).

[23] V. Gupta, C. A. Duarte, I. Babuška, and U. Banerjee, *A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **266**, 23 (2013).

[24] K. Kergrene, I. Babuška, and U. Banerjee, *Stable generalized finite element method and associated iterative schemes; application to interface problems,* Computer Methods in Applied Mechanics and Engineering **305**, 1 (2016).

[25] S. Soghrati, A. M. Aragón, C. Armando Duarte, and P. Geubelle, *An interface-enriched generalized FEM for problems with discontinuous gradient fields,* Int. J. Numer. Meth. Eng. **89**, 991 (2012).

[26] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *An interface-enriched generalized finite element method for level set-based topology optimization,* Structural and Multidisciplinary Optimization (2020).

[27] S. Soghrati, *Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations,* Journal of Computational Physics **275**, 41 (2014).

[28] A. M. Aragón, B. Liang, H. Ahmadian, and S. Soghrati, *On the stability and interpolating properties of the hierarchical interface-enriched finite element method,* Computer Methods in Applied Mechanics and Engineering **362**, 112671 (2020).

[29] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions,* International Journal for Numerical Methods in Engineering **120**, 1163 (2019).

[30] A. Cuba-Ramos, A. M. Aragón, S. Soghrati, P. H. Geubelle, and J. Molinari, *A new formulation for imposing dirichlet boundary conditions on non-matching meshes,* International Journal for Numerical Methods in Engineering **103**, 430 (2015).

[31] S. van den Boom, F. van Keulen, and A. Aragón, *Fully decoupling geometry from discretization in the bloch–floquet finite element analysis of phononic*

*crystals,* Computer Methods in Applied Mechanics and Engineering **382**, 113848 (2021).

[32] D. Liu, S. J. van den Boom, A. Simone, and A. Aragón, *An interface-enriched generalized finite element formulation for locking-free coupling of non-conforming discretizations and contact,* Submitted (2022).

[33] J. Zhang, F. van Keulen, and A. M. Aragón, *On tailoring fracture resistance of brittle structures: A level set interface-enriched topology optimization approach,* Computer Methods in Applied Mechanics and Engineering **388**, 114189 (2022).

[34] A. Aragón and A. Simone, *The discontinuity-enriched finite element method,* International Journal for Numerical Methods in Engineering **112**, 1589 (2017), cited By 0.

[35] J. Zhang, S. J. van den Boom, F. van Keulen, and A. M. Aragón, *A stable discontinuity-enriched finite element method for 3-d problems containing weak and strong discontinuities,* Computer Methods in Applied Mechanics and Engineering **355**, 1097 (2019).

[36] E. De Lazzari, S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A critical view on the use of non-uniform rational b-splines to improve geometry representation in enriched finite element methods,* International Journal for Numerical Methods in Engineering **122**, 1195 (2021).

[37] F. A. Sabet, A. Raeisi Najafi, E. Hamed, and I. Jasiuk, *Modelling of bone fracture and strength at different length scales: a review,* Interface focus **6**, 20150055 (2016).

[38] S. Erden and K. Ho, *3 - fiber reinforced composites,* in *Fiber Technology for Fiber-Reinforced Composites*, Woodhead Publishing Series in Composites Science and Engineering, edited by M. Ö. Seydibeyoğlu, A. K. Mohanty, and M. Misra (Woodhead Publishing, 2017) pp. 51 – 79.

[39] N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko, *Modeling holes and inclusions by level sets in the extended finite-element method,* Computer methods in applied mechanics and engineering **190**, 6183 (2001).

[40] B. Tranquart, P. Ladevèze, E. Baranger, and A. Mouret, *A computational approach for handling complex composite microstructures,* Composite Structures **94**, 2097 (2012).

[41] S. Soghrati and P. H. Geubelle, *A 3d interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries,* Computer Methods in Applied Mechanics and Engineering **217**, 46 (2012).

[42] A. Simone, C. A. Duarte, and E. Van der Giessen, *A generalized finite element method for polycrystals with discontinuous grain boundaries,* International Journal for Numerical Methods in Engineering **67**, 1122 (2006).

[43] D. Liu, J. Zhang, A. Aragón, and A. Simone, *The discontinuity-enriched finite element method for multiple intersecting discontinuities,* International Journal for Numerical Methods in Engineering .

[44] J. Sietsma, *Nucleation and growth during the austenite-to-ferrite phase transformation in steels after plastic deformation,* in *Phase Transformations in Steels* (Elsevier, 2012) pp. 505–526.

[45] J. Shen, *Advanced ceramics for dentistry* (Butterworth-Heinemann, 2013).

[46] D. Yi, Y. Wang, O. M. van 't Erve, L. Xu, H. Yuan, M. J. Veit, P. P. Balakrishnan, Y. Choi, A. T. N' Diaye, P. Shafer, *et al.*, *Emergent electric field control of phase transformation in oxide superlattices,* Nature communications **11**, 1 (2020).

[47] Y. Ruan, J. C. Liu, and O. Richmond, *A deforming finite element method for analysis of alloy solidification problems,* Finite elements in analysis and design **13**, 49 (1993).

[48] H. Ji, D. Chopp, and J. Dolbow, *A hybrid extended finite element/level set method for modeling phase transformations,* International Journal for Numerical Methods in Engineering **54**, 1209 (2002).

[49] R. Merle and J. Dolbow, *Solving thermal and phase change problems with the extended finite element method,* Computational mechanics **28**, 339 (2002).

[50] A. Cosimo, V. Fachinotti, and A. Cardona, *An enrichment scheme for solidification problems,* Computational Mechanics **52**, 17 (2013).

[51] P. Stapór, *The xfem for nonlinear thermal and phase change problems,* International Journal of Numerical Methods for Heat & Fluid Flow (2015).

[52] M. Li, H. Chaouki, J.-L. Robert, D. Ziegler, D. Martin, and M. Fafard, *Numerical simulation of stefan problem with ensuing melt flow through xfem/level set method,* Finite Elements in Analysis and Design **148**, 13 (2018).

[53] A. Faghri and Y. Zhang, *Transport phenomena in multiphase systems* (Elsevier, 2006).

[54] L. Pan, S. W. Webb, and C. M. Oldenburg, *Analytical solution for two-phase flow in a wellbore using the drift-flux model,* Advances in Water Resources **34**, 1656 (2011).

[55] G. B. Wallis, *One-dimensional two-phase flow* (Dover Publications, 2020).

[56] M. Massoud, *Engineering thermofluids. thermodynamics, fluid mechanics, and heat transfer; 2007. isbn 10 3-540-22292-8,* (2005).

[57] J. Chessa and T. Belytschko, *An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension,* International journal for numerical methods in engineering **58**, 2041 (2003).

[58] J. Chessa and T. Belytschko, *An extended finite element method for two-phase fluids,* J. Appl. Mech. **70**, 10 (2003).

[59] H. Sauerland and T.-P. Fries, *The stable xfem for two-phase flows,* Computers & Fluids **87**, 41 (2013).

[60] Z. Zhuang, Z. Liu, B. Cheng, and J. Liao, *Extended Finite Element Method: Tsinghua University Press Computational Mechanics Series* (Academic Press, 2014).

[61] O. Zienkiewicz, R. Taylor, and P. Nithiarasu, *Chapter 13 - fluid–structure interaction,* in *The Finite Element Method for Fluid Dynamics (Seventh Edition)*, edited by O. Zienkiewicz, R. Taylor, and P. Nithiarasu (Butterworth-Heinemann, Oxford, 2014) seventh edition ed., pp. 423 – 449.

[62] N. Takashi, *Ale finite element computations of fluid-structure interaction problems,* Computer methods in applied mechanics and engineering **112**, 291 (1994).

[63] E. Kuhl, S. Hulshoff, and R. De Borst, *An arbitrary lagrangian eulerian finite-element approach for fluid–structure interaction phenomena,* International journal for numerical methods in engineering **57**, 117 (2003).

[64] D. Rubenstein, W. Yin, and M. D. Frame, *Biofluid mechanics: an introduction to fluid mechanics, macrocirculation, and microcirculation* (Academic Press, 2015).

[65] A. Gerstenberger and W. A. Wall, *An extended finite element method/lagrange multiplier based approach for fluid–structure interaction,* Computer Methods in Applied Mechanics and Engineering **197**, 1699 (2008).

[66] A. Gerstenberger and W. A. Wall, *Enhancement of fixed-grid methods towards complex fluid–structure interaction applications,* International Journal for Numerical Methods in Fluids **57**, 1227 (2008).

[67] N. Jenkins and K. Maute, *Level set topology optimization of stationary fluid-structure interaction problems,* Structural and Multidisciplinary Optimization **52**, 179 (2015).

[68] W. J. Cantwell and J. Morton, *The significance of damage and defects and their detection in composite materials: A review,* The Journal of Strain Analysis for Engineering Design **27**, 29 (1992).

[69] M. Naebe, M. M. Abolhasani, H. Khayyam, A. Amini, and B. Fox, *Crack damage in polymers and composites: A review,* Polymer Reviews **56**, 31 (2016).

[70] N. Sukumar, N. Moës, B. Moran, and T. Belytschko, *Extended finite element method for three-dimensional crack modelling,* International journal for numerical methods in engineering **48**, 1549 (2000).

[71] N. Moës and T. Belytschko, *Extended finite element method for cohesive crack growth,* Engineering fracture mechanics **69**, 813 (2002).

[72] J. V. Cox, *An extended finite element method with analytical enrichment for cohesive crack modeling,* International Journal for Numerical Methods in Engineering **78**, 48 (2009).

[73] D. Hull and D. J. Bacon, *Introduction to dislocations* (Butterworth-Heinemann, 2011).

[74] P. Lagerlof, *Crystal dislocations: Their impact on physical properties of crystals,* (2018).

[75] G. Ventura, B. Moran, and T. Belytschko, *Dislocations by partition of unity,* International Journal for Numerical Methods in Engineering **62**, 1463 (2005).

[76] R. Gracie, G. Ventura, and T. Belytschko, *A new fast finite element method for dislocations based on interior discontinuities,* International Journal for Numerical Methods in Engineering **69**, 423 (2007).

[77] J. Robbins and T. E. Voth, *Modeling dislocations in a polycrystal using the generalized finite element method,* International Journal of Theoretical and Applied Multiscale Mechanics **2**, 95 (2011).

[78] D. Shuttle and I. Smith, *Numerical simulation of shear band formation in soils,* International journal for numerical and analytical methods in geomechanics **12**, 611 (1988).

[79] J. Rudnicki and J. Rice, *Conditions for the localization of deformation in pressure-sensitive dilatant materials,* Journal of the Mechanics and Physics of Solids **23**, 371 (1975).

[80] M. A. Meyers, V. F. Nesterenko, J. C. LaSalvia, and Q. Xue, *Shear localization in dynamic deformation of materials: microstructural evolution and self-organization,* Materials Science and Engineering: A **317**, 204 (2001).

[81] F. J. Humphreys and M. Hatherly, *Recrystallization and related annealing phenomena* (Elsevier, 2012).

[82] J. J. Gilman, *Micromechanics of shear banding,* Mechanics of materials **17**, 83 (1994).

[83] J. Desrues, J. Lanier, and P. Stutz, *Localization of the deformation in tests on sand sample,* Engineering Fracture Mechanics **21**, 909 (1985).

[84] E. Samaniego and T. Belytschko, *Continuum–discontinuum modelling of shear bands,* International Journal for Numerical Methods in Engineering **62**, 1857 (2005).

[85] E. Mikaeili and P. Liu, *Numerical modeling of shear band propagation in porous plastic dilatant materials by xfem,* Theoretical and Applied Fracture Mechanics **95**, 164 (2018).

[86] P. M. Areias and T. Belytschko, *Two-scale shear band evolution by local partition of unity,* International Journal for Numerical Methods in Engineering **66**, 878 (2006).

[87] P. M. Areias and T. Belytschko, *Two-scale method for shear bands: thermal effects and variable bandwidth,* International Journal for Numerical Methods in Engineering **72**, 658 (2007).

[88] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr, *On the design of cgal a computational geometry algorithms library,* Software: Practice and Experience **30**, 1167 (2000).

[89] G. Ren and R. Younis, *A coupled xfem-edfm numerical model for hydraulic fracture propagation,* in *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery*, 1 (European Association of Geoscientists & Engineers, 2018) pp. 1–16.

[90] A. Massing, M. G. Larson, and A. Logg, *Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions,* SIAM Journal on Scientific Computing **35**, C23 (2013).

[91] S. Popinet, *The gnu triangulated surface library,* (2004).

[92] N. Sukumar and J.-H. Prévost, *Modeling quasi-static crack growth with the extended finite element method part i: Computer implementation,* International journal of solids and structures **40**, 7513 (2003).

[93] J. Pereira, C. A. Duarte, D. Guoy, and X. Jiao, *hp-generalized fem and crack surface representation for non-planar 3-d cracks,* International Journal for Numerical Methods in Engineering **77**, 601 (2009).

[94] S. Soghrati and H. Ahmadian, *3d hierarchical interface-enriched finite element method: implementation and applications,* Journal of Computational Physics **299**, 45 (2015).

[95] J. Březina and P. Exner, *Fast algorithms for intersection of non-matching grids using plücker coordinates,* Computers & Mathematics with Applications **74**, 174 (2017).

[96] M. Yang, A. Nagarajan, B. Liang, and S. Soghrati, *New algorithms for virtual reconstruction of heterogeneous microstructures,* Computer Methods in Applied Mechanics and Engineering **338**, 275 (2018).

[97] J. R. Shewchuk, *Triangle: Engineering a 2d quality mesh generator and delaunay triangulator,* in *Workshop on Applied Computational Geometry* (Springer, 1996) pp. 203–222.

[98] H. Si, *Tetgen, a delaunay-based quality tetrahedral mesh generator,* ACM Transactions on Mathematical Software (TOMS) **41**, 1 (2015).

[99] C. Geuzaine and J.-F. Remacle, *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities,* International Journal for Numerical Methods in Engineering **79**, 1309 (2009).

[100] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,* Journal of computational physics **79**, 12 (1988).

[101] J. A. Sethian and P. Smereka, *Level set methods for fluid interfaces,* Annual review of fluid mechanics **35**, 341 (2003).

[102] S. Osher and N. Paragios, *Geometric level set methods in imaging, vision, and graphics* (Springer Science & Business Media, 2003).

[103] M. Y. Wang, X. Wang, and D. Guo, *A level set method for structural topology optimization,* Computer methods in applied mechanics and engineering **192**, 227 (2003).

[104] G. Allaire, F. Jouve, and A.-M. Toader, *Structural optimization using sensitivity analysis and a level-set method,* Journal of computational physics **194**, 363 (2004).

[105] I.-L. Chern and Y.-C. Shu, *A coupling interface method for elliptic interface problems,* Journal of Computational Physics **225**, 2138 (2007).

[106] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,* Advances in computational Mathematics **4**, 389 (1995).

[107] S. Wang and M. Y. Wang, *Radial basis functions and level set method for structural topology optimization,* International journal for numerical methods in engineering **65**, 2060 (2006).

[108] L. P. Chew, *Constrained delaunay triangulations,* Algorithmica **4**, 97 (1989).

[109] H. Si, *A quality tetrahedral mesh generator and three-dimensional delaunay triangulator,* Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany **81** (2006).

[110] G. Turk and M. Levoy, *Zippered polygon meshes from range images,* in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94 (ACM, New York, NY, USA, 1994) pp. 311–318.

# 4

# An Improved Stress Recovery Technique for the Unfitted Finite Element Analysis of Discontinuous Gradient Fields

*Stress analysis is an all-pervasive practice in the engineering design process. With displacement-based finite element analysis (FEA), stress fields are obtained in a post-processing step by computing the gradient of the displacement field—therefore less accurate. In enriched finite element analysis (EFEA), which provides unprecedented versatility by decoupling the finite element mesh from material interfaces, cracks, and structural boundaries, stress recovery is further aggravated when such discontinuities get arbitrarily close to nodes of the mesh; the presence of bad element aspect ratios and/or tiny integration elements often leads to stress overestimation, which could have a detrimental impact on nonlinear analyses (e.g., damage or plasticity) since stress concentrations are just a non-physical numerical artifact. In this work we propose a stress recovery procedure for enhancing the stress field in problems containing weak discontinuities—i.e., where the field gradient is discontinuous. The formulation is based on a stress improvement procedure (SIP) initially proposed for low-order finite elements with the standard FEA. Although generally applicable to all EFEA, we investigate the technique in the context of the Interface-enriched Generalized Finite Element Method (IGFEM). We compare the procedure to other post-processing techniques that are used to smoothen the stress field; we demonstrate that the modified SIP for EFEA provides an enhanced stress field that is more accurate than that obtained by directly applying the gradient to the primal field, even for standard FEM with fitted meshes.*

This chapter has been published in *International Journal for Numerical Methods in Engineering,* doi:10.1002/nme.6825.

## 4.1. Introduction

Standard finite element analysis (FEA) is today considered the standard procedure for solving boundary value problems in solid mechanics. However, the modeling of problems containing material interfaces and/or cracks requires *fitted* or *matching* discretizations, where finite element (FE) edges are aligned to such discontinuities [1]. The creation of such meshes can prove fairly challenging for complex geometric configurations, particularly in 3-D [2]. Furthermore, problems where discontinuities are not known *a priori* but evolve throughout the simulation, such as topology optimization, mandate for a different analysis procedure because remeshing is not only prone to fail but also increases computational costs [3]. Enriched finite element analysis (EFEA) [4–9] provides an elegant solution to the use of fitted meshes by decoupling discontinuities from the FE mesh. In these methods, which have become fairly popular in the computational mechanics community because of their versatility and the possibility of using simple (usually structured) FE meshes, the standard FE approximation is extended or enhanced by means of an enriched FE space that properly resolves the discontinuities' kinematics—*e.g.*, via a Heaviside function for cracks (strong discontinuities) [5] or distance-based functions for material interfaces (weak discontinuities) [10]. However, even though complete discontinuity-mesh decoupling is attained, the accuracy of gradient fields greatly depends on the resulting enriched FE space, which may be severely affected by the way elements are cut by discontinuities.

It is well known that finite elements with bad aspect ratios are prone to significant errors in gradient fields [11]. But even when meshes are properly constructed, stress fields are usually obtained by taking the gradient of the primal field—therefore less accurate—and are $C^{-1}$-continuous at element edges. Consequently, recovery/smoothing techniques that post-process the primal field have been proposed to improve the accuracy of the stress field. Improving gradient fields was first conducted by Brauchli [12], who obtained a consistent stress field based on the theory of conjugate approximations [13]; because the approximation functions used to interpolate stresses are expressed as a linear combination of linearly independent functions defined over the entire domain, the resulting stress field is continuous across element boundaries. Hinton and Campbell [14] later proposed local and global smoothing techniques based on a least squares procedure, whereby enhanced stresses are obtained by minimizing their difference with respect to directly-calculated stresses. The superconvergent patch recovery (SPR) by Zienkiewicz and Zhu [15, 16] was introduced afterwards as a more computationally efficient method that uses an element patch surrounding the nodes where stress is sought; in this technique, the same shape functions used to interpolate the displacement field are used to construct a smooth stress field, whose polynomial form is obtained by solving a least squares problem at superconvergent points (locations where stress accuracy is the highest within the element). Subsequently, Wiberg and Abdulwahab [17] proposed an enhanced version of SPR by taking the equilibrium equation into account (SPRE), where the residual of the governing equilibrium equation obtained with the smooth stress interpolation is considered in the least squares procedure. However, the quality of recovered stresses at boundaries is

worse than that obtained in the interior. Wiberg *et al.* [18] further proposed an improved extension of SPR considering both the equilibrium equation and boundary conditions (SPREB), where the recovered stress field satisfies the appropriate prescribed boundary conditions by adding the corresponding residuals to the least squares fit. A similar idea was also introduced by Blacker and Belytschko [19], who only focused on natural boundary conditions. Later, Ródenas *et al.* [20] proposed an improvement of SPR that uses constraint equations (SPR-C); this method shares similarities with SPREB, where stress interpolation polynomials are required to satisfy, in the local patch, equilibrium and compatibility equations, and boundary conditions. With several modifications, this technique was extended to recover the stress field in singular elasticity problems by decomposing the primal field in smooth and non-smooth components [21, 22]; while recovering the smooth part follows the standard SPR-C procedure, an expression based on stress intensity factors (SIFs) that describes the asymptotic fields near the crack tip is used to recover the singular part.



Even though SPR-based techniques perform well in recovering gradient fields, they rely on the existence of superconvergent points; these points are not always defined, for instance in quadratic triangular elements [23]. Therefore, several post-processing recovery methods that do not rely on superconvergent points were proposed. Tabbara *et al.* [24] used a moving least square (MLS) method to construct a local interpolation of the displacement field, where the recovered strain field is obtained by taking the gradient of the displacement polynomial interpolants. Later, the recovery by equilibrium in patches (REP) [25, 26] was proposed to extract a stress field by satisfying equilibrium equations in a weak sense over patches of elements; a smooth stress field is obtained by using a least squares scheme, where the error between enhanced and directly-calculated stresses is projected onto the finite element strain space over the patch. Instead of minimizing the difference between recovered and directly-calculated stresses over the patch, Ubertini proposed another recovery technique by considering compatibility in patches (RCP) [27], whereby the strain error (difference between the directly-calculated and enhanced strains) is orthogonal to the space of enhanced stresses over each patch. Rather than using the support of a node (*i.e.*, union of elements sharing it) as the patch [27], Benedetti *et al.* [28] further developed RCP by considering patches composed of a main target element and all its surrounding elements; once the enhanced stress interpolant is obtained over the patch, the stress in the target element is recovered, and without considering results from adjacent patches. Later, Payen and Bathe [29] proposed a stress improvement procedure (SIP) derived from a mixed formulation based on the Hu-Washizu principle, which is effective in obtaining accurate stress predictions. A mixed formulation is also the only option for getting optimal finite element discretizations when solving problems including (almost) incompressible media, thin structures, or multiphysics phenomena (see [29] and references therein). Within SIP, the point-wise relationship between stress and strain is relaxed; two equations are used to obtain the recovered stress field, one that requires the stress error (difference between enhanced and directly-calculated stresses) is orthogonal to the space of self-equilibrated stresses, and another one that enforces equilibrium over

the element patch (in weak sense) with the enhanced stress field. This recovery procedure has several advantages over its predecessors: Unlike RCP, this technique does not use a particular stress field defined *a priori* that satisfies equilibrium within the patch. Moreover, the number of basic equations used is independent of the number and type of elements considered in the stress calculation domain. Finally, as a limitation of using low-order elements to model nearly incompressible materials, a spurious checkerboard pattern of stresses can be avoided by using this recovery technique [30]. Recently, Sharma *et al.* [31] extended this method to recover the stress field in 3D low-order finite element meshes. The preceding procedures were not proposed for problems with discontinuities that exhibit singular stress fields. Xiao and Karihaloo [32] introduced another recovery approach for fracture problems, which they named the statically admissible stress (SAR) recovery scheme; this method, which is based on complete polynomial functions that satisfy the equilibrium equation and traction boundary conditions in each patch, uses MLS to enforce stress continuity between patches; unlike SPR-C, the smooth and singular stress fields are handled together, where the original stress field near the crack tip is obtained by using hybrid crack elements [33].



Regarding EFEA, many stress recovery procedures have been proposed in the context of the extended/generalized finite element method (X/GFEM). The SAR technique, which was designed to recover the stress field in fracture problems with standard FEM, was later extended to improve the accuracy of the stress approximation in X/GFEM [34]. Duflot and Bordas [35] used the global derivative recovery method—which requires the entire model as the calculation domain—to smooth the strain field for linear elastic fracture mechanics problems; the procedure works by minimizing the square of the $L^2$-norm of the difference between the directly-calculated and smoothened strain fields, the latter consisting of three parts (*i.e.*, a smooth component interpolated by standard shape functions, a non-smooth discontinuous part enriched by Heaviside function, and an enrichment used to capture the singularity near the crack tip). Jin *et al.* [36] used this technique to quantify the interpolation error, a measure that is then used to drive mesh adaptivity. Later, the technique was used to recover the strain field for problems that contain only weak discontinuities, with an *a posteriori* error estimate based on the recovered strain that is used for adaptive local mesh refinement [37, 38]. As an alternative to the global derivative recovery method, a derivative recovery procedure based on extended moving least squares (XMLS) was proposed to obtain a smooth strain field [39, 40]; a smoothened displacement field is first constructed with the original nodal displacements and MLS shape functions, where near-tip asymptotic functions are added to capture singularities; the recovered strains are then obtained by applying the gradients to this new displacement field. SPR was also adapted to X/GFEM (SPR-XFEM) [41] with three major modifications: direct calculation of recovered stresses at integration points (not mesh nodes), decomposing the stress field into singular and smooth parts, and using different stress interpolation polynomials at each side of the crack [19]. Later, this technique was used to evaluate upper error bounds [42] and to create a recovery based goal-oriented error estimator for XFEM approximations [43]. While these works were modifications/extensions of methods

proposed for standard FEM, other procedures were also introduced exclusively for X/GFEM. Prange *et al.* [44] developed a recovery procedure based on a global least squares projection of raw stresses calculated from X/GFEM approximations in fracture problems with arbitrarily distributed cracks and inelastic material behavior. Lins *et al.* [45] later adapted this approach to improve the stress field within stable GFEM (SGFEM); because SGFEM modifies the X/GFEM enrichment functions to solve the issue of ill-conditioned stiffness matrices [46–48], the enrichment functions used to interpolate the recovered stress field are also modified. Instead of using a global least squares projection, Lins *et al.* [49] proposed a more computationally efficient recovery technique, whereby a consistent block-diagonal projection operator [50] is used to perform a locally weighted least squares projection of directly-calculated stresses over patches of elements. Sharma and Maute [51] introduced the ghost penalty method [52] into X/GFEM for penalizing the jump in displacement gradients across element boundaries near material interfaces; this method is later combined with a smoothing technique defined over the entire domain to obtain a smooth stress field.



Contrary to X/GFEM, where enrichments are associated to nodes of the original FE discretization, interface- and discontinuity-enriched finite element procedures seek to enhance the approximation by associating enrichments to nodes that are created directly along discontinuities. The Interface-enriched Generalized Finite Element Method (IGFEM) [7], the Hierarchical Interface-enriched Finite Element Method (HIFEM) [53, 54], and the Discontinuity-Enriched Finite Element Method (DE-FEM) [8, 9, 55] thus provide an alternative enriched approach for solving problems with discontinuities. These procedures decouple the background mesh from discontinuities as in X/GFEM, but they do not have many of the latter's drawbacks: By creating enriched DOFs at the intersections between the background mesh and discontinuities, these EFEA techniques are endowed with intrinsic stability with regard to the condition number of stiffness matrices [54], no loss of accuracy in blending elements (neighboring to cut elements) due to the locality of enrichment functions [8], straightforward enforcement of non-homogeneous essential boundary conditions [56], and smooth recovered traction profiles from Dirichlet boundaries [56, 57]. Nevertheless, because these methods recover the standard FEM space by means of an enrichment, they also suffer from inaccurate stress fields caused by sliver elements; it has been reported that stresses can be overestimated more than 150% [58].

In this paper we propose a stress recovery technique for EFEA based on SIP, which as stated above has several advantages over other recovery techniques. SIP has been shown to enhance stress fields from displacement-based FEM results, with higher convergence rates than those of directly-calculated stresses. While the method is generally applicable and thus could also be used with X/GFEM, we demonstrate the procedure on weak discontinuities resolved by IGFEM. When solving multiphase material problems, only elements with the same material properties are considered when constructing the patch of elements for stress recovery, as initially suggested by Payen and Bathe [29]. The performance of the proposed method is studied by means of three numerical examples. With Eshelby's inclusion problem

we explore the accuracy of stress fields obtained using standard FEM with fitted meshes and IGFEM with unfitted meshes. The recovery technique coupled to IGFEM provides more accurate stresses, for the same mesh size, than directly-calculated stresses obtained by standard FEM on fitted meshes. The second example studies an elliptical cavity in an infinite plate under remote loading, where we first adjust the ratio between the elliptical axes. The corresponding results show that the proposed recovery technique generally provides more accurate approximations than the direct calculation. Later, we consider a circular cavity under the same boundary condition, where the radius of the circle is adjusted to create different enriched discretizations for the same structured background mesh. In addition to evaluating elemental stresses, we also calculate nodal stresses obtained via the proposed method and other smoothing procedures based on directly-calculated stresses. The proposed stress recovery technique performs best for evaluating the stress concentration factor $K_t$, which is the ratio of the highest stress to the nominal far field stress. Furthermore, a convergence study shows the reliability of the proposed recovery technique with regard to mesh size. Finally, a pressurized sphere example demonstrates that the proposed recovery technique can also avoid overestimated stresses in tiny integration elements in 3-D.



## 4.2. Formulation

### 4.2.1. IGFEM-based analysis

Consider an open domain $\Omega \subset \mathbb{R}^d$ with closure $\overline{\Omega}$ that is composed of two disjoint regions $\Omega_1$ and $\Omega_2$, which are occupied by different phases. For simplicity, the 2-D case is shown in Figure 4.1. We denote by $\boldsymbol{u}$ the displacement field and by $\boldsymbol{u}_i$ its restriction to the $i$th domain $\Omega_i$, *i.e.*, $\boldsymbol{u_i} = \boldsymbol{u}|_{\Omega_i}$. A similar notation is used for other subscripted quantities. The domain's boundary $\Gamma \equiv \partial\Omega = \overline{\Omega} \setminus \Omega$ is smooth and consists of two disjoint regions $\Gamma_N$ and $\Gamma_D$, where surface tractions $\bar{\boldsymbol{t}}$ and displacements $\bar{\boldsymbol{u}}$ are prescribed, respectively. We denote by $\boldsymbol{n}$ the outward unit normal to the boundary $\partial\Omega$. The material interface between domains $\Omega_1$ and $\Omega_2$ is denoted as $\Gamma_{12}$ and has unit outward normal $\boldsymbol{n}_{12}$.

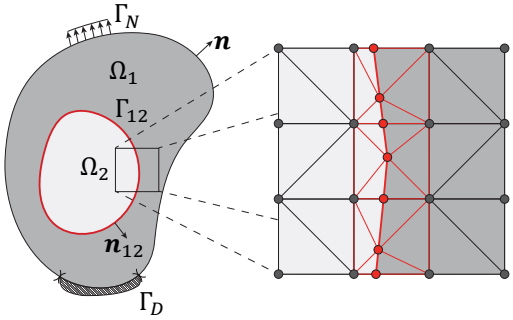



Figure 4.1: A domain $\Omega$ consists of two parts $\Omega_1$ and $\Omega_2$ with a smooth boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$. Dirichlet boundary conditions are prescribed on $\Gamma_D$ and surface tractions on $\Gamma_N$. For the discretized model, enriched nodes (marked with red) are created at the intersections between material interfaces (red segments) and element edges; integration elements (red triangles) are also created near the interface.

The strong form of the elastostatics boundary value problem is: Find the displacement field $\boldsymbol{u}$ such that

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}_i + \boldsymbol{b}_i &= 0 && \text{in } \Omega_i, i = 1,2 \\ \boldsymbol{u} &= \bar{\boldsymbol{u}} && \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} &= \bar{\boldsymbol{t}} && \text{on } \Gamma_N, \end{aligned} \tag{4.1}$$

with interface conditions

$$\begin{aligned} \boldsymbol{u}_1 &= \boldsymbol{u}_2 && \text{on } \Gamma_{12}, \\ \boldsymbol{\sigma}_1 \cdot \boldsymbol{n}_{12} &= \boldsymbol{\sigma}_2 \cdot \boldsymbol{n}_{12} && \text{on } \Gamma_{12}, \end{aligned}$$

where $\nabla\cdot$ denotes the divergence operator, $\boldsymbol{\sigma}$ Cauchy's stress tensor, and $\boldsymbol{b}$ the body force. We assume a linear elastic material behavior and thus $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u})$, where $\mathbf{C}$ is the elastic modulus tensor, and $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\intercal}\right)$ the infinitesimal strain tensor. The weak form of the boundary value problem is: Find $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}(\Omega)$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}) \qquad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}(\Omega), \tag{4.2}$$

where $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ are vector-valued function sets defined as

$$\boldsymbol{\mathcal{U}}(\Omega) = \left\{ \boldsymbol{u} \in \mathbb{R}^d \middle| \ \boldsymbol{u} \in \left[\mathcal{L}^2(\Omega)\right]^d, \boldsymbol{u}|_{\Omega_i} \in \left[\mathcal{H}^1(\Omega_i)\right]^d \right\}, \tag{4.3}$$

$$\boldsymbol{\mathcal{V}}(\Omega) = \left\{ \boldsymbol{v} \in \boldsymbol{\mathcal{U}}(\Omega), \ \boldsymbol{v}|_{\Gamma_D} = 0 \right\}, \tag{4.4}$$

with $\mathcal{L}^2$ denoting the Hilbert space of square-integrable functions and $\mathcal{H}^1$ the first-order Sobolev space, respectively. The bilinear $a(\boldsymbol{u}, \boldsymbol{v})$ and linear $\ell(\boldsymbol{v})$ forms are given by

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \ \mathrm{d}\Omega, \tag{4.5}$$

and

$$\ell(\boldsymbol{v}) = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{b} \ \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{v} \cdot \bar{\boldsymbol{t}} \ \mathrm{d}\Gamma. \tag{4.6}$$

In order to solve the problem above, $\Omega$ is discretized by a finite element mesh—not necessarily fitting to material interfaces—such that the the discretized domain $\overline{\Omega}^h = \cup_{i=1}^E \overline{e}_i$, with $e_i$ denoting the $i$th finite element and $E$ the total number of elements. Then, the finite-dimensional form of Equation (4.2) is expressed as

$$\sum_{i=1}^{E} \int_{e_i} \boldsymbol{\epsilon}(\boldsymbol{v}^h) : \mathbf{C} : \boldsymbol{\epsilon}(\boldsymbol{u}^h) \ \mathrm{d}\Omega = \sum_{i=1}^{E} \int_{e_i} \boldsymbol{v}^h \cdot \boldsymbol{b} \ \mathrm{d}\Omega + \sum_{i=1}^{E} \int_{\partial e_i \cap \Gamma_N} \boldsymbol{v}^h \cdot \bar{\boldsymbol{t}} \ \mathrm{d}\Gamma, \tag{4.7}$$

where $\boldsymbol{u}^h \in \boldsymbol{\mathcal{U}}^h \subset \boldsymbol{\mathcal{U}}$ and $\boldsymbol{v}^h \in \boldsymbol{\mathcal{V}}^h \subset \boldsymbol{\mathcal{V}}$ are trial and test functions, respectively. In the Interface-enriched Generalized Finite Element Method (IGFEM), the displacement field $\boldsymbol{u}^h$ is expressed as

$$\boldsymbol{u}^h = \underbrace{\sum_{i \in \iota_h} N_i(\boldsymbol{x}) \boldsymbol{U}_i}_{\text{standard FEM}} + \underbrace{\sum_{i \in \iota_w} \psi_i(\boldsymbol{x}) \boldsymbol{\alpha}_i}_{\text{enrichment}}, \tag{4.8}$$

where the first term is the standard FEM approximation, with $\iota_h$ denoting the index set of all standard nodes of the background mesh (dark circles in Figure 4.1), associated with standard shape functions $N_i$ and degree of freedoms (DOFs) $\boldsymbol{U}_i$; the second term represents the enrichment, where $\iota_w$ is the index set of enriched nodes that are associated with enrichment functions $\psi_i$ and enriched DOFs $\boldsymbol{\alpha}_i$. Enriched nodes (red circles in Figure 4.1) are created at the intersections between element edges and material interfaces. Cut background elements are split into *integration elements*, which as the name suggests, are used for the numerical quadrature of local stiffness and force vector arrays; in addition to be used for constructing enrichment functions $\psi_i$ and to ensure that the least number of integration points are used for their numerical quadrature (since $\psi_i \in C^0$), integration elements are also helpful in the post-processing stage to properly visualize the primal field [9]. Enrichment functions $\psi_i$ are constructed by means of Lagrange shape functions of integration elements. They attain their maximum value at the location of enriched nodes and decrease linearly to zero at other nodes in the cut elements (see Figure 4.2).



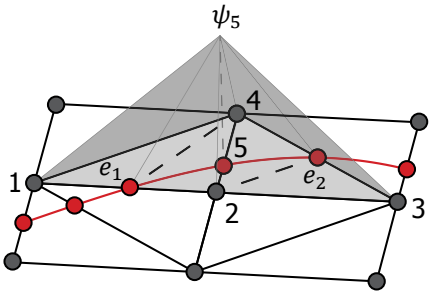



Figure 4.2: Enrichment function $\psi_5$ created by the aid of Lagrange shape functions in integration elements, which are created by considering the material interface (marked with a red curve) the function is non-zero only in cut elements $e_1$ (with nodes $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$) and element $e_2$ (with nodes $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$). The enrichment function, which is associated with enriched DOFs $\boldsymbol{\alpha}_5$, attains a maximum unit value at node $\boldsymbol{x}_5$ and decreases linearly to zero at all other nodes.

Background elements that are not cut by discontinuities follow the standard FEM procedure for constructing their corresponding stiffness matrix $\boldsymbol{k}_e$ and force vector $\boldsymbol{f}_e$. For integration elements, these arrays are computed as

$$\boldsymbol{k}_e = \int_e \boldsymbol{B}^{\mathrm{T}} \mathbf{C} \boldsymbol{B} \, \mathrm{d}e\,, \quad \text{and} \quad \boldsymbol{f}_e = \int_e \begin{bmatrix} \boldsymbol{N} & \boldsymbol{\psi} \end{bmatrix}^{\mathrm{T}} \boldsymbol{b} \, \mathrm{d}e + \int_{\partial e \cap \Gamma^{\mathrm{N}}} \begin{bmatrix} \boldsymbol{N} & \boldsymbol{\psi} \end{bmatrix}^{\mathrm{T}} \bar{\boldsymbol{t}} \, \mathrm{d}\partial e \tag{4.9}$$

where $\boldsymbol{B} \equiv \begin{bmatrix} \boldsymbol{\partial N} & \boldsymbol{\partial \psi} \end{bmatrix}$ is the strain-displacement matrix, $\boldsymbol{N}$ and $\boldsymbol{\psi}$ are the element's standard shape functions and enrichment functions, respectively, and the differential operator $\boldsymbol{\partial}$ defined in 2-D or 3-D is given, respectively, by

$$\boldsymbol{\partial} \equiv \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^{\mathrm{T}}, \tag{4.10}$$

or

$$\boldsymbol{\partial} \equiv \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^{\mathsf{T}}. \tag{4.11}$$

The global stiffness matrix $\boldsymbol{K}$ and force vector $\boldsymbol{F}$ then are obtained by

$$\boldsymbol{K} = \mathop{\mathbb{A}}_{i=1}^{E} \boldsymbol{k}_i, \qquad \boldsymbol{F} = \mathop{\mathbb{A}}_{i=1}^{E} \boldsymbol{f}_i, \tag{4.12}$$

where $\mathbb{A}$ is the standard finite element assembly operator. For more details on IGFEM's formulation see [7, 54].

IGFEM not only retains the most salient feature of X/GFEM—decoupling the background mesh from discontinuities—but also keeps the attractive properties of standard FEM: As enrichment functions are constructed with Lagrange shape functions of integration elements, they are exactly zero at original mesh nodes; DOFs associated with these therefore keep their physical meaning. Standard shape functions retain the Kronecker-$\delta$ property, *i.e.*, $N_i(\boldsymbol{x}_j) = \delta_{ij}$ for any standard node $\boldsymbol{x}_j$. Furthermore, prescribing non-zero essential boundary conditions along discontinuities is as straightforward as in the standard FEM [56], with smooth recovered tractions (reactions) [56, 57, 59]. The computer implementation is therefore considerably simpler than that of X/GFEM since it only requires a few modification to a displacement-based FE code [9]. Most importantly, intrinsic stability is attained in IGFEM by scaling enrichment functions with a proper scaling factor as interfaces approach mesh nodes [54]. With such scaling the rate of growth of the condition number is the same as that of standard FEM on fitted meshes, that is, $\mathcal{O}(h^{-2})$, where $h$ is the mesh size. Furthermore, even without such scaling factor—which causes the condition number of the global stiffness matrix to grow unbounded as interfaces get arbitrarily close to mesh nodes—a simple Jacobi-like preconditioner recovers stability [54].

### 4.2.2. Stress improvement procedure

SIP is derived from a mixed formulation based on the Hu-Washizu principle [60]. It relaxes the stress-strain relationship point-wise, but enhances the fulfillment of equilibrium over the patch. Following the procedure proposed by Payen and Bathe [29], the recovered stress $\boldsymbol{\sigma}_e$ of the $e$th element should satisfy the equilibrium in weak form over the calculation domain $\mathcal{E}$ (*i.e.*, a patch of elements), and the projection of the difference between enhanced and directly-calculated stresses onto the space of self-equilibrated stresses should be zero. To wit,

$$\sum_{\{e \in \mathcal{E}\}} \int_e \delta \tilde{\boldsymbol{\sigma}}_e^{\mathsf{T}} \left\{ \boldsymbol{\sigma}_e - \boldsymbol{\sigma}_e^h \right\} \mathrm{d}e = 0, \tag{4.13}$$

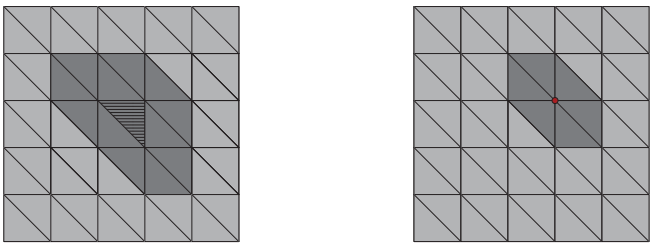
and

$$\sum_{\{e\in\mathcal{E}\}}\int_e \delta\boldsymbol{\zeta}_e^{\mathrm{I}}\left\{\mathrm{div}\left(\boldsymbol{\sigma}_e\right)+\boldsymbol{b}\right\}\,\mathrm{d}e=0, \tag{4.14}$$

where, denoting by $P_k$ is the space of complete polynomials of degree $k$ in the patch, $\boldsymbol{\zeta}_e$ belongs to the vector-valued space $[P_1]^2$, $\tilde{\boldsymbol{\sigma}}_e$ is any element in the tensor-valued space of the self-equilibrated stress defined as $\left\{\tilde{\boldsymbol{\sigma}}_e \mid \tilde{\boldsymbol{\sigma}}_e \in [P_2]^{2\times 2}, \nabla\cdot(\tilde{\boldsymbol{\sigma}}_e)=0\right\}$, and $\boldsymbol{\sigma}_e^h$ is the directly-calculated stress at the $e$th element, *i.e.*, $\boldsymbol{\sigma}_e^h = \mathbf{C}\boldsymbol{B}\boldsymbol{U}_e$, where $\boldsymbol{U}_e$ is the local element DOF vector. As shown in Figure 4.3(a), for evaluating the recovered stress of the hatched element, a patch of elements (in darker shade) is considered as the calculation domain. The quantities above can be interpolated as

$$\boldsymbol{\sigma}_e=\mathbf{E}_\sigma\hat{\boldsymbol{\sigma}},\quad \delta\boldsymbol{\zeta}_e=\mathbf{E}_\zeta\hat{\boldsymbol{\zeta}},\quad \delta\tilde{\boldsymbol{\sigma}}_e=\bar{\mathbf{E}}_\sigma\hat{\tilde{\boldsymbol{\sigma}}}, \tag{4.15}$$

where $\hat{\boldsymbol{\sigma}}$, $\hat{\boldsymbol{\zeta}}$, and $\hat{\tilde{\boldsymbol{\sigma}}}$ are coefficient vectors, and $\mathbf{E}_\sigma$, $\bar{\mathbf{E}}_\sigma$, and $\mathbf{E}_\zeta$ are interpolation



(a) (b)

Figure 4.3: (a) 2-D Patch of elements (darker shade) used to recover the stress of the target element (hatched); (b) Nodal patch (darker shade) used to compute the enhanced average stress at a node (red circle).

matrices as defined in Appendix A.2. Noteworthy, a complete polynomial of degree 2 is used to construct the interpolation matrices $\mathbf{E}_\sigma$ and $\bar{\mathbf{E}}_\sigma$, thereby having an expected order of convergence $\mathcal{O}(h^2)$ [29], where $h$ is the mesh size. Equations (4.13) and (4.14) can then be expressed as

$$\left[\sum_{\{e\in\mathcal{E}\}}\begin{pmatrix}\int_e \bar{\mathbf{E}}_\sigma^{\mathrm{I}}\mathbf{E}_\sigma\,\mathrm{d}e\\ \int_e \mathbf{E}_\zeta^{\mathrm{I}}\boldsymbol{\partial}^{\mathrm{I}}\mathbf{E}_\sigma\,\mathrm{d}e\end{pmatrix}\right]\hat{\boldsymbol{\sigma}}=\left\{\sum_{\{e\in\mathcal{E}\}}\begin{pmatrix}\int_e \bar{\mathbf{E}}_\sigma^{\mathrm{I}}\boldsymbol{\sigma}_e^h\,\mathrm{d}e\\ -\int_e \mathbf{E}_\zeta^{\mathrm{I}}\boldsymbol{b}\,\mathrm{d}e\end{pmatrix}\right\}, \tag{4.16}$$

which is solved for coefficient vector $\hat{\boldsymbol{\sigma}}$ of the target element, after which its corresponding recovered enhanced stress is simply $\boldsymbol{\sigma}_e=\mathbf{E}_\sigma(\boldsymbol{x})\hat{\boldsymbol{\sigma}}$. Gauss quadrature is used to evaluate Equation (4.16). Considering the interpolation matrices and the differential operator, the highest-order term is found in $\bar{\mathbf{E}}_\sigma^{\mathrm{I}}\mathbf{E}_\sigma$. Since the maximum polynomial order is quartic, six integration points are then used to exactly integrate the integrands.

A nodal stress field can be constructed by recovering stresses in all elements and averaging their values at nodes. In order to obtain the recovered stress $\bar{\boldsymbol{\sigma}}_j$ at the $j$th node, a *nodal patch* $\mathcal{E}_j$ containing $N_j$ elements sharing the node is first assembled (see Figure 4.3(b)). Then, the nodal stress $\bar{\boldsymbol{\sigma}}_j$ is simply computed as the average, *i.e.*,

$$\bar{\boldsymbol{\sigma}}_j = \frac{\sum\limits_{\{e \in \mathcal{E}_j\}} \boldsymbol{\sigma}_e(\boldsymbol{x}_j)}{N_j}, \tag{4.17}$$

where $\boldsymbol{\sigma}_e(\boldsymbol{x}_j) = \mathbf{E}_\sigma(\boldsymbol{x}_j)\hat{\boldsymbol{\sigma}}_e$ is the stress of the $e$th element evaluated at the $j$th node.

In IGFEM-based analysis, elements at either side of material interfaces are assigned different material properties. Therefore, the calculation domain considers only a patch of elements with the same material properties. Figure 4.4 shows three cases of element patches used for recovering the stress near a material interface (marked with red) in both 2-D and 3-D, where the union of elements connected to a target element is set as the stress calculation domain. A similar strategy is used

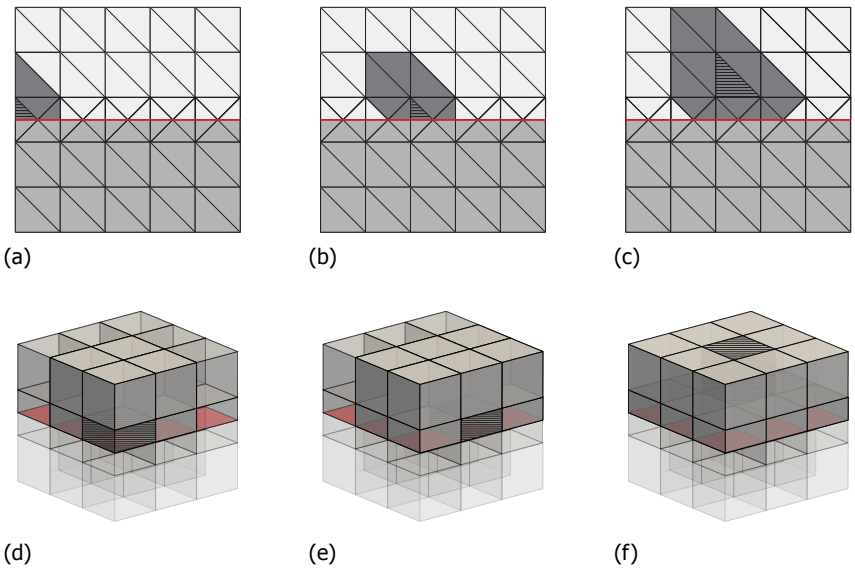



Figure 4.4: Computational domain used to compute the enhanced element stress considers only elements on the same side of the interface (shown in red). (a-c) 2-D element patch shown in darkest shade; (d-f) 3-D element patch shown as opaque.

to obtain the enhanced nodal stress (see Figure 4.5). There is a special scenario for recovering the stress of a node at a corner, where a larger patch of elements is used for making the nodal stress distribution smoother when constructing the calculation domain (see Figures 4.5(a) and (d) for 2-D and 3-D, respectively). Figures 4.5(c) and (f) show a situation where the patch of a node, which is not located along the material interface, is composed of both background and integration elements. It

is worth noting that the stress calculation domain (the patch) is tied to the finite element discretization, and therefore refining the original background mesh creates a discretization where discontinuities cut more background elements; however, the configuration of the element patch used to recover the stress distribution does not change for structured meshes regardless of the mesh size.



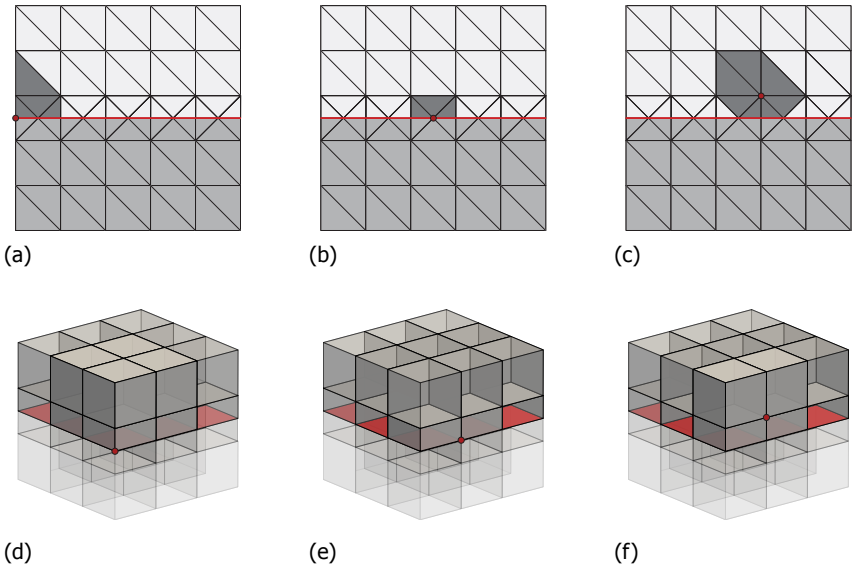



Figure 4.5: Computational domain used to compute the enhanced node stress considers only elements on the same side of the interface (shown in red). (a-c) 2-D element patch shown in darkest shade; (d-f) 3-D element patch shown as opaque.

### 4.2.3. Alternative smoothing formulations

We compare the proposed stress recovery technique for both elemental and nodal stresses to other two stress smoothing procedures. The first one averages directly-calculated stresses on a target element or node considering a patch; the averaged stress is computed as

$$\bar{\boldsymbol{\sigma}}^h = \frac{\sum \boldsymbol{\sigma}_e^h}{N_e}, \tag{4.18}$$

where $N_e$ is the number of the element in the patch, which is taken to be the same as that used for the proposed recovery technique. We use $\bar{\boldsymbol{\sigma}}_e^h$ and $\bar{\boldsymbol{\sigma}}_i^h$ to denote the averaged stresses of the $e$th element and $i$th node, respectively. We also consider an area-weighted average stress, computed as

$$\bar{\boldsymbol{\sigma}}^{h\triangleright} = \frac{\sum \boldsymbol{\sigma}_e^h A_e}{\sum A_e}, \tag{4.19}$$

where $A_e$ is the area of the $e$th element. Similarly, $\bar{\boldsymbol{\sigma}}_e^{h\triangleright}$ and $\bar{\boldsymbol{\sigma}}_i^{h\triangleright}$ denote the area-weighted averaged stresses of the $e$th element and $i$th node, respectively.

## 4.3. Numerical examples

In this section three examples are investigated to demonstrate the effectiveness of the proposed approach. The finite element analysis is performed under plane strain conditions, and no units are used so results can be interpreted in any consistent unit system. We perform a convergence study for both examples to investigate the accuracy of the proposed approach with increasingly finer meshes, and we use the $H_0$-norm of the stress to quantify the error:

$$\|\omega\|_{H_0} = \sqrt{\sum_{e\in\Omega^h}\left(\frac{\int_e (\boldsymbol{\sigma}_{\text{ex}}-\boldsymbol{\sigma}_e)^{\mathsf{T}}\cdot(\boldsymbol{\sigma}_{\text{ex}}-\boldsymbol{\sigma}_e)\,\mathrm{d}e}{\int_e \boldsymbol{\sigma}_{\text{ex}}^{\mathsf{T}}\cdot\boldsymbol{\sigma}_{\text{ex}}\,\mathrm{d}e}\right)} \tag{4.20}$$

where $\boldsymbol{\sigma}_{\text{ex}}$ is the exact analytical stress. This global error is evaluated by summing up the contribution of all standard and integration elements. For the error in directly-calculated stresses, $\boldsymbol{\sigma}_e$ is replaced by with $\boldsymbol{\sigma}_e^h$.



### 4.3.1. Eshelby's inclusion problem

As shown in Figure 4.6, a circular inclusion with radius $r_i$ is embedded into a matrix with radius $r_o = 2$. The mismatch in material properties at the interface between

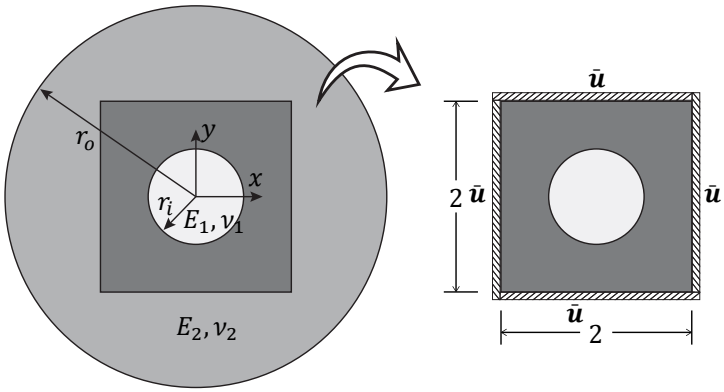



Figure 4.6: Schematic of Eshelby's inclusion problem with outside radius $r_o = 2$ and inside radius $r_i$. For the numerical analysis, a square domain of size $2 \times 2$ (darker shade) is considered, with the analytical displacement $\bar{\boldsymbol{u}}$ prescribed on the boundary. For the material properties, $E_2/E_1 = 10$.

matrix and inclusion is responsible for the discontinuous gradient field. Young's moduli and Poisson ratios for the inclusion and matrix are, respectively, $E_1 = 1, \nu_1 = 0.25$ and $E_2 = 10, \nu_2 = 0.3$. Dirichlet boundary conditions $u_r = r_o$ and $u_\theta = 0$ are imposed on the matrix's outer boundary (along $r = r_o$). The exact solution for the displacement field in polar coordinates is given by [10, 56, 59]

$$\begin{aligned} u_r &= \begin{cases} rf(r_i) & \text{for } 0 \le r \le r_i \\ rf(r) & \text{for } r_i \le r \le r_o \end{cases}, \\ u_\theta &= 0, \end{aligned} \tag{4.21}$$

where the function $f(r) = \left(1 - \frac{r_o^2}{r^2}\right) C + \frac{r_o^2}{r^2}$. In this equation, $C$ is a function of material properties:

$$C = \frac{(\lambda_1 + \mu_1 + \mu_2)\, r_o^2}{(\lambda_2 + \mu_2)\, r_i^2 + (\lambda_1 + \mu_1)\left(r_o^2 - r_i^2\right) + \mu_2 r_o^2}, \tag{4.22}$$

where $\mu_i, \lambda_i, i = 1,2$ are the Lamé constants, which can be obtained as a function of $E_i$ and $\nu_i$ as

$$\begin{aligned} \lambda_i &= \frac{E_i \nu_i}{(1+\nu_i)(1-2\nu_i)}, \\ \mu_i &= \frac{E_i}{2(1+\nu_i)}. \end{aligned} \tag{4.23}$$



The stress field for this problem is given by

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + \mu \left(\varepsilon_{ij} + \varepsilon_{ji}\right), \tag{4.24}$$

where $\delta_{ij}$ is the Kronecker delta, $i, j, k \in \{r, \theta\}$, and the exact strain field for this problem is given by

$$\begin{aligned} \varepsilon_{rr} &= f(r_i), & \varepsilon_{\theta\theta} &= f(r_i), & \varepsilon_{r\theta} &= 0, & &\text{for } 0 \leqslant r \leqslant r_i, \\ \varepsilon_{rr} &= \left(1 + \frac{r_o^2}{r^2}\right) C - \frac{r_o^2}{r^2}, & \varepsilon_{\theta\theta} &= f(r), & \varepsilon_{r\theta} &= 0, & &\text{for } r_i \leqslant r \leqslant r_o. \end{aligned} \tag{4.25}$$

A $2 \times 2$ square computational domain is chosen, as shown in Figure 4.6, and the domain is discretized by a finite element mesh composed of $60 \times 60 \times 2$ constant strain triangular elements. The exact displacement given in Equation (4.21) is prescribed along the square's boundary. For studying the performance of the proposed recovery technique, the internal radius $r_i$ is set to increase from $r_i = 0.35$ to $r_i = 0.42$ with step size $\Delta r_i = 0.035$, which ensures the creation of integration elements with bad aspect ratios and/or tiny areas along the interface. Furthermore, we compare results with standard FEM using conforming meshes, where cut elements are replaced by standard FEM elements with the same geometry as integration elements in IGFEM. This approach is called the Conformal Decomposition Finite Element Method (CDFEM) [61–63].

Figure 4.7 shows the maximum von Mises stress as a function of the internal radius $r_i$. Directly-calculated and recovered stresses for standard FEM and IGFEM are compared to the analytical solution. It can be seen that IGFEM recovered stresses are more accurate than directly-calculated ones for both standard FEM on matching meshes and IGFEM with a fixed unfitted mesh. As IGFEM can fully recover the approximation space of standard FEM, curves of directly-calculated and recovered stresses obtained under standard FEM on the same conforming discretizations overlap with those of IGFEM. The stresses closest to the analytical solution are obtained by the recovery technique applied to standard FEM on fitted meshes at the expense of losing the versatility of using a mesh that is fully decoupled from the

interface. Figure 4.8 further compares the proposed procedure to the other two smoothing techniques discussed in Chapter 4.2.3. It is shown that the maximum von Mises stress computed with averaged stress fields given by Equations (4.18) and (4.19) is worse than that obtained by the proposed approach.

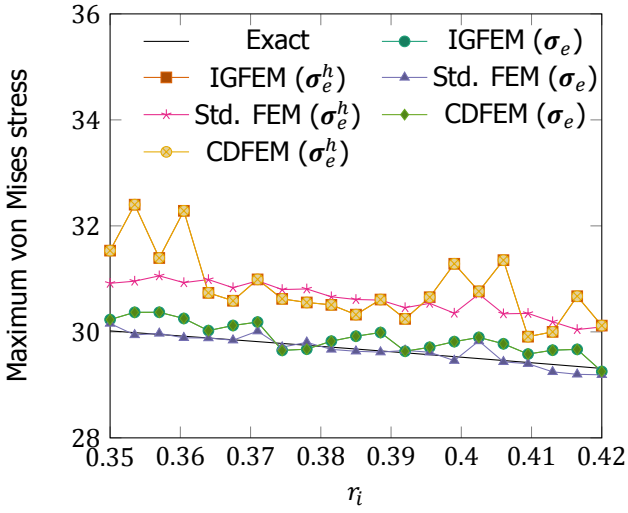



Figure 4.7: Maximum von Mises stresses associated with the internal radius $r_i$ evaluated by standard FEM with fitted meshes, IGFEM with a structured background mesh, and CDFEM with the conforming discretizations created in IGFEM.

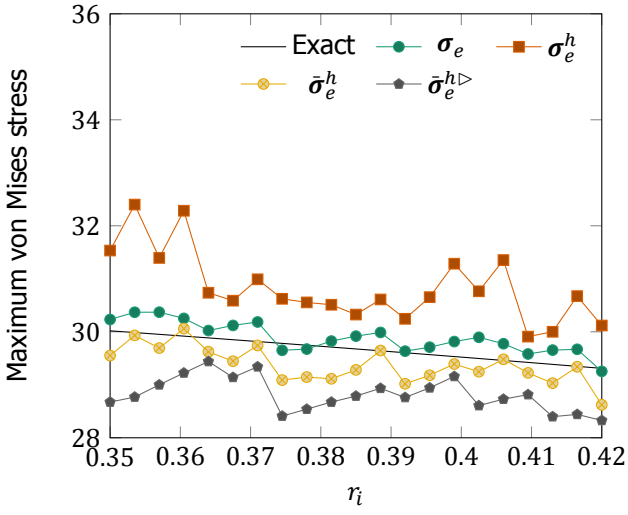



Figure 4.8: Maximum von Mises stress associated with the internal radius $r_i$ obtained with the averaged and area-weighted smoothing formulations using IGFEM.

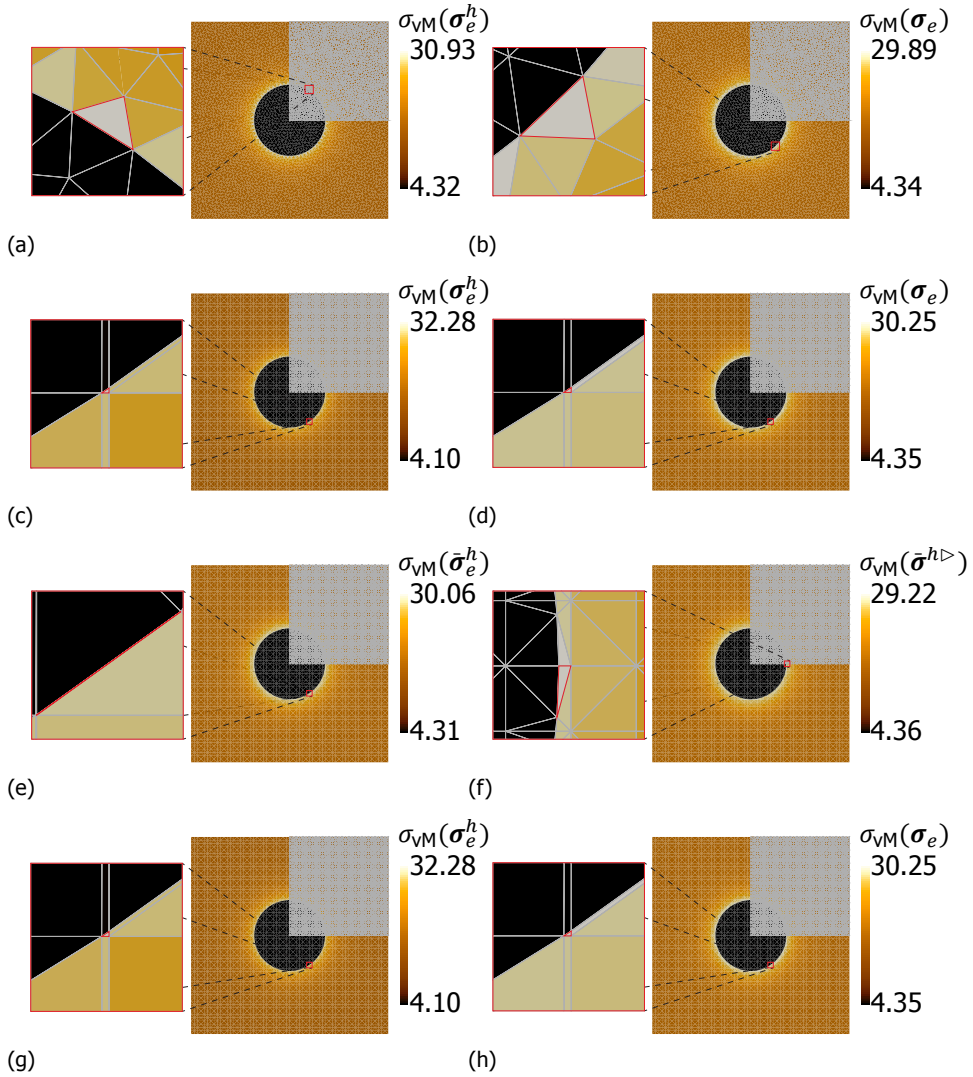



Figure 4.9: von Mises stress fields for an internal radius $r_i = 0.3605$: (a, b) Standard FEM on a fitted mesh; (c, d, e, f) IGFEM on an unfitted mesh; (g, h) Standard FEM on a conforming discretization created using IGFEM's integration elements (CDFEM); The fields correspond to (a, c, g) directly-calculated, (b, d, h) recovered, (e) averaged, and (f) area-weighted stresses.

The von Mises stress distributions for $r_i = 0.3605$ are given in Figure 4.9. The fields for directly-calculated and recovered stresses using standard FEM on good-quality fitted meshes are given in Figures 4.9(a) and (b), respectively. Figures 4.9(c)–(f) show the results for IGFEM-based analysis. Figure 4.9(c) displays an element along the material interface with the peak stress, which is caused by the tiny elemental area. However, the recovery technique proposed removes the stress over-

estimation and makes the stress distribution smoother around that region (see Figure 4.9(d)). The von Mises stress fields obtained with the averaged and area-weighted elemental stresses are displayed on Figures 4.9(e) and (f), which shows the averaging equations yield an underestimated stress. Figures 4.9(g) and (h) show von Mises distributions evaluated with directly-calculated and recovered stresses with CDFEM, which are the same to those of IGFEM.

Finally, four background meshes with $30 \times 30 \times 2$, $60 \times 60 \times 2$, $120 \times 120 \times 2$, and $240 \times 240 \times 2$ linear triangular elements are used for the convergence analysis with $r_i = 0.371$. Figure 4.10 shows the global error defined in Equation (4.20) as a function of mesh size $h$. It can be seen that the recovered stress $\boldsymbol{\sigma}_e$ convergences faster (with a rate of $1.20$) than the directly-calculated stress $\boldsymbol{\sigma}_e^h$. The recovered stress with a coarse mesh can reach the same level of accuracy of the directly-calculated stress with a refined mesh.



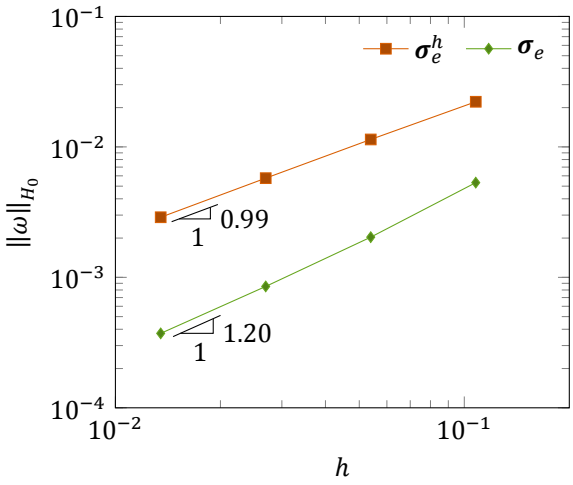



Figure 4.10: $H_0$-norm of the error in stress as a function of mesh size $h$ with $r_i = 0.371$. The curves for recovered stress $\boldsymbol{\sigma}_e$ and directly-calculated stress $\boldsymbol{\sigma}_e^h$ show that the former is not only more accurate for any given mesh size, but also that it converges at a faster rate.

### 4.3.2. An elliptical cavity in an infinite plate under remote load

Figure 4.11 shows a traction-free elliptical hole in an infinite domain that is subjected to a distant tensile stress $\sigma_0$ in the $x$ direction, where the elliptical axes $2a$ and $2b$ are aligned with the $x$ and $y$ coordinate axes, respectively. The analytical stress field for this problem is given by [64]:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \sigma_0 \left( \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} - p_a p_b \left( \frac{1}{2} \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix} - \left( \frac{b}{a} + \frac{1}{2} \right) \begin{Bmatrix} H_4 \\ H_1 \\ H_5 \end{Bmatrix} \right) \right), \tag{4.26}$$

where

$$
\begin{aligned}
H_1 &= \frac{a^2\rho_a^2\rho_b^2 + b^2\rho_a^2 + ab\rho_a\rho_b}{(a\rho_b + b\rho_a)^2} - \rho_b^2 n_x^2 - \rho_a^2 n_y^2 + \left(5\rho_a^2 + 5\rho_b^2 - 4\eta\right) n_x^2 n_y^2, \\
H_2 &= \frac{\rho_b a\left(a\rho_b + b\rho_a + 2b\rho_a\rho_b^2 + a\rho_b^3\right)}{(a\rho_b + b\rho_a)^2} + n_y^2\left[2 - 6\rho_b^2 + \left(\rho_a^2 + 9\rho_b^2 - 4\eta\right) n_y^2\right], \\
H_3 &= n_x n_y\left[1 - 3\rho_b^2 + \left(3\rho_a^2 + 7\rho_b^2 - 4\eta\right) n_y^2\right], \\
H_4 &= \frac{\rho_a b\left(a\rho_b + b\rho_a + 2a\rho_a^2\rho_b + b\rho_a^3\right)}{(a\rho_b + b\rho_a)^2} + n_x^2\left[2 - 6\rho_a^2 + \left(9\rho_a^2 + \rho_b^2 - 4\eta\right) n_x^2\right], \\
H_5 &= n_x n_y\left[1 - 3\rho_a^2 + \left(7\rho_a^2 + 3\rho_b^2 - 4\eta\right) n_x^2\right], \\
\rho_a &= \frac{a}{\sqrt{a^2 + \lambda}}, \\
\rho_b &= \frac{b}{\sqrt{b^2 + \lambda}},
\end{aligned}
\tag{4.27}
$$

wherein the above equation

$$
\begin{aligned}
\eta &= \rho_a^2 n_x^2 + \rho_b^2 n_y^2 + 1, \\
\boldsymbol{n} &= (n_x, n_y) = \frac{1}{\sqrt{x^2(b^2 + \lambda)^2 + y^2(a^2 + \lambda)^2}}\left(x(b^2 + \lambda), y(a^2 + \lambda)\right), \\
\lambda &= \frac{1}{2}\left(x^2 + y^2 - a^2 - b^2 + \sqrt{\left(x^2 + y^2 - a^2 + b^2\right)^2 + 4\left(a^2 - b^2\right) y^2}\right).
\end{aligned}
\tag{4.28}
$$

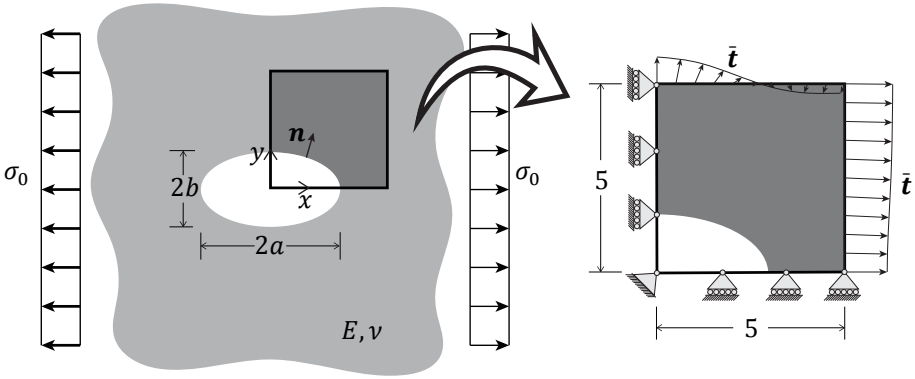



Figure 4.11: Schematic of an elliptical cavity with semi-principal axes $a$ and $b$ in an infinite plate under remote loading $\sigma_0$ in the $x$ direction. For the analysis, only the shaded area is considered due to symmetry (with appropriate symmetry boundary conditions) and exact tractions obtained from Equation (4.26) are applied on the right and top sides.

Due to the symmetry, only the $5 \times 5$ region (darker shade) is considered as the analysis domain, and is discretized with a mesh of $60 \times 60 \times 2$ constant strain trian-

gles. Symmetry boundary conditions are then imposed along left and bottom edges, and tractions obtained from the analytical equation with $\sigma_0 = 1$ are prescribed on top and right sides. In order to avoid an ill-conditioned stiffness matrix, a material with Young's modulus $10^{-9}$ is assigned to the void part; the solid part has Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.3$.

For this problem we adjust the ratio between the semi-major axis $a$ and semi-minor axis $b$ to investigate the performance of the proposed recovery technique, where $a$ and $b$ are chosen from a set of values $\{2.49, 1.245, 0.498, 0.249, 0.1245\}$, resulting $a/b$ ratios from $0.05$ to $20$. In addition, we solve the problem on three different discretizations, *i.e.*, $60 \times 60 \times 2$, $120 \times 120 \times 2$, and $240 \times 240 \times 2$ triangular elements with corresponding mesh sizes $h = 0.1348$, $0.0674$, and $0.0337$, respectively. For simplicity, the maximum von Mises stress $\sigma_{\text{vM}}$ obtained via direct calculation and the recovery technique is normalized by the analytical result for the corresponding ratio.



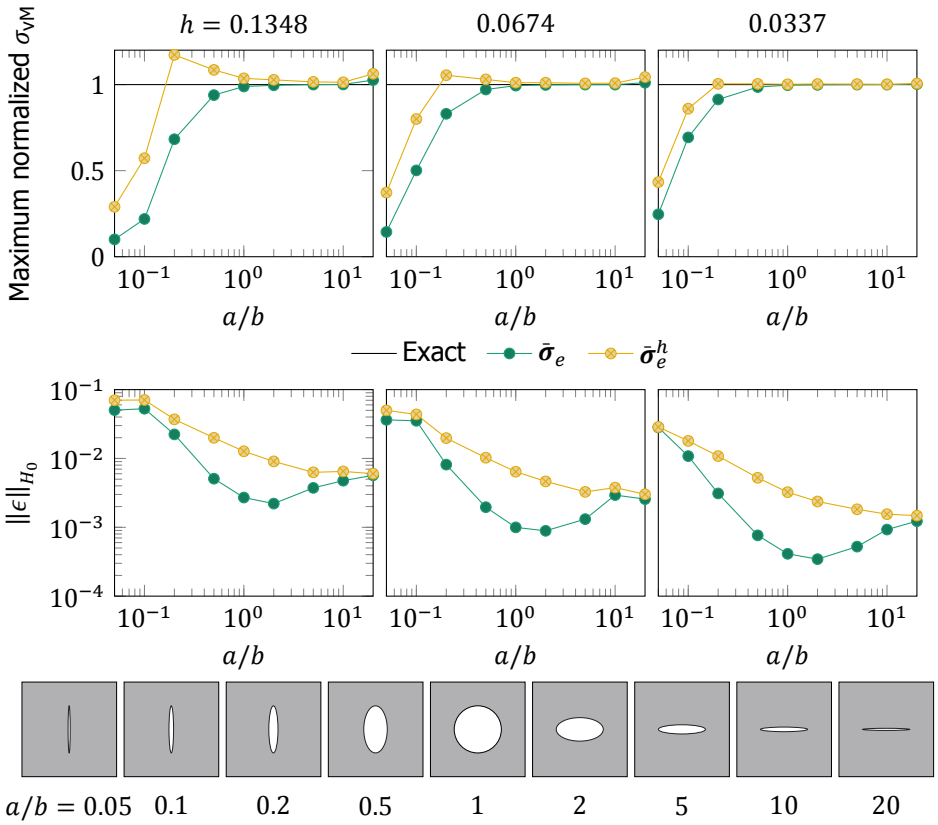



Figure 4.12: Maximum normalized von Mises stress $\sigma_{\text{vM}}$ (top) and $H_0$-norm of the error in stress (middle) obtained with directly-calculated and recovered stresses as a function of $a/b$. Three different discretizations are used with $60 \times 60 \times 2$, $120 \times 120 \times 2$ and $240 \times 240 \times 2$ elements. The corresponding mesh sizes are $h = 0.1348$, $0.0674$, and $0.0337$, respectively. The cavity configurations for different values of $a/b$ are shown at the bottom.

Figure 4.12 shows the normalized von Mises stress values as a function of $a/b$, where it can be seen that the proposed recovery technique generally provides more accurate approximations than the direct calculation. However, as the ratio $a/b$ is decreased, the cavity progressively morphs into a crack perpendicular to the remote loading. As a result, the stress field becomes increasingly singular, and both directly-calculated and recovered stresses do not behave well. The corresponding errors in the $H_0$ norm are also given in the figure, where the difference between the recovery technique and the direct calculation becomes small for small and large values of $a/b$.

We now set $a = b$ to consider a circular cavity in an infinite plate with the same boundary conditions. The analyses are performed with different values of radius $a$, increasing from $a = 1.895$ to $a = 2.095$ with step $\Delta a = 0.01$. Figure 4.13 shows the stress concentration factor $K_t$—*i.e.*, the ratio of the highest stress to the nominal far field stress [65]—evaluated by means of Equations (4.17), (4.18), and (4.19) for different radii $a$. For this problem $K_t = 3$ (Equation (4.26) for $(x, y) = (0, a)$) and thus it can be seen that all numerical values fluctuate around this value, since different values of radius $a$ also imply different finite element discretizations. Averaged and area-weighted averaged stresses show large differences between the maximum and minimum values, and thus they are greatly effected by the discretization. For instance, there is a huge drop in the stress concentration factor as $a$ increases

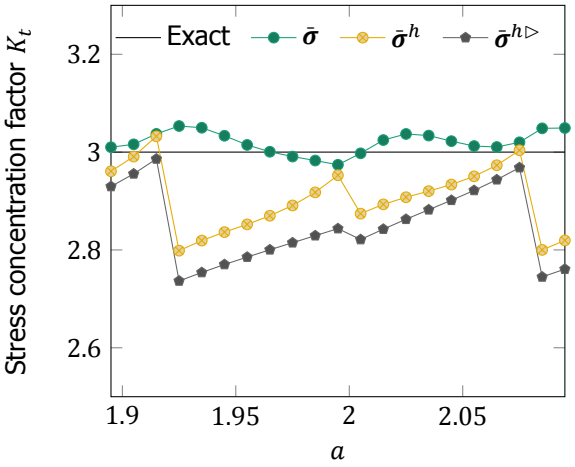



Figure 4.13: Stress concentration factor $K_t$ associated with radius $a$ that is evaluated by averaged and area-weighted average stresses, recovery technique, and analytical solution, respectively.

from $1.915$ to $1.925$. With $a = 1.915$, Figure 4.14(a) shows the element patch (in darker shade) used to evaluate $K_t$ at the location of a node (red circle). Since patch elements are close to the expected coordinate, they provide an accurate approximation of the stress concentration factor. As the radius $a$ increases to $1.925$, a larger patch is considered as the calculation domain (see Figure 4.14(b)). However, considering more distant elements in the patch, which have lower stress, yields an underestimated stress concentration factor. The proposed recovery technique predicts the best approximation of the analytical value as its curve is the most stable

one. Notice that the same patches are used in the proposed recovery technique, which is therefore less sensitive to changes in the discretization. Figures 4.14(c) and (d) show the same configuration of element patches within a refined mesh used to recover the stress of nodes shown Figures 4.14(a) and (b).

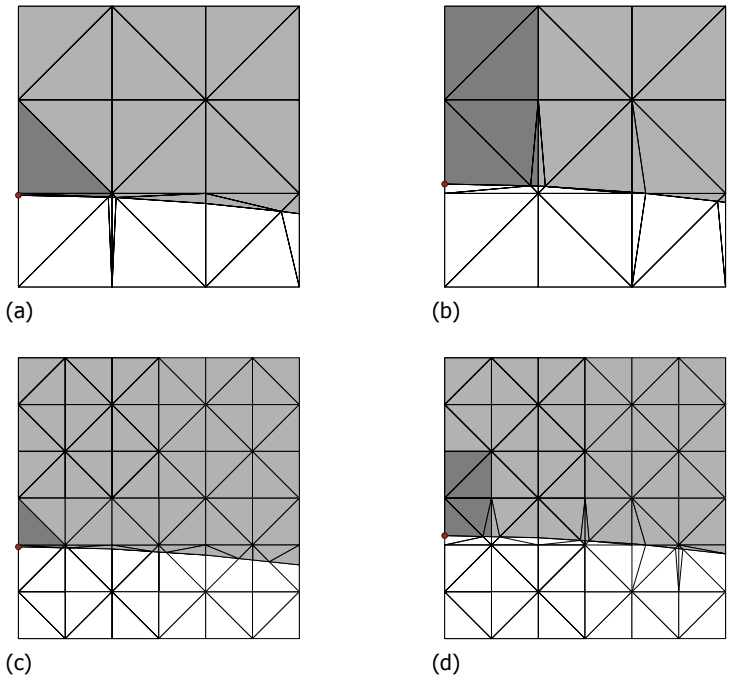



Figure 4.14: Element patches (darker shade) used to evaluate the stress concentration factor $K_t$ at the location of node (red circle). $a = 1.915$ for (a, c) and $a = 1.925$ for (b, d). The patches' shape remains the same as meshes are refined.

Furthermore, we perform a convergence study by using background meshes with increasingly smaller mesh size $h$ to investigate the behavior of each formulation. In addition to the mesh with $60 \times 60 \times 2$ elements used earlier, we consider extra meshes with $30 \times 30 \times 2$, $120 \times 120 \times 2$, and $240 \times 240 \times 2$ linear triangular elements. For a given mesh size, minimum, maximum, and average values were obtained for the same radii range, *i.e.*, $a = [1.895, 2.095]$. Results are reported in Figure 4.15, where the curves correspond to average values and the shades span from minimum to maximum values of the stress concentration factor $K_t$. While all formulations approach the analytical value as the mesh is refined, the stress recovery technique shows the most accurate prediction, as its shaded area is the smallest one among all formulations. With the same background meshes, we also calculate the global $H_0$-norm of the error for $a = 1.895$. Note that this study looks again at elemental stresses for computing Equation (4.20). Figure 4.16 shows that the convergence rate of the recovered stress is $1.34$, which is in agreement with the value

obtained when applying the SIP recovery technique on standard FEM tetrahedral elements [31].



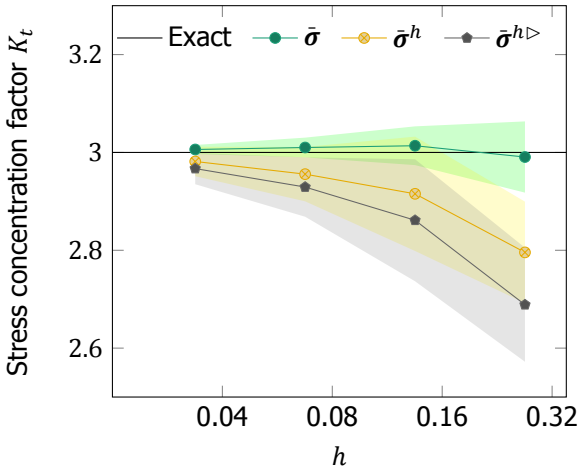



Figure 4.15: Stress concentration factor $K_t$ as a function of mesh size $h$. In this convergence study, meshes with $30 \times 30 \times 2$, $60 \times 60 \times 2$, $120 \times 120 \times 2$, and $240 \times 240 \times 2$ linear triangular elements were used. For each mesh size, the figure reports minimum, average, and maximum values of $K_t$ for radii in the range $a = [1.895, 2.095]$.

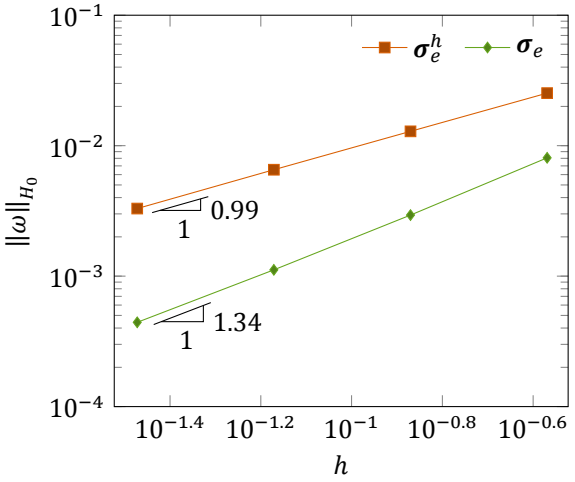



Figure 4.16: $H_0$-norm of the error in stress as a function of mesh size $h$ with $a = 1.895$. The curves for recovered stress $\boldsymbol{\sigma}_e$ and directly-calculated stress $\boldsymbol{\sigma}_e^h$ show that the former is not only more accurate for any given mesh size, but also that it converges at a faster rate.

### 4.3.3. Pressurized sphere

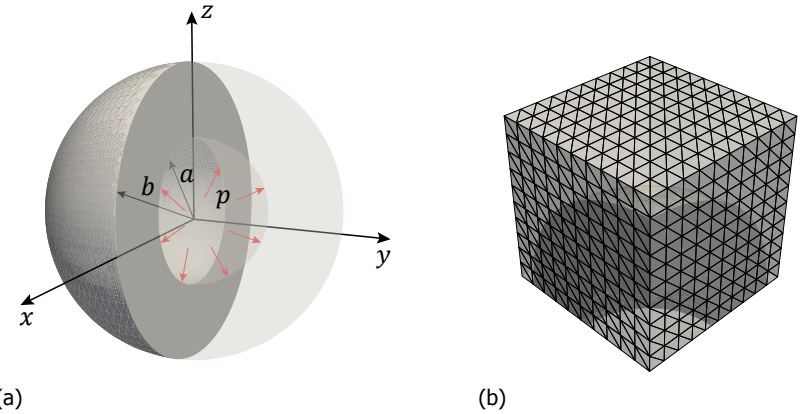
A hollow sphere with internal radius $a = 1$ and external radius $b = 2$ is considered in this example, where a uniform pressure $p = 1$ is applied to the internal surface (see Figure 4.17(a)). The analytical displacements and stresses for this problem are given by [66]:

$$\begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} = \frac{a^3 p \left(2r^3(1-2\nu) + b^3(1+\nu)\right)}{2\left(-a^3+b^3\right)Er^2} \begin{Bmatrix} \cos\phi \sin\theta \\ \sin\phi \sin\theta \\ \cos\theta \end{Bmatrix}, \tag{4.29}$$

and

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} = \frac{a^3 b^3 p}{8\left(a^3-b^3\right)r^3} \begin{Bmatrix} -\left(1+8\frac{r^3}{b^3}+3\left(\cos 2\theta + \cos 2\theta \cos 2\phi - \cos 2\phi\right)\right) \\ -\left(1+8\frac{r^3}{b^3}+3\left(\cos 2\theta - \cos 2\theta \cos 2\phi + \cos 2\phi\right)\right) \\ -8\frac{r^3}{b^3}+6\cos 2\theta \\ 6(\sin\theta)^2 \sin 2\phi \\ 6\sin 2\theta \sin\phi \\ 6\cos\phi \sin 2\theta \end{Bmatrix}, \tag{4.30}$$

respectively, where $r = \sqrt{x^2+y^2+z^2}$ is the distance from the sphere center, $\phi = \arctan(y/x)$, and $\theta = \arccos(z/r)$. Due to symmetry, we only consider




(a) (b)

Figure 4.17: (a) Schematic of an internally pressurized sphere with internal radius $a$ and external radius $b$ where the uniform pressure $p$ is imposed on the internal surface; (b) Due to symmetry, an 1/8 sphere is considered and immersed into a background mesh with $10 \times 10 \times 10 \times 6$ linear tetrahedral elements.

one eighth of the sphere for the analysis (see Figure 4.17(b)); consequently, symmetric boundary conditions are prescribed on the planar surfaces in the $x$, $y$, and

$z$ directions. The sphere portion is then immersed into a cubic domain with dimensions $2.25 \times 2.25 \times 2.25$, which is discretized by a structured mesh composed of $10 \times 10 \times 10 \times 6$ linear tetrahedra.

According to the analytical solution, the maximum value of von Mises stress is $\sigma_{\text{vM}} = 1.71$. However, the maximum von Mises stress obtained with directly calculated stresses is $\sigma_{\text{vM}} = 2.11$ (see Figure 4.18(a)), where peak stresses are the result of several tiny integration elements (marked with gray). Figure 4.18(b) shows that the recovered stresses are much closer to the exact solutions. The corresponding error distributions are shown in Figures 4.18(c) and 4.18(d), where the maximum value of the latter is one order of magnitude smaller than that of the former. In addition, background meshes with $5 \times 5 \times 5 \times 6$, $10 \times 10 \times 10 \times 6$, $20 \times 20 \times 20 \times 6$, and $40 \times 40 \times 40 \times 6$ tetrahedral elements are used to perform a convergence study, and we compute the $H_0$-norm of the error for both directly-calculated and recovered stresses. Figure 4.19 shows a much faster convergence rate for recovered stresses ($1.46$ vs. $0.89$). The convergence rate is on par with the value $1.49$ obtained by the recovery technique applied on standard FEM using fitted meshes by Sharma *et al.* [31].



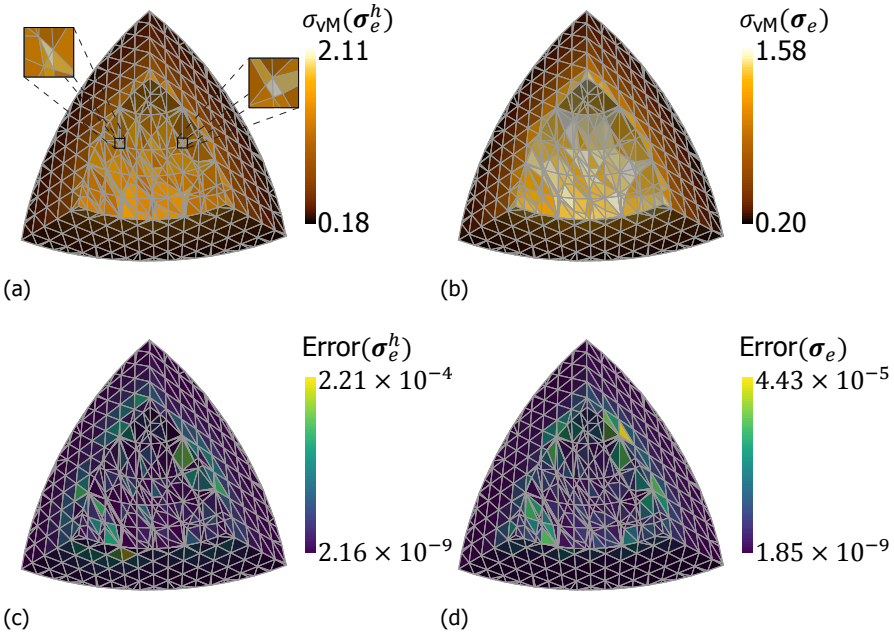



Figure 4.18: (a,b) von Mises stress fields obtained with (a) directly-calculated and (b) recovered stresses, where the former shows several tiny integration elements (marked with red) with peak stresses and the latter displays a more smooth distribution. (c,d) Element-wise error distributions in the $H_0$ form under (a) directly-calculated and (b) recovered stresses, where the maximum value of the latter is one magnitude smaller than that of the former.

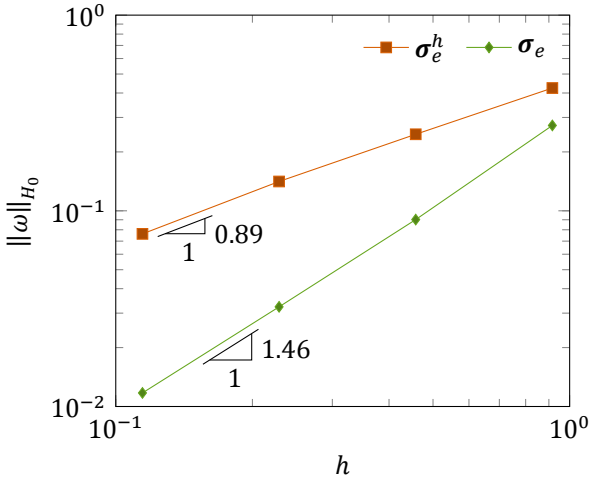



Figure 4.19: $H_0$-norm of the error in stress as a function of mesh size $h$, where the recovered stress $\boldsymbol{\sigma}_e$ converges faster than the directly-calculated stress $\boldsymbol{\sigma}_e^h$, this time with a higher convergence rate of $1.46$.

## 4.4. Summary and conclusions

In this article we extended the stress improvement procedure (SIP) proposed by Payen and Bathe [29] to recover the stress field in problems with discontinuous gradient fields, whose solution is obtained by means of an enriched finite element analysis. We investigated the proposed recovery technique in conjunction with the Interface-enriched Generalized Finite Element Method (IGFEM), although we foresee no further developments when coupling the technique to other EFEA such as X/GFEM. In IGFEM material interfaces subdivide mesh elements into subdomains to which different material properties are assigned. In order to smooth the stress field close to the interface, the calculation domain used is carefully constructed by selecting a patch of (integration) elements with the same material properties. We also used the recovery technique to evaluate nodal stresses along discontinuities, where *only* integration elements at the same side of discontinuities are used as the element patch. For corner nodes along the discontinuity, a larger patch of elements that involves standard uncut elements is used as the calculation domain.

The technique was demonstrated by means of 2-D and 3-D numerical examples. With Eshelby's inclusion problem, we showed that recovered IGFEM stresses are more accurate than directly-calculated ones for both standard FEM and IGFEM. Therefore, for IGFEM the stress recovery technique provides an elegant means to approximate the stress field compared to standard FEM on fitted meshes. The circular cavity example was used to investigate the recovery of nodal stress fields. In order to avoid a singular stiffness matrix, the cavity was assigned a material with a Young's modulus orders of magnitude smaller than that of the solid material. Averaged and area-weighted smoothing techniques were also used to evaluate the nodal stresses. In comparison, the proposed recovery technique does not only pro-

vide a more accurate result for the value of stress concentration factor, but is also less sensitive to the choice of discretization (see Figure 4.13). For the 3-D pressurized sphere example, it was shown that the proposed recovery technique can avoid overestimated stresses in tiny integration elements. More importantly, the convergence rate of the recovered stresses associated with mesh size $h$ is close to the value $1.49$ obtained from standard FEM using the recovery approach on fitted meshes by Sharma *et al.* [31]. Note that although in all examples we used analytical solutions, the proposed methodology could also be used for *a posteriori* error estimation when solving problems without closed-form solutions (similarly to other recovery approaches such as SPR)

Even though this work focused on static material interfaces, the proposed procedure does not suffer any modification for problems with evolving interfaces, *e.g.*, stress-based topology optimization, for which an accurate approximation of the stress field is paramount [67]. In fact, the proposed method has been used to recover the stress field in an incoming article that designs structures with tailored brittle fracture resitance [68], which hinges on an accurate evaluation of energy release rates—which in turn greatly depend on the approximation of stress field.



As the proposed recovery technique aims to improve the stress approximation for a linear field using a quadratic interpolant in a target element, future work may develop the proposed methodology to recover stress fields of higher order. This could be done via increasing the order of the approximation—for both standard and enrichment functions—and using an order higher interpolant to recover the stress field. The proposed technique can also be extended straightforwardly to recover stress fields in problems with strong discontinuities such as fracture, for which the Discontinuity Enriched Finite Element Method (DE-FEM) has been proposed recently [8, 9]; DE-FEM is a generalization of IGFEM for the seamless treatment of both weak and strong discontinuities with a unique formulation, and inherits all of IGFEM's advantages. Nevertheless, the DE-FEM formulation does not use singular enrichments so a modification would be required to the enriched formulation if accurate singular stress fields are sought. For instance, following the work of Duflot and Bordas [35], the stress field could be decomposed into smooth, non-smooth non-singular, and non-smooth singular components. While the first two could be handled using the same strategy mentioned in this paper, the last one could use an analytical formulation based on stress intensity factors that describes the asymptotic fields near the crack tip. This could greatly improve the recovered stress at crack fronts.

As a final note, although the proposed recovery approach can greatly improve the accuracy of the stress field obtained with IGFEM, it is just a post-processing technique that does not solve the fundamental issue at its core: Because IGFEM recovers—through an enriched procedure—the same standard finite element space, the recovery of gradient fields is still susceptible to the way elements are cut by interfaces. Therefore, addressing this issue at the root would require for alternative means to create the enriched finite element space.

# References


[1] J. Wood, *Finite element analysis of composite structures,* Composite Structures **29**, 219 (1994).

[2] Q. Du and D. Wang, *Recent progress in robust and quality delaunay mesh generation,* Journal of Computational and Applied Mathematics **195**, 8 (2006), special Issue: The International Symposium on Computing and Information (ISCI2004).

[3] H. Li, T. Yamada, P. Jolivet, K. Furuta, T. Kondoh, K. Izui, and S. Nishiwaki, *Full-scale 3d structural topology optimization using adaptive mesh refinement based on the level-set method,* Finite Elements in Analysis and Design **194**, 103561 (2021).

[4] J. T. Oden, C. A. Duarte, and O. C. Zienkiewicz, *A new cloud-based hp finite element method,* Computer Methods in Applied Mechanics and Engineering **153**, 117 (1998).

[5] N. Moës, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing,* International Journal for Numerical Methods in Engineering **46**, 131 (1999).

[6] T. P. Fries and T. Belytschko, *The extended/generalized finite element method: An overview of the method and its applications,* International Journal for Numerical Methods in Engineering **84**, 253.

[7] S. Soghrati, A. M. Aragón, C. A. Duarte, and P. H. Geubelle, *An interface-enriched generalized fem for problems with discontinuous gradient fields,* International Journal for Numerical Methods in Engineering **89**, 991 (2012).

[8] A. M. Aragón and A. Simone, *The discontinuity-enriched finite element method,* International Journal for Numerical Methods in Engineering **112**, 1589 (2017).

[9] J. Zhang, S. J. van den Boom, F. van Keulen, and A. M. Aragón, *A stable discontinuity-enriched finite element method for 3-d problems containing weak and strong discontinuities,* Computer Methods in Applied Mechanics and Engineering **355**, 1097 (2019).

[10] N. Sukumar, D. Chopp, N. Moës, and T. Belytschko, *Modeling holes and inclusions by level sets in the extended finite-element method,* Computer Methods in Applied Mechanics and Engineering **190**, 6183 (2001).

[11] J. R. Shewchuk, *What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures*, Tech. Rep. (In Proc. of the 11th International Meshing Roundtable, 2002).

[12] J. T. Oden and H. J. Brauchli, *On the calculation of consistent stress distributions in finite element approximations,* International Journal for Numerical Methods in Engineering **3**, 317 (1971).

[13] H. Brauchli and J. T. Oden, *Conjugate approximation functions in finite-element analysis,* Quarterly of Applied Mathematics **29**, 65 (1971).

[14] E. Hinton and J. S. Campbell, *Local and global smoothing of discontinuous finite element functions using a least squares method,* International Journal for Numerical Methods in Engineering **8**, 461 (1974).

[15] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique,* International Journal for Numerical Methods in Engineering **33**, 1331 (1992).

[16] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity,* International Journal for Numerical Methods in Engineering **33**, 1365 (1992).

[17] N.-E. Wiberg and F. Abdulwahab, *Patch recovery based on superconvergent derivatives and equilibrium,* International Journal for Numerical Methods in Engineering **36**, 2703 (1993).

[18] N.-E. Wiberg, F. Abdulwahab, and S. Ziukas, *Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions,* International Journal for Numerical Methods in Engineering **37**, 3417 (1994).

[19] T. Blacker and T. Belytschko, *Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements,* International Journal for Numerical Methods in Engineering **37**, 517 (1994).

[20] J. J. Ródenas, M. Tur, F. J. Fuenmayor, and A. Vercher, *Improvement of the superconvergent patch recovery technique by the use of constraint equations: the spr-c technique,* International Journal for Numerical Methods in Engineering **70**, 705 (2007).

[21] O. A. González-Estrada, S. Natarajan, J. J. Ródenas, H. Nguyen-Xuan, and S. P. Bordas, *Efficient recovery-based error estimation for the smoothed finite element method for smooth and singular linear elasticity,* Computational Mechanics **52**, 37 (2013).

[22] O. A. González-Estrada, E. Nadal, J. Ródenas, P. Kerfriden, S. P.-A. Bordas, and F. Fuenmayor, *Mesh adaptivity driven by goal-oriented locally equilibrated superconvergent patch recovery,* Computational Mechanics **53**, 957 (2014).

[23] S. Rajendran and K. Liew, *Optimal stress sampling points of plane triangular elements for patch recovery of nodal stresses,* International journal for numerical methods in engineering **58**, 579 (2003).

[24] M. Tabbara, T. Blacker, and T. Belytschko, *Finite element derivative recovery by moving least square interpolants,* Computer Methods in Applied Mechanics and Engineering **117**, 211 (1994).

[25] B. Boroomand and O. C. Zienkiewicz, *Recovery by equilibrium in patches (rep),* International journal for numerical methods in engineering **40**, 137 (1997).

[26] B. Boroomand and O. C. Zienkiewicz, *An improved rep recovery and the effectivity robustness test,* International Journal for numerical methods in engineering **40**, 3247 (1997).

[27] F. Ubertini, *Patch recovery based on complementary energy,* International Journal for Numerical Methods in Engineering **59**, 1501 (2004).

[28] A. Benedetti, S. de Miranda, and F. Ubertini, *A posteriori error estimation based on the superconvergent recovery by compatibility in patches,* International Journal for Numerical Methods in Engineering **67**, 108 (2006).

[29] D. J. Payen and K.-J. Bathe, *A stress improvement procedure,* Computers & Structures **112-113**, 311 (2012).

[30] K.-J. ”u. r. Bathe, *Finite element procedures: Klaus-jurgen bathe,* Englewood Cliffs, NJ Klaus-J ”u rgen Bathe, cop (2006).

[31] R. Sharma, J. Zhang, M. Langelaar, F. van Keulen, and A. M. Aragón, *An improved stress recovery technique for low-order 3d finite elements,* International Journal for Numerical Methods in Engineering **114**, 88 (2018).

[32] Q. Xiao and B. Karihaloo, *Statically admissible stress recovery for crack problems,* Proc. ICF11 , 20 (2005).

[33] Q. Xiao, B. L. Karihaloo, and X. Liu, *Direct determination of sif and higher order terms of mixed mode cracks by a hybrid crack element,* International Journal of Fracture **125**, 207 (2004).

[34] Q.-Z. Xiao and B. L. Karihaloo, *Improving the accuracy of xfem crack tip fields using higher order quadrature and statically admissible stress recovery,* International Journal for Numerical Methods in Engineering **66**, 1378 (2006).

[35] M. Duflot and S. Bordas, *A posteriori error estimation for extended finite elements by an extended global recovery,* International Journal for Numerical Methods in Engineering **76**, 1123 (2008).

[36] Y. Jin, O. González-Estrada, O. Pierard, and S. Bordas, *Error-controlled adaptive extended finite element method for 3d linear elastic crack propagation,* Computer Methods in Applied Mechanics and Engineering **318**, 319 (2017).

[37] Z. Wang, T. Yu, T. Q. Bui, N. A. Trinh, N. T. H. Luong, N. D. Duc, and D. H. Doan, *Numerical modeling of 3-d inclusions and voids by a novel adaptive xfem,* Advances in Engineering Software **102**, 105 (2016).

[38] T. Yu and T. Q. Bui, *Numerical simulation of 2-d weak and strong discontinuities by a novel approach based on xfem with local mesh refinement,* Computers & Structures **196**, 112 (2018).

[39] S. Bordas and M. Duflot, *Derivative recovery and a posteriori error estimate for extended finite elements,* Computer Methods in Applied Mechanics and Engineering **196**, 3381 (2007).

[40] S. Bordas, M. Duflot, and P. Le, *A simple error estimator for extended finite elements,* Communications in Numerical Methods in Engineering **24**, 961 (2008).

[41] J. J. Ródenas, O. A. González-Estrada, J. E. Tarancón, and F. J. Fuenmayor, *A recovery-type error estimator for the extended finite element method based on singular+smooth stress field splitting,* International Journal for Numerical Methods in Engineering **76**, 545 (2008).



[42] J. Ródenas, O. González-Estrada, P. Díez, and F. Fuenmayor, *Accurate recovery-based upper error bounds for the extended finite element framework,* Computer Methods in Applied Mechanics and Engineering **199**, 2607 (2010).

[43] O. González-Estrada, J. Ródenas, S. Bordas, E. Nadal, P. Kerfriden, and F. Fuenmayor, *Locally equilibrated stress recovery for goal oriented error estimation in the extended finite element method,* Computers & Structures **152**, 1 (2015).

[44] C. Prange, S. Loehnert, and P. Wriggers, *Error estimation for crack simulations using the xfem,* International Journal for Numerical Methods in Engineering **91**, 1459 (2012).

[45] R. Lins, M. Ferreira, S. Proença, and C. Duarte, *An a-posteriori error estimator for linear elastic fracture mechanics using the stable generalized/extended finite element method,* Computational Mechanics **56**, 947 (2015).

[46] I. Babuška and U. Banerjee, *Stable generalized finite element method (SGFEM),* Computer Methods in Applied Mechanics and Engineering **201-204**, 91 (2012).

[47] V. Gupta, C. Duarte, I. Babuška, and U. Banerjee, *A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **266**, 23 (2013).

[48] V. Gupta, C. Duarte, I. Babuška, and U. Banerjee, *Stable GFEM (SGFEM): Improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics,* Computer Methods in Applied Mechanics and Engineering **289**, 355 (2015).

[49] R. Lins, S. P. Proença, and C. A. Duarte, *Efficient and accurate stress recovery procedure and a posteriori error estimator for the stable generalized/extended finite element method,* International Journal for Numerical Methods in Engineering **119**, 1279 (2019).

[50] M. A. Schweitzer, *Variational mass lumping in the partition of unity method,* SIAM Journal on Scientific Computing **35**, A1073 (2013).

[51] A. Sharma and K. Maute, *Stress-based topology optimization using spatial gradient stabilized xfem,* Structural and Multidisciplinary Optimization **57**, 17 (2018).

[52] E. Burman and P. Hansbo, *Fictitious domain methods using cut elements: Iii. a stabilized nitsche method for stokes' problem,* ESAIM: Mathematical Modelling and Numerical Analysis **48**, 859 (2014).

[53] S. Soghrati, *Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations,* Journal of Computational Physics **275**, 41 (2014).

[54] A. M. Aragón, B. Liang, H. Ahmadian, and S. Soghrati, *On the stability and interpolating properties of the hierarchical interface-enriched finite element method,* Computer Methods in Applied Mechanics and Engineering **362**, 112671 (2020).

[55] E. De Lazzari, S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A critical view on the use of non-uniform rational b-splines to improve geometry representation in enriched finite element methods, International Journal for Numerical Methods in Engineering*, International Journal for Numerical Methods in Engineering **122**, 1195 (2021).

[56] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions,* International Journal for Numerical Methods in Engineering **120**, 1163 (2019).

[57] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *Cover image,* International Journal for Numerical Methods in Engineering **120**, i (2019).

[58] A. Nagarajan and S. Soghrati, *Conforming to interface structured adaptive mesh refinement: 3d algorithm and implementation,* Computational Mechanics **62**, 1213 (2018).

[59] A. C. Ramos, A. M. Aragón, S. Soghrati, P. H. Geubelle, and J.-F. Molinari, *A new formulation for imposing dirichlet boundary conditions on non-matching meshes,* International Journal for Numerical Methods in Engineering **103**, 430 (2015).

[60] K. Washizu, *Variational Methods in Elasticity and Plasticity* (Pergamon Press, 1982).

[61] D. R. Noble, E. P. Newren, and J. B. Lechman, *A conformal decomposition finite element method for modeling stationary fluid interface problems,* International Journal for Numerical Methods in Fluids **63**, 725 (2010).

[62] R. M. Kramer and D. R. Noble, *A conformal decomposition finite element method for arbitrary discontinuities on moving interfaces,* International Journal for Numerical Methods in Engineering **100**, 87 (2014).

[63] *A Conformal Decomposition Finite Element Method for Dynamic Wetting Applications*, Fluids Engineering Division Summer Meeting (2017).

[64] X. Jin, Z. Wang, Q. Zhou, L. M. Keer, and Q. Wang, *On the solution of an elliptical inhomogeneity in plane elasticity by the equivalent inclusion method,* Journal of Elasticity **114**, 1 (2014).

[65] E. Kirsch, *Die theorie der elastizität und die bedürfnisse der festigkeitslehre,* Zeitschrift des Vereines deutscher Ingenieure **42**, 797 (1898).



[66] F. Daneshmand and M. J. Kazemzadeh-Parsi, *Static and dynamic analysis of 2d and 3d elastic solids using the modified fgfem,* Finite Elements in Analysis and Design **45**, 755 (2009).

[67] M. Polajnar, F. Kosel, and R. Drazumeric, *Structural optimization using global stress-deviation objective function via the level-set method,* Structural and Multidisciplinary Optimization **55**, 91 (2017).

[68] J. Zhang, F. van Keulen, and A. M. Aragón, *On tailoring fracture resistance of brittle structures: A level set interface-enriched topology optimization approach,* Computer Methods in Applied Mechanics and Engineering **388**, 114189 (2022).

# 5

# On Tailoring Fracture Resistance of Brittle Structures: A Level Set Interface-enriched Topology Optimization Approach

*In this chapter we propose a fully immersed topology optimization procedure to design structures with tailored fracture resistance under linear elastic fracture mechanics assumptions for brittle materials. A level set function discretized by radial basis functions is used to represent the topology, and the Interface-enriched Generalized Finite Element Method (IGFEM) is applied to obtain an accurate structural response. Unlike most fracture-based topology optimization approaches, which predefine cracks at specific locations at the beginning, the technique assumes that cracks can nucleate at right angles from the boundary, at the location of enriched nodes that are added to enhance the finite element approximation. Instead of performing multiple finite element analyses to evaluate the energy release rates (ERRs) of all potential cracks—a procedure that would be computationally intractable—we approximate them by means of topological derivatives after a single enriched finite element analysis of the uncracked domain. ERRs are then aggregated to construct the objective function, and the corresponding sensitivity formulation is derived analytically by means of an adjoint formulation. Several numerical examples demonstrate the technique's ability to tailor fracture resistance, including the well-known benchmark L-shaped bracket and a multiple-loading optimization problem for obtaining a structure with fracture resistance anisotropy.*

This chapter has been published in *Computer Methods in Applied Mechanics and Engineering*, 388, 114189 (2022).

## 5.1. Introduction

Cracks in engineering structures, which could develop during their manufacturing or service life, may affect adversely the mechanical performance and even lead to catastrophic failure (particularly brittle fracture). When it is either impractical or just too expensive to simply change a defective component, a structure could be reconditioned by simply applying adhesively bonded patches [1–3]. Nevertheless, without any doubt the best way to mitigate the effect of cracks is preventing them from materializing in the first place. Structural failure due to fracture should therefore be considered in the early stages of structural design. At worst, the engineer should follow guidelines of best practices to avoid unsafe designs. At best, computational tools should be used to obtain designs that have been optimized to reduce the likelihood of fracture or other mechanisms that could compromise structural integrity. One such tool is topology optimization [4–7], which has become a popular design technique in real-world industrial applications [8, 9]. Based on structural failure criteria, topology optimization procedures can be classified into three categories: stress-, damage-, and fracture-based approaches.



Stress-based topology optimization is by far the most widely used technique. In this procedure, the stress state in the final design must fulfill a stress requirement, for instance, not exceeding the yield strength. Therefore, yield criteria such as von Mises [10–13] and Drucker Prager [14, 15] are enforced as constraints. Moreover, various static failure theories for brittle and ductile materials have also been used in topology optimization [16]. Since stress is a local quantity, a large number of stress constraints must be satisfied, making the optimization complex and computationally demanding [17, 18]. To circumvent this, one approach widely used is to employ aggregation functions that group all stress-based terms into a single quantity; aggregation functions include the Kreisselmeier-Steinhauser (KS) [19], $P$-mean [20], and $P$-norm [21]. As a caveat, although these stress measures simplify the computational implementation, they also "globalize" stress so local values are harder to enforce *exactly*. As a result, various techniques have been proposed to recover stress locality, including divide-and-conquer strategies, where aggregation functions are used on various disjoint subdomains [22, 23] and stress fields that are weighted by the structural boundary's curvature [24] (which significantly influences stress concentrations [25]). An alternative approach to the use of aggregation functions was recently proposed, whereby an augmented Lagrangian formulation is used to deal with a large number of local stress constraints in the objective function directly [26]. Instead of considering stresses as constraints, the stress distribution can also be directly minimized, for which the optimization requires a global stress measure [27–29].

Damage-based topology optimization has also gained considerable attention in the past decade. The idea, which was first explored by Bendsøe and Diaz [30], introduced damage-related criteria either as an objective or constraint. They used a continuum damage model to reinforce an existing structure for minimum damage. Another procedure based on continuum damage was later proposed to maximize the stiffness of concrete structures by optimizing the rebar layout [31]. Amir [32] then extended this method to design concrete structures with minimum weight, in which

both the rebar and concrete layout were optimized simultaneously. James and Waisman [33] also used the above damage model to optimize structures with minimum weight under the constraint on maximum local damage. They later used their approach to solve multiple-loading optimization problems, where superposition was used to consider the influence of damage accumulation [34]. All these studies focused on continuum damage models for brittle materials. Elastoplastic damage models were also investigated for designing energy absorbing structures [35–37], where the objective was to maximize plastic work while constraining maximum damage.

As strength and fracture toughness are usually inversely proportional to each other in many materials [38], fracture criteria have also been explored in structural optimization. Works in this category can be classified according to whether cracks are *stationary* or *evolving* (allowed to nucleate and/or propagate). In the former category cracks are therefore predefined in the computational domain and topology optimization is used to mitigate their effect. Kang *et al.* [39] borrowed linear elastic fracture mechanics (LEFM) concepts and considered the energy release rate (ERR)—amount of energy available for crack extension—evaluated by means of the $J$-integral [40] in their optimization. In fact, they solved a multi-objective optimization problem, where structural compliance and ERRs were considered as separate objective functions optimized simultaneously, yielding a Pareto set of solutions that revealed their trade-offs. Hu *et al.* [41] later followed a similar strategy to design structures including cracks at specific locations, where they used bi-directional evolutionary structural optimization (BESO) to update the structural topology and the extended/generalized finite element method (X/GFEM) for the structural analysis. As an alternative to avoid fracture failure by optimizing the topology of the structure with cracks, Klarbring *et al.* [3] optimized an adhesively bonded patch fixed to a structure, thereby minimizing the crack energy release rate. In addition to gradient-based optimization, improving fracture resilience in materials and structures has also been investigated with gradient-free algorithms [42].

Evolving cracks have also been studied in topology optimization, primarily by means of phase-field methods [43–45]. Xia *et al.* [46] maximized fracture resistance of quasi-brittle composites for multiple predefined cracks that were allowed to propagate using a phase-field approach. Russ and Waisman [47] used a phase-field approach within a SIMP-based topology optimization, and minimized the weight of a structure while putting a constraint on the fracture surface energy. The same authors later added the fracture surface energy to the objective function and weighted its priority with a scaling factor [48]. Da and Yvonnet [49] combined BESO with the phase-field method to design a composite material with improved fracture resistance by considering interfacial failure. Instead of using density-based methods, Wu *et al.* [50] incorporated a phase-field fracture model into the level set-based topology optimization to improve a structure's capability to withstand fracture; both crack initiation and propagation were considered during the optimization, which sought to optimize the reinforcement layout in two-phase composite materials. Afterwards, the same authors developed their methodology to include nonlinear finite element analysis (FEA) when designing structures with enhanced fracture resis-

tance [51]. The use of phase-field methods, however, is not without issues [45]. For instance, the length scale parameter used to define the transition zone tends to become small, requiring a very fine mesh especially at the vicinity of a crack. Accurate phase-field models are therefore computationally demanding.

For designing a structure that minimizes the likelihood of fracture, it would be desired to consider the possibility of cracks nucleating anywhere in the computational design domain. However, the design of structures with enhanced fracture resistance, which considers cracks nucleating at many locations under LEFM assumptions, has only been scarcely explored to date. This is mainly because of the vast computational demands required for such a design approach: A single TO iteration would have to evaluate every potential crack nucleating at multiple locations in the solid domain, for instance, by means of FEA with special finite element (FE) meshes tailored to resolve accurately the cracks' singular stress fields. To complicate things further, every potential crack could have an arbitrary orientation. Then ERRs would have to be computed for every potential crack, for example, by means of the $J$-integral. As a result, such an approach to design is simply intractable. To cope with this limitation, some studies have relied heavily on simplifying assumptions. In shape optimization, Jones *et al.* [52] proposed a modified biological algorithm to obtain the optimal shape of a hole under the assumption that cracks could nucleate only at right angles from it; stress intensity factors (SIFs) were evaluated by the finite element alternating technique, whereby a single FEA of the uncracked body is used together with analytical functions for any crack in a post-processing step [53]. Later, Das *et al.* [54] presented a modified evolutionary structural optimization (ESO) algorithm to optimize the shape of structures for maximum fracture resistance, also allowing cracks nucleating along the boundary but with SIFs calculated using an analytical expression derived by Kujawski [55]. Regarding topology optimization, Challis *et al.* [56] used a failure model that considered *pseudo-cracks* initiating at element nodes along the domain boundary. The objective function for fracture resistance is derived from the change of structural compliance at these locations using the "*virtual crack extension*" technique, whereby element nodes are moved in the direction of the inward normal to the boundary to mimic crack nucleation. They acknowledge that such objective function is close to only considering the energy density of each node in tension along the boundary, which results in similar optimized designs as those obtained from compliance minimization in most cases. To date, no work has attempted to conduct topology optimization under LEFM assumptions, where cracks are allowed to nucleate at many locations in the computational domain.

In this work we propose an LEFM-based topology optimization procedure to design structures with tailored fracture resistance by optimizing an aggregation of ERRs. The methodology builds on our previous work in the context of compliance minimization [57], where we use a level set function to describe topology and the Interface-enriched Generalized Finite Element Method (IGFEM) to analyze the structural response [58]. Cracks are allowed to nucleate perpendicularly to solid-void interfaces, at the location of enriched nodes that are added in IGFEM to resolve the displacement field accurately. ERRs of all potential cracks are obtained

by means of topological derivatives [59], for which a *single* FEA of the *uncracked* domain is required. Consequently, topological derivatives are the key ingredient to make the topology optimization tractable. Since an accurate stress field is required to obtain ERRs, we use a stress recovery technique for low-order finite elements to smoothen the stress field [60–62], and thus avoid the stress overestimation that usually arises in enriched FEM [63]. In this optimization framework, ERRs calculated at each potential crack are aggregated into a single term using the $P$-mean function [20]. In addition, an alternative formulation based on von Mises stresses is proposed to design structures with uniform stress distribution. Their corresponding analytical sensitivity formulations are derived by using an adjoint formulation. The method of moving asymptotes (MMA) is set as the optimizer to update the design variables [64]. The capability of the proposed technique is showcased on various examples that tailor fracture resistance. First, we solve the shape optimization problem of a square design domain with a hole in the center under biaxial tension; we show the optimal design of a circular hole with a uniform distribution of energy release rates is obtained. Then, the topology of the L-shaped bracket is optimized, where the sharp re-entrant corner in the initial design is removed and a smooth round corner emerges in the optimized design. We compare this result with that of a von Mises stress minimization problem aimed at obtaining a uniform stress distribution. The latter's optimized result also eliminates the sharp re-entrant corner with the stress concentration. Finally, we obtain an optimized design for maximum fracture resistance anisotropy by solving a multiple loading optimization problem; the objective of this problem is to maximize energy release rates when the domain is compressed horizontally, while simultaneously minimize them when compressed vertically.



## 5.2. Formulation

Consider an open bounded domain $\Omega \subset \mathbb{R}^2$ defined within a fixed background domain $\Delta$. As shown in Figure 5.1, the domain $\Omega$ is composed of isotropic linear elastic solid material with Young's modulus $E_1$ and Poisson's ratio $\nu_1$, and its smooth boundary consists of two-non-overlapping regions $\Gamma_D$ and $\Gamma_N$, where Dirichlet boundary conditions $\bar{\boldsymbol{u}}$ and surface tractions $\bar{\boldsymbol{t}}$ are prescribed, respectively. Boundary $\Gamma_N$ is free to move during the optimization except for a region $\bar{\Gamma}_N \subset \Gamma_N$ that remains fixed together with $\Gamma_D$. We denote $\overline{\Omega}$ as a closure of the domain $\Omega$ and $\Delta \backslash \Omega$ as the part of the domain occupied by void with with Young's modulus $E_2$ and Poisson's ratio $\nu_2$.

The displacement field $\boldsymbol{u}$ is a unique solution to the boundary value problem that describes the static equilibrium and the corresponding boundary conditions:

$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{b} & \text{in } \Omega, \\ \boldsymbol{u} = \bar{\boldsymbol{u}} & \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}} & \text{on } \Gamma_N, \end{cases} \tag{5.1}$$

where $\nabla\cdot$ represents the divergence operator, $\boldsymbol{\sigma}$ is Cauchy's stress tensor, $\boldsymbol{b}$ denotes the body force, and $\boldsymbol{n}$ is the outward normal to the boundary $\partial\Omega$. Under the assumption of linear elastic material behavior, the stress tensor $\boldsymbol{\sigma}$ is related to strain

$\boldsymbol{\varepsilon}$ by the Hooke's law as $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u})$, where $\mathbf{C}$ is the constitutive tensor, and $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathrm{T}}\right)$ is the infinitesimal strain tensor.

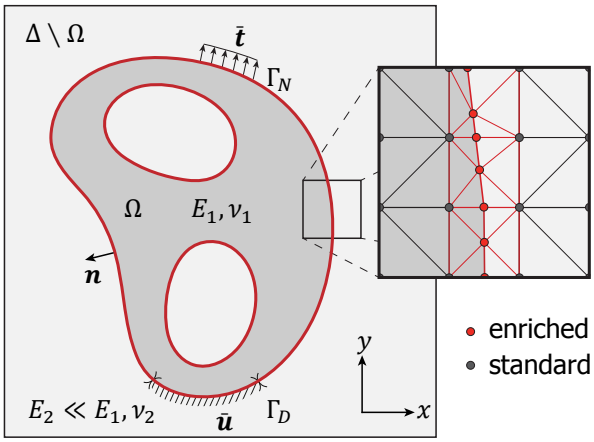



Figure 5.1: A solid domain $\Omega$ with a smooth boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ defined in a fixed background domain $\Delta$. Dirichlet boundary conditions are prescribed on $\Gamma_D$, and surface tractions are prescribed on $\Gamma_N$. For the discretized model, enriched nodes (marked with red circles) are created at intersections between $\partial\Omega$ and the edges of mesh elements. The integration elements are created near the boundary (marked with red triangles).

The weak formulation of this linear elasticity problem is: Find $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}, \tag{5.2}$$

where $\boldsymbol{\mathcal{U}}$ is the vector-valued set of kinematically admissible displacement fields and $\boldsymbol{\mathcal{V}}$ the vector-valued space of weight functions that satisfy homogeneous essential boundary conditions on $\Gamma_D$. The bilinear $a(\boldsymbol{u}, \boldsymbol{v})$ and linear $\ell(\boldsymbol{v})$ forms, which also represent the virtual work of internal and external forces, respectively, are given by

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, \mathrm{d}\Omega, \tag{5.3}$$

and

$$\ell(\boldsymbol{v}) = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{b} \, \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{v} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma. \tag{5.4}$$

In order to solve the problem above, the design domain $\Delta$ is discretized by a finite element mesh $\Delta^h = \mathrm{int}(\cup_i e_i)$, where $e_i$ is the $i$th finite element and $\mathrm{int}(\cdot)$ represents set. The finite-dimensional form of Equation (5.2) is then obtained as

$$\sum_i \int_{e_i} \boldsymbol{\varepsilon}(\boldsymbol{u}^h) : \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{v}^h) \, \mathrm{d}\Omega = \sum_i \int_{e_i} \boldsymbol{v}^h \cdot \boldsymbol{b} \, \mathrm{d}\Omega + \sum_i \int_{\partial e_i \cap \Gamma_N} \boldsymbol{v}^h \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma, \qquad \forall \boldsymbol{v}^h \in \boldsymbol{\mathcal{V}}^h \tag{5.5}$$

where $\boldsymbol{u}^h \in \mathcal{U}^h$ and $\boldsymbol{v}^h \in \mathcal{V}^h$ are the trial solution and weight function, respectively.

In the Interface-enriched Generalized Finite Element Method (IGFEM), the interaction between $\Delta^h$ and $\Gamma$ creates new (enriched) nodes (marked with red circles in Figure 5.1) at intersections between element edges and the boundary. Afterwards, cut elements in $\Delta^h$ are split into integration elements. The displacement field $\boldsymbol{u}^h$ is given by

$$\boldsymbol{u}^h = \underbrace{\sum_{i \in \iota_h} N_i(\boldsymbol{x}) \boldsymbol{U}_i}_{\text{standard FEM}} + \underbrace{\sum_{i \in \iota_w} \psi_i(\boldsymbol{x}) \boldsymbol{\alpha}_i}_{\text{enrichment}} . \tag{5.6}$$

In this equation, the first term is the standard FEM approximation, where $\iota_h$ denotes the index set of all nodes in $\Delta^h$ from the original background mesh (marked with black circles in Figure 5.1), and $N_i$ and $\boldsymbol{U}_i$ are the Lagrange shape function and degrees of freedom (DOFs) of the $i$th mesh node. In the second enrichment term, $\iota_w$ represents the index set of enriched nodes, and $\psi_i$ is the enrichment function associated with corresponding enriched DOFs $\boldsymbol{\alpha}_i$.

Regarding the evaluation of the local stiffness matrix $\boldsymbol{k}_e$ and force vector $\boldsymbol{f}_e$, elements that are not intersected follow standard FEM procedures. For integration elements, following a standard isoparametric procedure, $\boldsymbol{k}_e$ and $\boldsymbol{f}_e$ can be evaluated as

$$\boldsymbol{k}_e = \int_e \boldsymbol{B}^\intercal \mathbf{C} \boldsymbol{B} j_e \, \mathrm{d}\boldsymbol{\xi} , \quad \text{and} \quad \boldsymbol{f}_e = \int_e \begin{bmatrix} \boldsymbol{N} \\ \boldsymbol{\psi} \end{bmatrix} \boldsymbol{b} j_e \, \mathrm{d}\boldsymbol{\xi} + \int_{e \cap \Gamma_N} \begin{bmatrix} \boldsymbol{N} \\ \boldsymbol{\psi} \end{bmatrix} \bar{\boldsymbol{t}} j_e \, \mathrm{d}\partial\boldsymbol{\xi} \tag{5.7}$$

where $\boldsymbol{\xi}$ is the master coordinate, $\boldsymbol{N}$ and $\boldsymbol{\psi}$ are vectors that stack the element's standard shape functions and enrichment functions, respectively, $j_e$ is the Jacobian determinant (for the integration element's transformation), and $\boldsymbol{B} = \boldsymbol{\Delta}_{\boldsymbol{\xi}}^\intercal \begin{bmatrix} \boldsymbol{N}^\intercal \boldsymbol{J}^{-\intercal} & \boldsymbol{\psi}^\intercal \boldsymbol{J}_e^{-\intercal} \end{bmatrix}$ is the strain-displacement matrix, where $\boldsymbol{J}_e^{-\intercal}$ and $\boldsymbol{J}^{-\intercal}$ are the transpose of the inverse of the Jacobian of the isoparamatric mapping for the integration and parent elements, respectively, and the differential operator $\boldsymbol{\Delta}_{\boldsymbol{\xi}}$ is given by

$$\boldsymbol{\Delta}_{\boldsymbol{\xi}} \equiv \begin{bmatrix} \frac{\partial}{\partial \xi_1} & 0 & \frac{\partial}{\partial \xi_2} \\ 0 & \frac{\partial}{\partial \xi_2} & \frac{\partial}{\partial \xi_1} \end{bmatrix}^\intercal . \tag{5.8}$$

Considering the contribution of all elements in the discretization, the global stiffness matrix $\boldsymbol{K}$ and force vector $\boldsymbol{F}$ are given by

$$\boldsymbol{K} = \mathop{\mathbb{A}}_i \boldsymbol{k}_i , \qquad \boldsymbol{F} = \mathop{\mathbb{A}}_i \boldsymbol{f}_i , \tag{5.9}$$

where $\mathbb{A}$ is the standard finite element assembly operator. For more details about IGFEM's formulation see references [58, 65].

Similarly to X/GFEM, the complexity of creating a fitted discretization in standard FEM is transferred in IGFEM to the enriched formulation, which requires advanced computational geometry operations for intersecting a usually structured background finite element mesh with the discontinuities. However, although IGFEM retains the

main feature of X/GFEM, it also keeps the attractive properties of standard FEM: Since enrichment functions are constructed with Lagrange shape functions of integration elements, their value is exactly zero at original mesh nodes (this property requires shifting in X/GFEM [66, 67]). This means that DOFs associated with background mesh nodes represent the displacement at their corresponding location, thus keeping their physical interpretation. Moreover, essential boundary conditions on discontinuities can be prescribed strongly after solving a local problem or via multiple point constraints (MPCs) [68]. For instance, as shown in Figure 5.2(a), an original element $e$ with $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ is cut by a material interface (marked with a red segment), where enriched nodes $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ are created. In order to impose the displacement field $\bar{\boldsymbol{u}}$ on nodes $\boldsymbol{x}_1, \boldsymbol{x}_2$ and $\boldsymbol{x}_4$, we then obtain the following formulation based on Equation (5.6)

$$\boldsymbol{\alpha}_4 = \bar{\boldsymbol{u}}(\boldsymbol{x}_4) - N_1(\boldsymbol{x}_4)\bar{\boldsymbol{u}}(\boldsymbol{x}_1) - N_2(\boldsymbol{x}_4)\bar{\boldsymbol{u}}(\boldsymbol{x}_2). \tag{5.10}$$

This equation shows that the displacement boundary condition is imposed on enriched DOFs in a strong form, which is the same as that in the standard FEM.



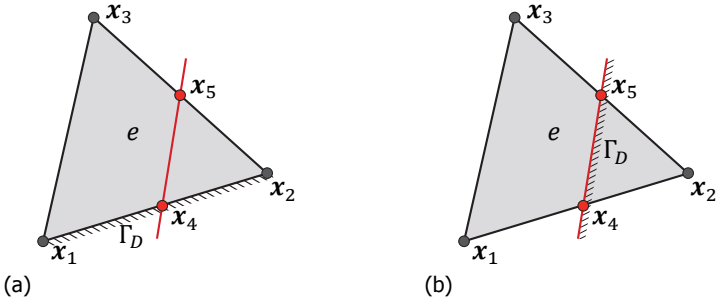



Figure 5.2: Two enriched nodes $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ are created as the background mesh element with connectivity $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ is cut by a material interface (marked with a red segment), where Dirichlet boundary conditions are imposed on line elements containing (a) both enriched and original nodes, and (b) only enriched nodes.

Considering the same cut element, Figure 5.2(b) displays another scenario where the displacement $\bar{\boldsymbol{u}}$ is only prescribed on enriched nodes $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ using MPCs. According to Equation (5.6), displacements of enriched DOFs can be expressed as

$$\begin{aligned} \boldsymbol{\alpha}_4 &= \bar{\boldsymbol{u}}(\boldsymbol{x}_4) - N_1(\boldsymbol{x}_4)\boldsymbol{U}_1 - N_2(\boldsymbol{x}_4)\boldsymbol{U}_2, \\ \boldsymbol{\alpha}_5 &= \bar{\boldsymbol{u}}(\boldsymbol{x}_5) - N_2(\boldsymbol{x}_5)\boldsymbol{U}_2 - N_3(\boldsymbol{x}_5)\boldsymbol{U}_3. \end{aligned} \tag{5.11}$$

These two equations can be rewritten in the matrix form as

$$\underbrace{\begin{bmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \\ \boldsymbol{U}_3 \\ \boldsymbol{\alpha}_4 \\ \boldsymbol{\alpha}_5 \end{bmatrix}}_{\boldsymbol{U}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -N_1(\boldsymbol{x}_4)\cdot\mathbf{I} & -N_2(\boldsymbol{x}_4)\cdot\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -N_2(\boldsymbol{x}_5)\cdot\mathbf{I} & -N_3(\boldsymbol{x}_5)\cdot\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{bmatrix} \tilde{\boldsymbol{U}}_1 \\ \tilde{\boldsymbol{U}}_2 \\ \tilde{\boldsymbol{U}}_3 \\ \tilde{\boldsymbol{\alpha}}_4 \\ \tilde{\boldsymbol{\alpha}}_5 \end{bmatrix}}_{\tilde{\boldsymbol{U}}} + \underbrace{\begin{bmatrix} \vec{0} \\ \vec{0} \\ \vec{0} \\ \bar{\boldsymbol{u}}(\boldsymbol{x}_4) \\ \bar{\boldsymbol{u}}(\boldsymbol{x}_5) \end{bmatrix}}_{\boldsymbol{g}}, \tag{5.12}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{0}$ is a zero matrix with the same dimension, and $\vec{0}$ is a vector with two zero components. $\boldsymbol{T}$ is a transformation matrix containing the relation between original DOFs $\boldsymbol{U}$ and new DOFs $\tilde{\boldsymbol{U}}$, and $\boldsymbol{g}$ is a vector of the prescribed values $\bar{\boldsymbol{u}}$. In this case, the original equilibrium equation $\boldsymbol{KU} = \boldsymbol{F}$, where $\boldsymbol{K}$ and $\boldsymbol{F}$ are the global stiffness matrix and force vector, respectively, is modified to $\tilde{\boldsymbol{K}}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{F}}$. $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{F}}$ are expressed as

$$\tilde{\boldsymbol{K}} = \boldsymbol{T}^{\intercal}\boldsymbol{K}\boldsymbol{T}, \tag{5.13}$$

and

$$\tilde{\boldsymbol{F}} = \boldsymbol{T}^{\intercal}(\boldsymbol{F} - \boldsymbol{K}\boldsymbol{g}), \tag{5.14}$$

respectively. If displacement boundary conditions are only prescribed on original mesh nodes, then $\boldsymbol{T}$ is an identity matrix and $\boldsymbol{g}$ is a zero vector, which leads to $\tilde{\boldsymbol{K}} = \boldsymbol{K}$ and $\tilde{\boldsymbol{F}} = \boldsymbol{F}$. The transformation to the stiffness matrix given by Equations (5.13) and (5.14) will change the sparsity of the global stiffness matrix. Consider in Figure 5.3(a) a background mesh with $3 \times 3 \times 2$ linear triangular elements that is intersected by an interface (marked with red), where Dirichlet boundary conditions are prescribed. The change in sparsity before and after applying MPCs is shown in Figures 5.3(b) and (c), where non-zero terms (marked with blue squares) show that no new terms are generated after the transformation.

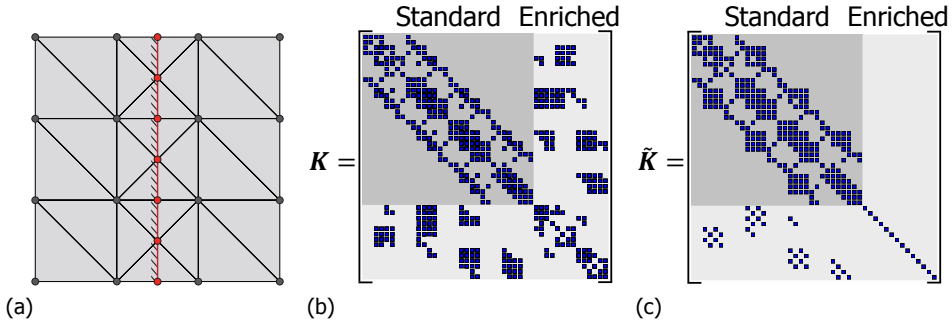



Figure 5.3: (a) A background mesh with $3 \times 3 \times 2$ triangular elements intersects an interface (marked with red), where essential boundary conditions are prescribed; (b, c) Non-zeros terms (marked with blue squares) of the global stiffness matrix before (b) and after (c) applying multiple-point constrains.

## 5.3. Topology optimization formulation

By means of topology optimization, we seek to solve the following problem

$$\begin{aligned} &\text{minimize } J(\boldsymbol{s}) \\ &\text{subject to } \tilde{\boldsymbol{K}}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{F}}, \\ &V_{\mathrm{s}} \leq V_{\mathrm{c}}. \end{aligned} \tag{5.15}$$

where $J(\boldsymbol{s})$ is an objective function associated with design variable $\boldsymbol{s}$, and more details will be explained later. $V_{\mathrm{s}}$ is the volume occupied by solid material, and $V_{\mathrm{c}}$ is its maximum allowed value.

The objective function $J(\boldsymbol{s})$ simplified as $J$ is an aggregation function in $P$-mean form that collects energy release rates of all potential cracks, and it is defined as

$$J = \left( \frac{1}{N_{\text{node}}} \sum_{i=1}^{N_{\text{node}}} G_i^p \right)^{\frac{1}{p}}, \tag{5.16}$$

where $p$ is an integer exponent, $N_{\text{node}}$ is the number of nodes along external and internal boundaries that are under tension, and $G_i$ is the energy release rate at the $i$th enriched node.

### 5.3.1. Topology description

In this work the location of the boundary $\Gamma$ is represented by a level set function $\phi$, which was first introduced to structural topology optimization by Sethian and Wiegmann [69]. This implicit function is defined as

$$\begin{aligned} \phi(\boldsymbol{x}) &= 0 \quad \forall \boldsymbol{x} \in \Gamma, \\ \phi(\boldsymbol{x}) &< 0 \quad \forall \boldsymbol{x} \in \overline{\Omega}, \\ \phi(\boldsymbol{x}) &> 0 \quad \forall \boldsymbol{x} \in \Delta \backslash \Omega. \end{aligned} \tag{5.17}$$

Compactly supported radial basis functions (RBFs) are used to interpolate the level set function for a number of reasons [70]: *i)* The smoothness of RBFs results in a smooth level set function; *ii)* By increasing the support area of RBFs, each design variable (weight associated with the RBF) has a higher influence on the level set function and thus in the evolution of the material boundary, which can make the optimization process converge faster; *iii)* As the RBF grid is decoupled from the finite element discretization of the domain, the design space dimensionality and the background mesh size can be set independently. The level set function $\phi(x)$ is then expressed as

$$\phi(\boldsymbol{x}) = \boldsymbol{Z}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{s} = \sum_{i=1}^{N_k} \zeta_i(\boldsymbol{x}) s_i, \tag{5.18}$$

where $N_k$ is the number of RBFs, $\zeta_i(\boldsymbol{x})$ is $i$th compactly supported RBF, and $s_i$ is the corresponding expansion coefficient. $\boldsymbol{Z}(\mathbf{x})$ is a matrix including all RBFs:

$$\boldsymbol{Z}(\mathbf{x}) = \left[ \zeta_1(\mathbf{x}), \zeta_2(\mathbf{x}), \dots, \zeta_{N_k}(\mathbf{x}) \right]^{\mathsf{T}}, \tag{5.19}$$

and $\boldsymbol{s}$ is a vector of the corresponding expansion coefficient:

$$\boldsymbol{s} = \left[ s_1, s_2, \dots, s_{N_k} \right]^{\mathsf{T}}. \tag{5.20}$$

In this work we adopt $\zeta_i(\boldsymbol{x})$ with $C^2$ smoothness [71]:

$$\zeta_i(\boldsymbol{x}) = \max\left(0, 1 - r_i(\boldsymbol{x})\right)^4 \left(4 r_i(\boldsymbol{x}) + 1\right), \tag{5.21}$$

where the scaling parameter $r_i(\boldsymbol{x})$ is defined as:

$$r_i(\boldsymbol{x}) = \frac{\sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|}}{r_s}. \tag{5.22}$$

In the above equation, $\sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|}$ is the distance between coordinate $\boldsymbol{x}$ and the center $\boldsymbol{x}_i$ of $i$th RBF, and $r_s$ is the radius of support.

Regarding the update procedure for the level set function, mathematical programming algorithms such as sequential quadratic programming (SQP) [72] and the method of moving asymptotes (MMA) [64], become increasingly popular in level set-based topology optimization [57, 73–76]. Because our level set function is parameterized using compactly supported RBFs, it is readily suitable for incorporation in mathematical schemes because of the explicit sensitivities [74]. Therefore, MMA is used to update the design variables, and the optimization terminates when reaching a given maximum number of iterations.

It is worth noting that our procedure is not standard, since the solution of the Hamilton Jacobi (H-J) equation has traditionally governed the evolution of the topology for level set-based topology optimization procedures [77, 78]. However, this first-order partial differential equation is solved by explicit methods with upwind schemes [79, 80], where the time step must satisfy the Courant Friedrichs Lewy (CFL) condition for stability and convergence [81]. As a result, updating level set function requires more iterations than our approach, with the consequent increase of computational resources—in addition to the increase in computational time required to solve the H-J equation in each iteration. Moreover, it is necessary to extend the velocity field from the structural boundary to the whole design domain or at least to a narrow band along the boundary [69]. Finally, regularization, which requires solving another H-J equation, should be integrated into the update procedure for obtaining accurate optimized results [82].



### 5.3.2. Evaluation of energy release rate

Based on LEFM considerations, the energy release rate $G$ is defined as

$$G = \frac{1}{\overline{E}_1}\left(K_{\mathrm{I}}^2 + K_{\mathrm{II}}^2\right), \tag{5.23}$$

where $\overline{E}_1 = E_1/\left(1-\nu_1^2\right)$ for plane strain and $\overline{E}_1 = E_1$ for plane stress. $K_{\mathrm{I}}$ and $K_{\mathrm{II}}$ are the stress intensity factors under modes I and II, respectively. According to Silva *et al.* [59], $K_{\mathrm{I}}$ and $K_{\mathrm{II}}$ can be computed for an infinitesimal crack at a location $\boldsymbol{x}_i = (x_i, y_i)$ along the boundary $\Gamma$ as

$$\begin{bmatrix} K_{\mathrm{I}}(\boldsymbol{x}_i,\eta,\gamma,\beta) \\ K_{\mathrm{II}}(\boldsymbol{x}_i,\eta,\gamma,\beta) \end{bmatrix} = \sqrt{\pi\eta}\underbrace{\begin{bmatrix} h_{11}(\gamma) & h_{12}(\gamma) \\ h_{21}(\gamma) & h_{22}(\gamma) \end{bmatrix}}_{\boldsymbol{H}(\gamma)}\begin{bmatrix} \sigma_{\theta\theta}(\boldsymbol{x}_i,\gamma,\beta) \\ \sigma_{r\theta}(\boldsymbol{x}_i,\gamma,\beta) \end{bmatrix}, \tag{5.24}$$

where, as illustrated in Figure 5.4, $\eta$ is the crack length, $\gamma$ is the angle between the crack and the inward normal of the local boundary, and $\beta$ is the angle between the

global coordinate system and a local coordinate system located at $\boldsymbol{x}_i$. As $h_{ij}(\gamma), i,j \in \{1,2\}$ is a complicated polynomial function associated with angle $\gamma$ [83], its definition is given in Appendix A.3. $\sigma_{\theta\theta}$ and $\sigma_{r\theta}$ are stress components defined under the polar coordinate system $(r,\theta)$. Equation (5.23) can then be rewritten by replacing $K_{\mathrm{I}}$ and $K_{\mathrm{II}}$ with Equation (5.24) as

$$G(\boldsymbol{x}_i,\eta,\gamma,\beta) = \frac{\pi\eta}{\overline{\overline{E}}_1}\begin{bmatrix}\sigma_{\theta\theta}(\boldsymbol{x}_i,\gamma,\beta)\\ \sigma_{r\theta}(\boldsymbol{x}_i,\gamma,\beta)\end{bmatrix}^{\mathsf{T}} \boldsymbol{H}^{\mathsf{T}}(\gamma)\boldsymbol{H}(\gamma)\begin{bmatrix}\sigma_{\theta\theta}(\boldsymbol{x}_i,\gamma,\beta)\\ \sigma_{r\theta}(\boldsymbol{x}_i,\gamma,\beta)\end{bmatrix}. \tag{5.25}$$

The energy release rate $G(\boldsymbol{x}_i,\eta,\gamma,\beta)$ as a function of stress $\boldsymbol{\sigma} = \begin{bmatrix}\sigma_{xx} & \sigma_{xy} & \sigma_{xy} & \sigma_{yy}\end{bmatrix}^{\mathsf{T}}$ in global coordinates is

$$G(\boldsymbol{x}_i,\eta,\gamma,\beta) = \frac{\pi\eta}{\overline{\overline{E}}_1}\boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathsf{T}}\boldsymbol{Q}(\gamma,\beta)\boldsymbol{\sigma}(\boldsymbol{x}_i), \tag{5.26}$$

where $\boldsymbol{Q}(\gamma,\beta)$ accounts for the transformation between coordinate systems (see details in Appendix A.4).



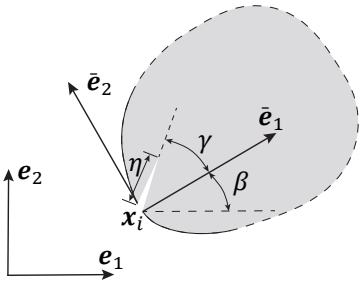



Figure 5.4: The illustration of a crack with length $\eta$ nucleating at node $\boldsymbol{x}_i$, where $\gamma$ is the angle between this crack and the internal normal of structural boundary, and $\beta$ is the angle between the global and local coordinate systems.

Noteworthy, the results obtained with Equation (5.24) are in good agreement with those evaluated via finite element analyses when crack size is smaller than 5% of the domain size [59]. In addition, the accuracy of energy release rates obtained by topological derivatives can be increased by considering high-order terms into the formulation when treating longer cracks [84].

### 5.3.3. Stress calculation

Because the stress field is obtained by applying the gradient to the displacement field, directly computed stresses from linear finite element solutions are piece-wise constant. Since the stress field plays a critical role in the evaluation of the energy release rate, it is important to use an appropriate method to approximate it. Even though enriched finite element techniques yield more accurate solutions than those obtained by the Ersatz material approach [85], they could also yield poor approximation for the stress field when interfaces get arbitrarily close to background mesh nodes.

To circumvent the issue, several pre- and post-processing techniques have been proposed. One approach is to eliminate small elements by merging nodes in close proximity into a single node [86–88]. This approach can totally prevent the overestimation of the stress, but it requires meddling with the mesh or the interfaces. Another strategy smoothens the stress of an element with small areas by taking its neighboring elements into account. For instance, an average-weighted formulation considering element areas can be used to post-process the stress of these tiny elements [89].!Although this method can provide quite good approximations [90], it does not avoid the overestimation of the stress completely.!Yet another technique interpolates the stress field by using stress recovery approaches, such as the superconvergent patch recovery technique [87, 91].!Even though sensitivities of the objective function with respect to design variables become intricate, the evaluation of the recovered stress field is more accurate than other techniques. Therefore, the stress recovery technique proposed in Chapter 4 is adopted to smoothen the stress field.

### 5.3.4. Sensitivity analysis

The sensitivity of the objective function $J$ with respect to design variables $\boldsymbol{s}$ is derived by using the adjoint variable method. A Lagrangian function of the objective, which is constructed by using the adjoint vector $\boldsymbol{\lambda}$, is expressed as $L = J + \boldsymbol{\lambda}^{\mathsf{T}}(\tilde{\boldsymbol{K}}\tilde{\boldsymbol{U}} - \tilde{\boldsymbol{F}})$ Then the derivative of $L$ with respect to the $j$th design variable $s_j$ is given by

$$\begin{aligned}\frac{\mathrm{d}L}{\mathrm{d}s_j} &= \frac{\partial J}{\partial s_j} + \frac{\partial J}{\partial \tilde{\boldsymbol{U}}}\frac{\partial \tilde{\boldsymbol{U}}}{\partial s_j} + \boldsymbol{\lambda}^{\mathsf{T}}\left(\frac{\partial\left(\tilde{\boldsymbol{K}}\tilde{\boldsymbol{U}}\right)}{\partial s_j} - \frac{\partial \tilde{\boldsymbol{F}}}{\partial s_j}\right) \\ &= \frac{\partial J}{\partial s_j} + \left(\frac{\partial J}{\partial \tilde{\boldsymbol{U}}} + \boldsymbol{\lambda}^{\mathsf{T}}\tilde{\boldsymbol{K}}\right)\frac{\partial \tilde{\boldsymbol{U}}}{\partial \boldsymbol{s}_j} + \boldsymbol{\lambda}^{\mathsf{T}}\left(\frac{\partial \tilde{\boldsymbol{K}}}{\partial s_j}\tilde{\boldsymbol{U}} - \frac{\partial \tilde{\boldsymbol{F}}}{\partial s_j}\right).\end{aligned} \tag{5.27}$$

In order to obtain the adjoint vector $\boldsymbol{\lambda}$, the following adjoint equation is solved:

$$\frac{\partial J}{\partial \tilde{\boldsymbol{U}}} + \boldsymbol{\lambda}^{\mathsf{T}}\tilde{\boldsymbol{K}} = 0. \tag{5.28}$$

Details about the evaluation of $\partial J/\partial s_j$, and $\partial J/\partial \tilde{\boldsymbol{U}}$ are given in Appendix A.5.1. $\partial \tilde{\boldsymbol{K}}/\partial s_j$ and $\partial \tilde{\boldsymbol{F}}/\partial s_j$ are explained in detail in Appendix A.5.2.

## 5.4. Numerical examples

In this section, several numerical examples are investigated to demonstrate the capability of the proposed approach in obtaining designs with tailored fracture resistance. The crack length is set to 1% of the domain size to ensure the validity of Equation (5.24). No units are given to material properties, tractions, nor domain dimensions, so results can be interpreted in any consistent unit system. If not explicitly specified, Young's moduli are taken as $E_{\mathrm{s}} = 1$ and $E_{\mathrm{v}} = 10^{-6}$ for solid and void materials, respectively. The Poisson's ratio for both materials is $\nu = 0.3$. Linear triangular elements are used to discretize the design domain—for which a single

gauss point is sufficient in standard and integration elements—and the finite element analysis is conducted under plane strain conditions. Topological gradient information, which has traditionally been used to nucleate holes in the interior of the design domain during topology optimization [92, 93], is not considered in the proposed procedure. Therefore, all examples start with an initial design seeded with holes.

### 5.4.1. Shape optimization

Firstly, a shape optimization example is studied, where unit magnitude tractions $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are applied on the sides of a $2 \times 2$ square domain as illustrated in Figure 5.5. Because the design region is immersed into a background domain, the tractions are prescribed by properly integrating the force vector in cut elements, as described in Chapter 5.2. Due to symmetry, only a quarter of the design domain is considered with symmetric boundary conditions. A value $p = 8$ is used in the objective function, and the volume of solid material is constrained at $V_s = 0.875$. A background mesh with $20 \times 20 \times 2$ linear triangular elements is used to discretize the bacground domain, and the maximum number of iterations is set to 100.



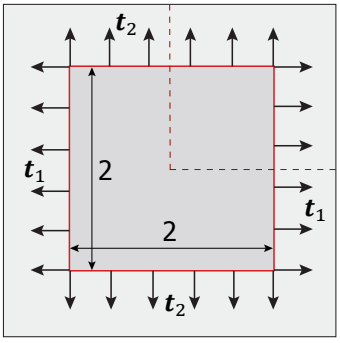



Figure 5.5: Schematic of the biaxial tension example with tractions $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ prescribed to edges of a square design domain as shown. The design domain is immersed into a larger background domain that is discretized by constant strain triangles. Due to symmetry, only a quarter of this domain is considered.

Figure 5.6(a) shows the initial design, which has a hole with triangular shape at the bottom-left corner. The corresponding optimized design with a quarter of a circular hole is displayed in Figure 5.6(b). Figures 5.6(a) and (b) also show their corresponding energy release rate distributions. From Figure 5.6(a), it can be seen that energy release rates around top and right vertices of the triangle are much higher than elsewhere because of the stress concentrations. However, energy release rates in the optimized design are uniform along the circular boundary.

The convergence for both the objective and material volume fraction is shown in Figure 5.7, where it can be seen that the former converges at around 50 iterations. Even though the volume of the optimized design is not the same as the constrained value, it still satisfies the constraint. Structures obtained at 10, 30, and 50 iterations are also displayed in this figure. Note that even though the volume

of the optimized design is not the same as the constrained value, the constraint is still satisfied; although the MMA optimizer minimizes the objective and satisfies the volume constraint in parallel, it assigns higher priority to the former as a relatively small Lagrange multiplier is associated with the latter. The inset in Figure 5.7 shows how the volume increases after 45 iterations.

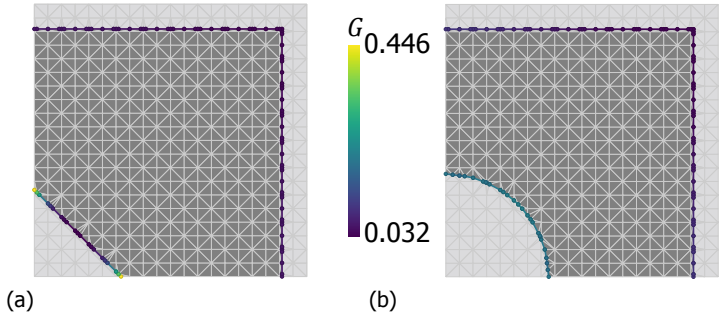



Figure 5.6: (a) Initial design with a triangular hole, where energy release rates around top and right vertices of the triangle are much higher than those at other places in the initial design; (b) Optimized design with a quarter of circle, where the distribution of energy release rates is uniform along the material interface in the optimized structure.



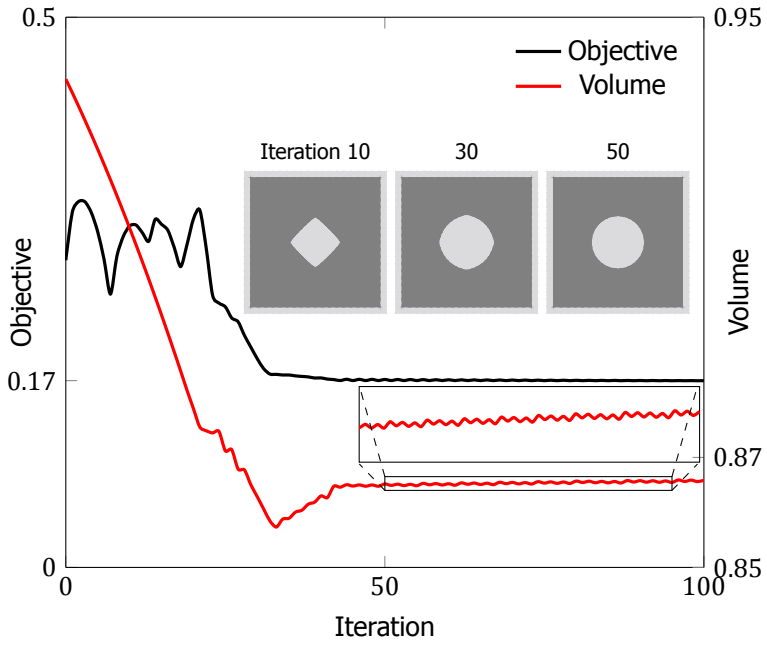



Figure 5.7: Convergence plot of the biaxial tension example displays that objective converges around after 50 iterations. Designs obtained after 10, 30, and 50 iterations are shown.

### 5.4.2. L-shaped bracket

We investigate the performance of the proposed method in optimizing the topology of an L-shaped bracket, domain that is also immersed in a square computational domain with dimensions $11 \times 11$. The latter is discretized with a background FE mesh composed of $100 \times 100 \times 2$ constant strain triangles. As shown in Figure 5.8, the area outside the L-shaped bracket, together with a small area in the vicinity of the load, is considered as a non-design domain. Homogeneous Dirichlet boundary

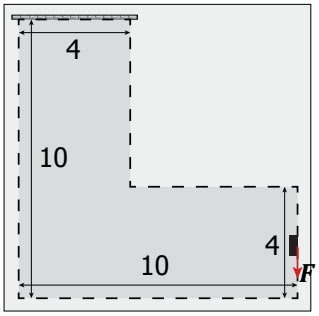
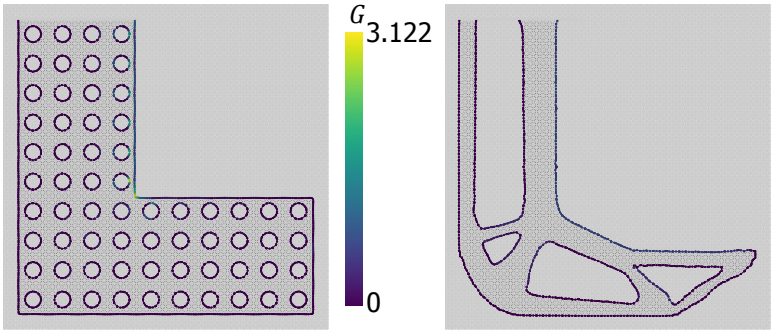



Figure 5.8: Schematic of the L-shaped bracket where the top part is fixed and a loading $\|\boldsymbol{F}\| = 0.25$ is applied on the middle of the right side.

condition are prescribed on the immersed top edge of the bracket, as discussed in Chapter 5.2, and a vertical downward load $\boldsymbol{F} = -1/4\,\boldsymbol{e}_2$ is applied on the middle of right side. A value $p = 8$ is used in the objective function, and the volume constraint is set to $V_c = 25.6$ (40% of the volume of the L-shaped bracket fully occupied by solid material). The maximum number of iterations is set to 500, and the initial design with many circular holes is shown in Figure 5.9(a).




(a) (b)

Figure 5.9: (a) Initial design and its corresponding distribution of energy release rate, and (b) Final design obtained with minimizing the energy release rate and its corresponding distribution.

The optimized structure is shown in Figure 5.9(b), and the corresponding energy release rate distribution is given in the same figure. It can be seen that the sharp in-

ternal corner in the initial design with high energy release rate has been removed. At this location, a rounded corner appears in the final result with a uniform distribution of energy release rate values. Figure 5.9(b) shows zigzagging of some edges in the optimized design where the level set function is not perfectly smooth. This approximation artifact, which actually improves the objective function, is caused by the discretization. This issue is discussed in detail in van den Boom *et al.* [57]. The convergence curves and optimized structures obtained at 50, 100, 200, and 300 steps are shown in Figure 5.10. Oscillations of the objective function are observed

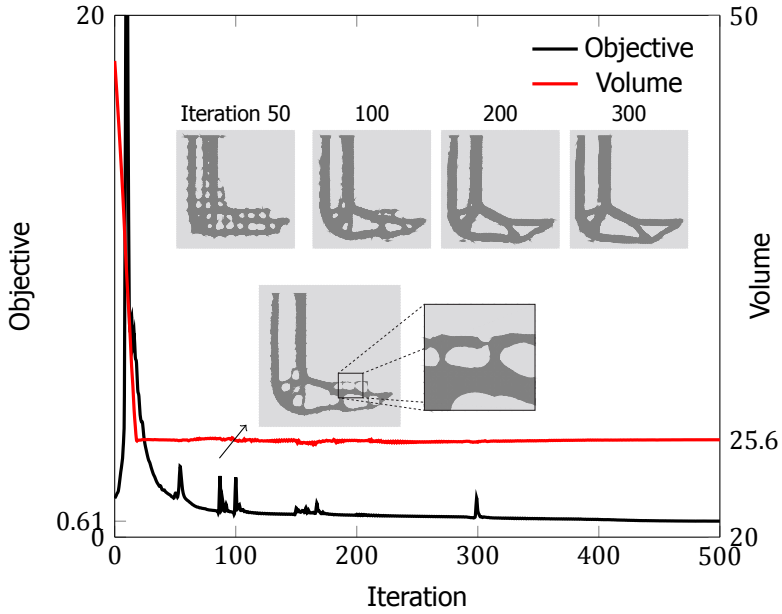
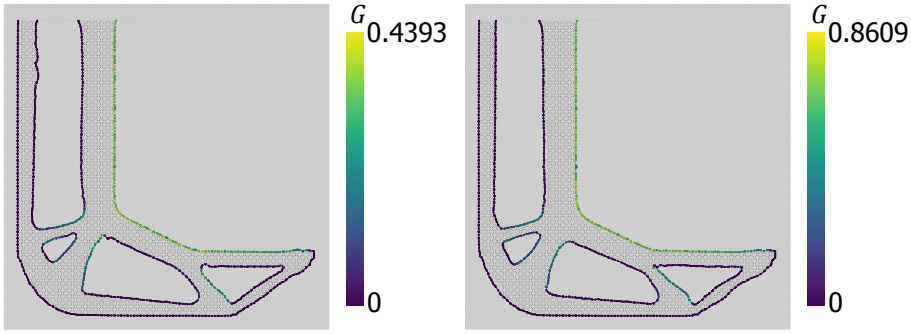



Figure 5.10: Convergence plot of objective and volume when optimizing the energy release rate; optimized designs obtained at 50, 100, 200, and 300 iterations; the design obtained after 87 iterations with a thin-wall structure.

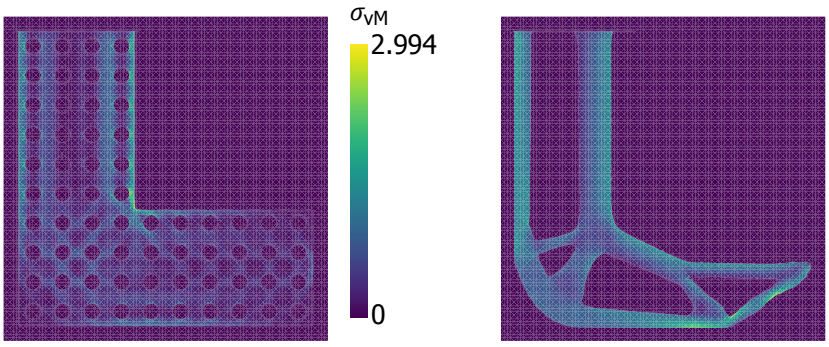
during the optimization, particularly at the beginning. Since energy release rates are obtained from the stress field, these oscillations are caused by elements with high stress values. Although the stress recovery technique is used to smoothen the stress field, avoiding stress overestimation caused by bad aspect ratios of integration elements, the merging of holes in the initial design is responsible for thin solid structures with high stresses. An optimized design obtained after setting the crack length to 0.5% of the domain size is also shown in Figure 5.11(a), where it can be seen that the final result has the same topology with slightly different shapes compared to the original optimized design of Figure 5.11(b). The maximum value of the energy release rate of the new design is almost half of that obtained by the original optimized structure—notice the linear relationship between the crack length and the energy release rate in Equations (5.25) and (5.26). It is worth noting that the optimization is very sensitive to the move limit of the MMA optimizer. If a

large move limit is used, the final design could be disconnected, which is actually beneficial to the objective function because stress is nil in isolated regions. A way to eliminate this effect is to consider the structural compliance as a constraint [94]. In addition, we also show the von Mises stress distribution of initial and optimized designs in Figures 5.12(a) and (b), respectively. It is worth noting that while there are locations with high stresses in the optimized structure, energy release rates are low and therefore the L-shaped bracket will not fracture.




(a) (b)

Figure 5.11: Final designs obtained assuming cracks with lengths of (a) 0.5% and (b) 1% of the domain size. The same structural topology is obtained.




(a) (b)

Figure 5.12: The von Mises stress distributions of (a) the initial design and (b) optimized structure with minimal energy release rate.

Because energy release rates are derived from the stress field, one may wonder how optimized structures compare with those obtained via stress optimization. Consider the following von Mises stress-based objective function $J_{\sigma_{\text{vM}}}$, which aims at obtaining a uniform stress throughout the entire domain:

$$J_{\sigma_{\text{vM}}} = \frac{1}{N_{\text{e}}} \sum_{i=1}^{N_{\text{e}}} (\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}})^q \,, \tag{5.29}$$

where $N_e$ is the number of elements in the design domain with solid material, and $q$ is an integer exponent, $\sigma_{\text{VM}i}$ is the von Mises stress of the $i$th element, and $\tilde{\sigma}_{\text{VM}}$ is the average von Mises stress of all elements in the solid part. Detailed sensitivity formulations for this objective function with respect to design variable $\boldsymbol{s}$ and displacement field $\tilde{\boldsymbol{U}}$ are given in Appendix A.5.3. For consistency, we set $q$ to have the same value as $p$. The same initial design is used as well, with the corresponding von Mises stress field shown in Figure 5.12(a). The final design obtained with the above objective function is given in Figure 5.13(a), where a round corner also emerges to remove the stress concentration. In addition, Figure 5.13(b) shows the energy release rate distribution of the final design with the maximum value 0.8352, which is smaller than the optimized result 0.8609 given in Figure 5.9(b). As a lower value of maximum energy release rate is obtained when optimizing the stress distribution, it is apparent that the final design obtained when optimizing the energy release rate is a local optimum. The corresponding von Mises stress distribution, which is relatively smooth near the rounded corner, is shown in the same figure.



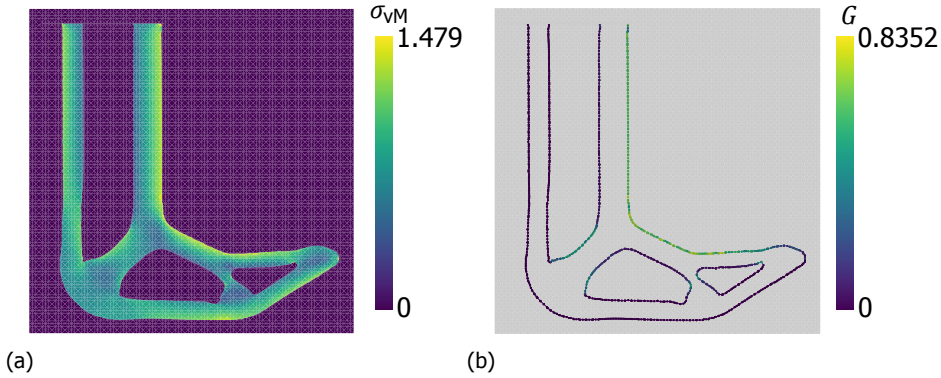



Figure 5.13: Final structure obtained by optimizing the difference between elemental and average von Mises stresses: (a) von Mises stress distribution in the optimized design; (b) energy release rates along the boundary.

The convergence plot and structural topologies obtained at 50, 100, 200 and 300 steps are shown in Figure 5.14, this time in semi-log scale because there are several orders of magnitude between the objective function values of initial and final designs. Since the value of $q$ is set to 8, this objective is a highly non-linear function of the stress field. Therefore, if the difference between elemental and averaged von Mises stresses $\sigma_{\text{VM}i} - \tilde{\sigma}_{\text{VM}}$ is greater than 1, the objective could be a large value. Conversely, if the difference is smaller than 1 the objective function approaches zero very rapidly. Moreover, the thin solid structures with stress concentration appear in the optimization process (see Figure 5.15(a)), which may result in peak objective values. In addition, the sudden change of the topology in the local region adjusts the loading path (see Figures 5.15(b) and (c)), which leads to the stress redistribution nearby. We then observe more oscillations during the optimization.

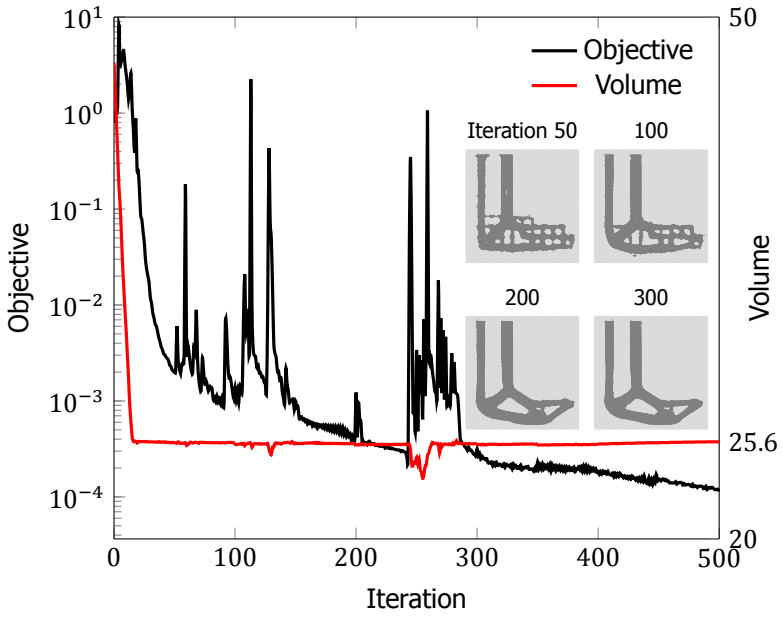



Figure 5.14: Convergence plot of objective and volume when optimizing the von Mises stress distribution and optimized designs obtained at 50, 100, 200, and 300 iterations.

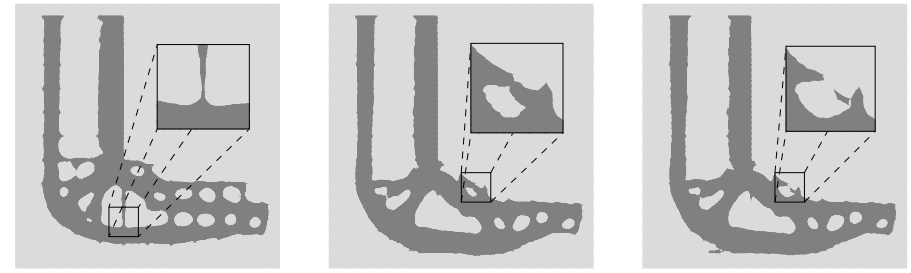



Figure 5.15: (a) Thin solid structure with high stress obtained at iteration 59; (b, c) The stress field is redistributed in the local area marked with a black square from (b) 127 to (c) 128 steps, which leads to a high stress distribution nearby.

### 5.4.3. Fracture anisotropy

The purpose of this example is to obtain an optimized design with fracture resistance anisotropy. In order to achieve this, a multiple-loading optimization problem is set up, where two unit magnitude compressive tractions $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are applied in are applied in two independent load cases, as shown in Figure 5.16. Therefore, two different finite element analyses are performed to obtain the structural solutions $\boldsymbol{u}_1^h(\boldsymbol{x})$ and $\boldsymbol{u}_2^h(\boldsymbol{x})$. Only a quarter of design domain is considered with dimensions

$1 \times 1$. In order to introduce fracture resistance anisotropy, energy release rates are minimized when compressing the computational domain along the vertical direction and maximized when compressed along the horizontal direction. This optimization, which is different from previous examples, is formally written as

$$\begin{aligned} &\text{minimize} && J = J_1 + J_2 = \frac{1}{N_{\text{node}}} \sum_{i=1}^{N_{\text{node}}} G_{1i} - \frac{1}{N_{\text{node}}} \sum_{i=1}^{N_{\text{node}}} G_{2i} \\ &\text{subject to} && \boldsymbol{K}_1 \boldsymbol{U}_1 = \boldsymbol{F}_1, \\ &&& \boldsymbol{K}_2 \boldsymbol{U}_2 = \boldsymbol{F}_2, \\ &&& V_{\text{s}} \leq V_{\text{c}}. \end{aligned} \tag{5.30}$$

In this equation, $J_i$, $i = \{1, 2\}$ is therefore an aggregation of energy release rate values for the corresponding boundary value problem $\boldsymbol{K}_i \boldsymbol{U}_i = \boldsymbol{F}_i$. Since these quantities are always positive, the negative sign is introduced to the second term $J_2$ to maximize energy release rates for the horizontal load case. Solid material is constrained at $V_{\text{c}} = 0.50$.

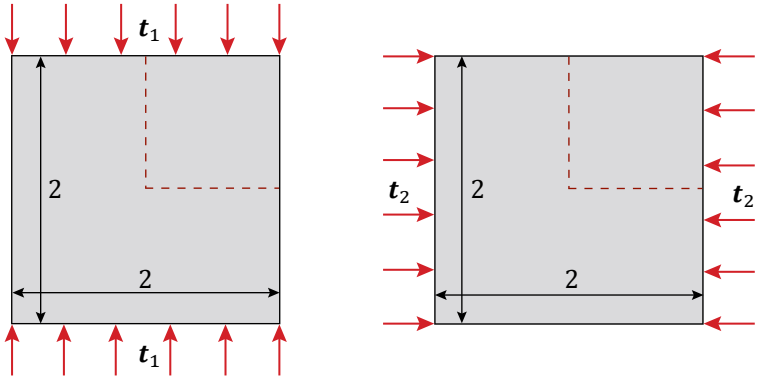



Figure 5.16: A domain with dimension $2 \times 2$ under compression, where $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are prescribed on the vertical and horizontal directions, respectively. Under the finite element analysis, a quarter of domain (marked with red dashed segments) is considered.

Figure 5.17(a) shows initial design including four holes in the whole domain (only one hole in the actual computational domain) where the ratio $J_2/J_1 = -1$. The corresponding final design in Figure 5.17(b) displays that more solid material is placed in along the vertical direction to make the structure tougher. Conversely, there are tiny bars that greatly contribute to maximizing energy release rates for the horizontal load case. For the optimized design, the ratio $J_2/J_1 = -14.85$. The convergence plot about $J_1$ and $J_2$ is given in Figure 5.18, where $J_1$ increases and $J_2$ decreases throughout the optimization. Since the magnitude of $J_2$ is larger than that of $J_1$, the optimizer gives priority to the former.

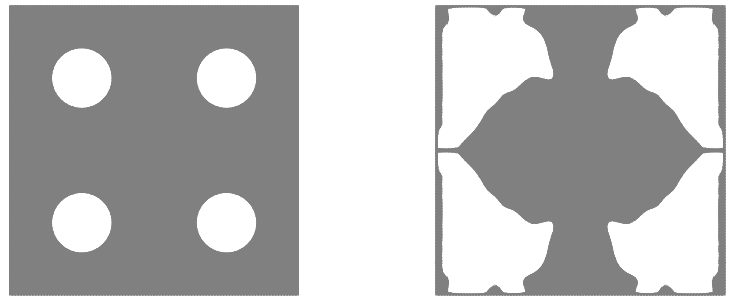



Figure 5.17: (a) Initial design with four holes in the whole domain where the ratio $J_2/J_1 = -1$; (b) The corresponding optimized design with $J_2/J_1 = -14.85$.



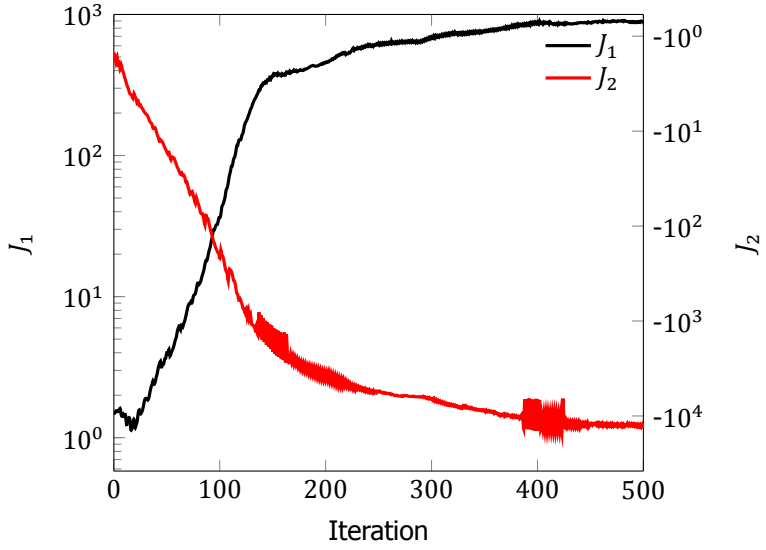



Figure 5.18: Convergence plot of $J_1$ and $J_2$ when optimizing the fracture resistance anisotropy.

## 5.5. Summary and conclusions

In this chapter we introduced a novel topology optimization procedure to design structures with tailored fracture resistance. The methodology uses a level set function discretized by radial basis functions to describe the topology, and an enriched finite element formulation to perform structural analysis. Contrary to other works, we do not predefine the location of cracks but instead assume these can nucleate perpendicularly to the structural boundary at discrete locations. Energy release rates of all potential cracks are then evaluated by conducting a single enriched finite element analysis of the intact uncracked model by means of topological derivatives. Since the stress field is used in the latter's formulation, a stress recovery

technique was proposed to enhance the stress field and thus overcome the issue of stress overestimation that burdens IGFEM—and other enriched FEMs alike. Since energy release rate is a local quantity, a $P$-mean aggregation function is used to build the objective function. Because in IGFEM it is straightforward to prescribe essential boundary conditions along discontinuities, the topology optimization can be performed by fully immersing the design domain into a background mesh.

By means of a shape optimization example, we showed that the proposed method recovers the optimal solution of a square plate with a circular hole, along which energy release rate values are uniform. Subsequently, an L-shaped bracket was embedded into a square computational domain discretized via a structured mesh. It was shown how the methodology removed in the final design the sharp inner corner with the stress concentration. Moreover, another objective function associated with von Mises stress was introduced to design a structure with a stress distribution as uniform as possible, which also removed the stress concentration by rounding the corner of the initial design. Finally, a multiple-loading optimization problem was used to design a structure with fracture resistance anisotropy. The technique was able to obtain a design that simultaneously maximized energy release rates when compressing the computational design horizontally while minimizing them when compressing vertically. As a result, the optimized structure is much tougher when compressing the structure vertically.



These are our final remarks:

- The methodology assume that cracks could only nucleate at right angles from the boundary. However, as discussed by Silva *et al.* [59], there are situations where this is not the most critical angle for crack nucleation. Our work could therefore be extended to find the critical angle for each potential crack, albeit at the expense of more computational resources and a more complex analytical sensitivity analysis formulation.

- We then used the $P$-mean function to aggregate all energy release rates calculated along the structural boundary. However, the value of $p$ affects the optimized result significantly. For instance, for $p = 2$ the sharp re-entrant corner in the L-bracket is not smoothened. Even though the energy release rate around that area has the highest value, the objective function approximates the maximum value inaccurately. While the aggregation function approaches to the maximum value as $p$ is increased, the function also becomes highly nonlinear and could thus lead to instabilities. As a result, there is always a trade-off between the accuracy of approximating the maximum value and the stability of the methodology.

- We also used an alternative objective function based on von Mises stress. The rationale behind obtaining a structure with a stress distribution as uniform as possible was to make the best use of material. Even though we did not perform the optimization for different values of the exponent $q$, optimized designs for this problem proved to be quite sensitive to this value, as found elsewhere with a similar stress-based formulation [95].

- The accuracy of energy release rates obtained by means of topological derivatives depends on the accuracy of the stress field. As a result, it is critical to choose an appropriate methodology to avoid stress oscillations during the optimization process that could degenerate the final design. The stress recovery technique adopted in this work cannot only prevent these issues, but also provide more accurate stresses than the directly-calculated stresses obtained by applying the gradient to the FE solution. However, the recovery technique also increases the complexity of the sensitivity formulation, which is the most intricate part of the this work. For instance, in order to obtain the sensitivity of recovered stresses with respect to design variables, the derivative of the equation used to evaluate the enhanced stress field should also be calculated, which makes the sensitivity formulation involved.

- Owing to properties of IGFEM, such as creating enriched nodes along discontinuities and using Lagrange shape functions of integration elements to construct enrichments, immersing design domains into any computational FE mesh gives tremendous flexibility. Even though we use the proposed method to obtain solid-void topologies, void material is still assigned to void areas to avoid ill-conditioned stiffness matrices. However, the proposed procedure can be extended to solve pure solid optimization problems by using techniques presented by van den Boom *et al.* [68].

- Compared to a stress-based optimization that considers stress values in all solid elements in the computational domain, our approach to optimize for fracture resistance only considers the evaluation of energy release rates along the boundaries. As a result, as the problems increase in size, asymptotically our approach is more efficient because it computes quantities in lower-dimensional manifolds. In addition, only enriched nodes under tension are considering when evaluating the objective function. To remain competitive, a stress-based optimization procedure would have to consider only the stress in solid elements crossed by interfaces. Nevertheless, such an approach would mandate for a procedure that detects such elements, complicating the computer implementation.

- Designing structures with enhanced fracture resistance in 3-D could also be done by developing the proposed topology optimization methodology. To that end, the most intricate part would be to obtain the formulation for the topological derivatives, for which some progress has already been made [96]. In addition, extra efforts are required to obtain the analytical sensitivity formulation for recovered stresses in 3-D.



# References

[1] Z. Wu, K. Ghosh, X. Qing, V. Karbhari, and F.-K. Chang, *Structural health monitoring of composite repair patches in bridge rehabilitation,* in *Smart Structures and Materials 2006: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 6174, edited by M. Tomizuka, C.-B. Yun,

and V. Giurgiutiu (International Society for Optics and Photonics, 2006) pp. 670 – 678.

[2] R. Brighenti, *Patch repair design optimisation for fracture and fatigue improvements of cracked plates,* International Journal of Solids and Structures **44**, 1115 (2007).

[3] A. Klarbring, B. Torstenfelt, U. Edlund, P. Schmidt, K. Simonsson, and H. Ansell, *Minimizing crack energy release rate by topology optimization,* Structural and Multidisciplinary Optimization **58**, 1695 (2018).

[4] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications* (Springer Science & Business Media, 2013).

[5] N. P. van Dijk, K. Maute, M. Langelaar, and F. Van Keulen, *Level-set methods for structural topology optimization: a review,* Structural and Multidisciplinary Optimization **48**, 437 (2013).

[6] O. Sigmund and K. Maute, *Topology optimization approaches,* Structural and Multidisciplinary Optimization **48**, 1031 (2013).

[7] W. Zhang, D. Li, P. Kang, X. Guo, and S.-K. Youn, *Explicit topology optimization using iga-based moving morphable void (mmv) approach,* Computer Methods in Applied Mechanics and Engineering **360**, 112685 (2020).

[8] C. B. W. Pedersen and P. Allinger, *Industrial implementation and applications of topology optimization and future needs,* in *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, edited by M. P. Bendsøe, N. Olhoff, and O. Sigmund (2006) pp. 229–238.

[9] J. Zhu, W. Zhang, and L. Xia, *Topology optimization in aircraft and aerospace structures design,* Archives of Computational Methods in Engineering **23**, 595 (2016).

[10] S. Amstutz and A. A. Novotny, *Topological optimization of structures subject to von mises stress constraints,* Structural and Multidisciplinary Optimization **41**, 407 (2010).

[11] W. Zhang, D. Li, J. Zhou, Z. Du, B. Li, and X. Guo, *A moving morphable void (mmv)-based explicit approach for topology optimization considering stress constraints,* Computer Methods in Applied Mechanics and Engineering **334**, 381 (2018).

[12] Z. Fan, L. Xia, W. Lai, Q. Xia, and T. Shi, *Evolutionary topology optimization of continuum structures with stress constraints,* Structural and Multidisciplinary Optimization **59**, 647 (2019).

[13] W. Zhang, S. Jiang, C. Liu, D. Li, P. Kang, S.-K. Youn, and X. Guo, *Stress-related topology optimization of shell structures using iga/tsa-based moving morphable void (mmv) approach,* Computer Methods in Applied Mechanics and Engineering **366**, 113036 (2020).

[14] M. Bruggi and P. Duysinx, *Topology optimization for minimum weight with compliance and stress constraints,* Structural and Multidisciplinary Optimization **46**, 369 (2012).

[15] Y. Luo and Z. Kang, *Topology optimization of continuum structures with drucker–prager yield stress constraints,* Computers & Structures **90**, 65 (2012).

[16] S. H. Jeong, S. H. Park, D.-H. Choi, and G. H. Yoon, *Topology optimization considering static failure theories for ductile and brittle materials,* Computers & Structures **110**, 116 (2012).

[17] R. Yang and C. Chen, *Stress-based topology optimization,* Structural optimization **12**, 98 (1996).

[18] P. Duysinx and M. P. Bendsøe, *Topology optimization of continuum structures with local stress constraints,* International journal for numerical methods in engineering **43**, 1453 (1998).



[19] G. Kreisselmeier and R. Steinhauser, *Systematic control design by optimizing a vector performance index,* IFAC Proceedings Volumes **12**, 113 (1979).

[20] E. Holmberg, B. Torstenfelt, and A. Klarbring, *Stress constrained topology optimization,* Structural and Multidisciplinary Optimization **48**, 33 (2013).

[21] P. Duysinx and O. Sigmund, *New developments in handling stress constraints in optimal material distribution,* in *7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization* (1998) p. 4906.

[22] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli, *Stress-based topology optimization for continua,* Structural and Multidisciplinary Optimization **41**, 605 (2010).

[23] J. París, F. Navarrina, I. Colominas, and M. Casteleiro, *Block aggregation of stress constraints in topology optimization of structures,* Advances in Engineering Software **41**, 433 (2010).

[24] W. Zhang, X. Guo, M. Y. Wang, and P. Wei, *Optimal topology design of continuum structures with stress concentration alleviation via level set method,* International Journal for Numerical Methods in Engineering **93**, 942 (2013).

[25] C. E. Inglis, *Stresses in a plate due to the presence of cracks and sharp corners,* Trans Inst Naval Archit **55**, 219 (1913).

[26] F. V. Senhora, O. Giraldo-Londono, I. F. Menezes, and G. H. Paulino, *Topology optimization with local stress constraints: a stress aggregation-free approach,* Structural and Multidisciplinary Optimization **62**, 1639 (2020).

[27] G. Allaire and F. Jouve, *Minimum stress optimal design with the level set method,* Engineering analysis with boundary elements **32**, 909 (2008).

[28] Q. Xia, T. Shi, S. Liu, and M. Y. Wang, *A level set solution to the stress-based structural shape and topology optimization,* Computers & Structures **90**, 55 (2012).

[29] R. Picelli, S. Townsend, C. Brampton, J. Norato, and H. Kim, *Stress-based shape and topology optimization with the level set method,* Computer Methods in Applied Mechanics and Engineering **329**, 1 (2018).

[30] M. P. Bendsøe and A. R. Díaz, *A method for treating damage related criteria in optimal topology design of continuum structures,* Structural optimization **16**, 108 (1998).

[31] O. Amir and O. Sigmund, *Reinforcement layout design for concrete structures based on continuum damage and truss topology optimization,* Structural and Multidisciplinary Optimization **47**, 157 (2013).

[32] O. Amir, *A topology optimization procedure for reinforced concrete structures,* Computers & Structures **114**, 46 (2013).

[33] K. A. James and H. Waisman, *Failure mitigation in optimal topology design using a coupled nonlinear continuum damage model,* Computer Methods in Applied Mechanics and Engineering **268**, 614 (2014).

[34] K. A. James and H. Waisman, *Topology optimization of structures under variable loading using a damage superposition approach,* International Journal for Numerical Methods in Engineering **101**, 375 (2015).

[35] L. Li, G. Zhang, and K. Khandelwal, *Topology optimization of energy absorbing structures with maximum damage constraint,* International Journal for Numerical Methods in Engineering **112**, 737 (2017).

[36] R. Alberdi and K. Khandelwal, *Topology optimization of pressure dependent elastoplastic energy absorbing structures with material damage constraints,* Finite Elements in Analysis and Design **133**, 42 (2017).

[37] L. Li, G. Zhang, and K. Khandelwal, *Failure resistant topology optimization of structures using nonlocal elastoplastic-damage model,* Structural and Multidisciplinary Optimization **58**, 1589 (2018).

[38] M. E. Launey and R. O. Ritchie, *On the fracture toughness of advanced materials,* Advanced Materials **21**, 2103 (2009).

[39] Z. Kang, P. Liu, and M. Li, *Topology optimization considering fracture mechanics behaviors at specified locations,* Structural and Multidisciplinary Optimization **55**, 1847 (2017).

[40] J. R. Rice, *A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks,* Journal of Applied Mechanics **35**, 379 (1968).

[41] J. Hu, S. Yao, N. Gan, Y. Xiong, and X. Chen, *Fracture strength topology optimization of structural specific position using a bi-directional evolutionary structural optimization method,* Engineering Optimization **52**, 583 (2020).

[42] G. X. Gu, L. Dimas, Z. Qin, and M. J. Buehler, *Optimization of composite fracture properties: method, validation, and applications,* Journal of Applied Mechanics **83**, 071006 (2016).

[43] M. Ambati, T. Gerasimov, and L. De Lorenzis, *A review on phase-field models of brittle fracture and a new fast hybrid formulation,* Computational Mechanics **55**, 383 (2015).

[44] T. Q. Bui and X. Hu, *A review of phase-field models, fundamentals and their applications to composite laminates,* Engineering Fracture Mechanics **248**, 107705 (2021).

[45] P. Diehl, R. Lipton, T. Wick, and M. Tyagi, *A comparative review of peridynamics and phase-field models for engineering fracture mechanics,* (2021).



[46] L. Xia, D. Da, and J. Yvonnet, *Topology optimization for maximizing the fracture resistance of quasi-brittle composites,* Computer Methods in Applied Mechanics and Engineering **332**, 234 (2018).

[47] J. B. Russ and H. Waisman, *Topology optimization for brittle fracture resistance,* Computer Methods in Applied Mechanics and Engineering **347**, 238 (2019).

[48] J. B. Russ and H. Waisman, *A novel topology optimization formulation for enhancing fracture resistance with a single quasi-brittle material,* International Journal for Numerical Methods in Engineering **121**, 2827 (2020).

[49] D. Da and J. Yvonnet, *Topology optimization for maximizing the fracture resistance of periodic quasi-brittle composites structures,* Materials **13**, 3279 (2020).

[50] C. Wu, J. Fang, S. Zhou, Z. Zhang, G. Sun, G. P. Steven, and Q. Li, *Level-set topology optimization for maximizing fracture resistance of brittle materials using phase-field fracture model,* International Journal for Numerical Methods in Engineering **121**, 2929 (2020).

[51] C. Wu, J. Fang, S. Zhou, Z. Zhang, G. Sun, G. P. Steven, and Q. Li, *A path-dependent level set topology optimization with fracture criterion,* Computers & Structures **249**, 106515 (2021).

[52] R. Jones, P. Chaperon, and M. Heller, *Structural optimisation with fracture strength constraints,* Engineering Fracture Mechanics **69**, 1403 (2002).

[53] T. Nishioka and S. Atluri, *Analytical solution for embedded elliptical cracks, and finite element alternating method for elliptical surface cracks, subjected to arbitrary loadings,* Engineering Fracture Mechanics **17**, 247 (1983).

[54] R. Das, R. Jones, and Y. Xie, *Design of structures for optimal static strength using eso,* Engineering Failure Analysis **12**, 61 (2005).

[55] D. Kujawski, *Estimations of stress intensity factors for small cracks at notches,* Fatigue & Fracture of Engineering Materials & Structures **14**, 953 (1991).

[56] V. J. Challis, A. P. Roberts, and A. H. Wilkins, *Fracture resistance via topology optimization,* Structural and Multidisciplinary Optimization **36**, 263 (2008).

[57] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *An interface-enriched generalized finite element method for level set-based topology optimization,* Structural and Multidisciplinary Optimization **63**, 1 (2021).

[58] S. Soghrati, A. M. Aragón, C. A. Duarte, and P. H. Geubelle, *An interface-enriched generalized fem for problems with discontinuous gradient fields,* International Journal for Numerical Methods in Engineering **89**, 991 (2012).

[59] M. Silva, P. H. Geubelle, and D. A. Tortorelli, *Energy release rate approximation for small surface-breaking cracks using the topological derivative,* Journal of the Mechanics and Physics of Solids **59**, 925 (2011).

[60] D. J. Payen and K.-J. Bathe, *A stress improvement procedure,* Computers & Structures **112-113**, 311 (2012).

[61] R. Sharma, J. Zhang, M. Langelaar, F. van Keulen, and A. M. Aragón, *An improved stress recovery technique for low-order 3d finite elements,* International Journal for Numerical Methods in Engineering **114**, 88 (2018).

[62] J. Zhang, F. van Keulen, and A. M. Aragón, *On tailoring fracture resistance: Level set topology optimization using an interface-enriched formulation,* Computer Methods in Applied Mechanics and Engineering (2021).

[63] R. Lins, S. P. Proença, and C. A. Duarte, *Efficient and accurate stress recovery procedure and a posteriori error estimator for the stable generalized/extended finite element method,* International Journal for Numerical Methods in Engineering **119**, 1279 (2019).

[64] K. Svanberg, *The method of moving asymptotes—a new method for structural optimization,* International Journal for Numerical Methods in Engineering **24**, 359 (1987).

[65] A. M. Aragón, B. Liang, H. Ahmadian, and S. Soghrati, *On the stability and interpolating properties of the hierarchical interface-enriched finite element method,* Computer Methods in Applied Mechanics and Engineering **362**, 112671 (2020).

[66] F. B. Barros, S. P. B. Proença, and C. S. de Barcellos, *On error estimator and p-adaptivity in the generalized finite element method,* International Journal for Numerical Methods in Engineering **60**, 2373 (2004).

[67] G. Zi and T. Belytschko, *New crack-tip elements for XFEM and applications to cohesive cracks,* International Journal for Numerical Methods in Engineering **57**, 2221 (2003).

[68] S. J. van den Boom, J. Zhang, F. van Keulen, and A. M. Aragón, *A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions,* International Journal for Numerical Methods in Engineering **120**, 1163 (2019).

[69] J. A. Sethian and A. Wiegmann, *Structural boundary design via level set and immersed interface methods,* Journal of computational physics **163**, 489 (2000).

[70] S. Wang and M. Y. Wang, *Radial basis functions and level set method for structural topology optimization,* International journal for numerical methods in engineering **65**, 2060 (2006).



[71] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,* Advances in computational Mathematics **4**, 389 (1995).

[72] J. Nocedal and S. J. Wright, *Sequential quadratic programming,* Numerical optimization , 529 (2006).

[73] E. Haber, *A multilevel, level-set method for optimizing eigenvalues in shape design problems,* Journal of Computational Physics **198**, 518 (2004).

[74] N. van Dijk, M. Langelaar, and F. van Keulen, *Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis,* International Journal for Numerical Methods in Engineering **91**, 67 (2012).

[75] P. Liu, Y. Luo, and Z. Kang, *Multi-material topology optimization considering interface behavior via xfem and level set method,* Computer Methods in Applied Mechanics and Engineering **308**, 113 (2016).

[76] M. Cui, C. Luo, G. Li, and M. Pan, *The parameterized level set method for structural topology optimization with shape sensitivity constraint factor,* Engineering with Computers **37**, 855 (2021).

[77] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, Vol. 3 (Cambridge university press, 1999).

[78] J. A. Sethian, *Evolution, implementation, and application of level set and fast marching methods for advancing fronts,* Journal of computational physics **169**, 503 (2001).

[79] M. Y. Wang, X. Wang, and D. Guo, *A level set method for structural topology optimization,* Computer Methods in Applied Mechanics and Engineering **192**, 227 (2003).

[80] G. Allaire, F. Jouve, and A.-M. Toader, *Structural optimization using sensitivity analysis and a level-set method,* Journal of Computational Physics **194**, 363 (2004).

[81] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, Vol. 153 (Springer Science & Business Media, 2006).

[82] L. Jiang and S. Chen, *Parametric structural shape & topology optimization with a variational distance-regularized level set method,* Computer Methods in Applied Mechanics and Engineering **321**, 316 (2017).

[83] M. Beghini, L. Bertini, and V. Fontanari, *Stress intensity factors for an inclined edge crack in a semiplane,* Engineering Fracture Mechanics **62**, 607 (1999).

[84] K. Alidoost, P. H. Geubelle, and D. A. Tortorelli, *Energy release rate approximation for edge cracks using higher-order topological derivatives,* International Journal of Fracture **210**, 187 (2018).

[85] P. Wei, M. Y. Wang, and X. Xing, *A study on x-fem in continuum structural optimization using a level set model,* Computer-Aided Design **42**, 708 (2010).

[86] N. Moës, A. Gravouil, and T. Belytschko, *Non-planar 3d crack growth by the extended finite element and level sets—part i: Mechanical model,* International Journal for Numerical Methods in Engineering **53**, 2549 (2002).

[87] M. Polajnar, F. Kosel, and R. Drazumeric, *Structural optimization using global stress-deviation objective function via the level-set method,* Structural and Multidisciplinary Optimization **55**, 91.

[88] S. H. Nguyen and H.-G. Kim, *Stress-constrained shape and topology optimization with the level set method using trimmed hexahedral meshes,* Computer Methods in Applied Mechanics and Engineering **366**, 113061 (2020).

[89] L. Van Miegroet, *Generalized shape optimization using XFEM and level set description*, Ph.D. thesis, Universit é de Li é ge, Li é ge, Belgium (2012).

[90] L. Noël and P. Duysinx, *Shape optimization of microstructural designs subject to local stress constraints within an xfem-level set framework,* Structural and Multidisciplinary Optimization **55**, 2323 (2017).

[91] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique,* International Journal for Numerical Methods in Engineering **33**, 1331 (1992).

[92] Y. Mei and X. Wang, *A level set method for structural topology optimization and its applications,* Advances in Engineering Software **35**, 415 (2004).

[93] G. Allaire, F. De Gournay, F. Jouve, and A.-M. Toader, *Structural optimization using topological and shape sensitivity via a level set method,* Control and cybernetics **34**, 59 (2005).

[94] X. Guo, W. Zhang, M. Y. Wang, and P. Wei, *Stress-related topology optimization via level set approach,* Computer Methods in Applied Mechanics and Engineering **200**, 3439 (2011).

[95] M. Y. Wang and L. Li, *Shape equilibrium constraint: a strategy for stress-constrained structural topology optimization,* Structural and Multidisciplinary Optimization **47**, 335 (2013).

[96] K. Alidoost, M. Feng, P. H. Geubelle, and D. A. Tortorelli, *Energy release rate approximation for small surface cracks in three-dimensional domains using the topological derivative,* Journal of Applied Mechanics **87** (2020).

# 6

# Conclusions and Recommendations

## 6.1. Conclusions

The work presented in this thesis first focused on accurate finite element analysis for solving fracture mechanics problems in 3-D. The Discontinuity-Enriched Finite Element Method (DE-FEM) was therefore extended to handle 3-D discontinuous models including both weak and strong discontinuities with a unified formulation. Since enriched formulations transfer the complexity from the matching mesh generation to the construction of the approximation space, a well-constructed object-oriented geometric engine was proposed to perform interactions between the background mesh and discontinuities, where new enriched discretizations are generated. However, integration elements with bad aspect ratios and/or relatively tiny areas could be created as discontinuities approach the background mesh nodes, which could lead to stress overestimation (inaccurate field gradients). Therefore, a stress improvement procedure (SIP) was proposed to enhance the approximation of discontinuous stress fields obtained from enriched finite element analysis. Finally, a computational methodology was proposed to carry out topology optimization for designing structures with tailored fracture resistance, where a formulation based on topological derivatives is used to evaluate energy release rates (ERRs). As this formulation highly depends on the stress fields, the proposed recovery technique was used to obtain an accurate stress approximation.

While extending DE-FEM to solve 3-D problems in Chapter 2, we showed that the DE-FEM formulation is able to generate two independent kinematic fields via a discontinuous patch test, where the importance of creating a conforming enriched discretization for integration on both sides of cracks was illustrated. Although no special enrichment functions are used to capture the singularity along the crack front, DE-FEM gave an accurate prediction of stress intensity factors (SIFs). Moreover, we demonstrated that DE-FEM is capable of dealing with the interplay between material interfaces and cracks within a complex-shaped femur model. More importantly, DE-FEM was demonstrated to be intrinsically stable with the use of a simple diagonal preconditioner.

Regarding the object-oriented geometric engine proposed in Chapter 3, we

found that it is critical to set appropriate tolerance values for the robustness when performing computational geometry operations, and that it is recommended to choose their values based on the background mesh size. Discontinuities can be handled in either sequential or hierarchical way. Noteworthy, the later could generate more integration elements, as children elements created by previous discontinuities could be split by the other discontinuities. In terms of efficiency and complexity, the sequential method is more appropriate for handling models with multiple discontinuities. Delaunay triangulation (tetrahedralization) behaved well for creating subdomains of cut elements in both 2-D and 3-D. Alternatively, it is also feasible to set specific routines to construct children elements for simple cases using straightforward rules, such as for splitting a triangle/tetrahedron completely.

In Chapter 4 it was shown that the proposed recovery technique can greatly improve the accuracy of the discontinuous stress field obtained with the Interface-enriched Generalized Finite Element Method (IGFEM), which even performs better than the directly calculated stresses from standard FEM with matching meshes. It is straightforward to extend the proposed technique to recover stress fields for fracture mechanics problems, where more attention should be paid to capture the stress singularity at the crack front. However, the recovery technique as a post-processing step cannot solve the fundamental issues, which arise from the approximation space constructed using enriched FEMs. Therefore, an alternative means to create the enriched finite element space is required for addressing this issue at the root.



Chapter 5 introduced a novel topology optimization methodology to design structures with enhanced or anisotropic fracture resistance. As IGFEM was used to perform the structural analysis, the proposed procedure can be extended to solve optimization problems, which do not require to assign weak materials to void parts for avoiding the ill-conditioning issue. IGFEM provides tremendous flexibility to optimize immersed design domains into any background mesh. As stress oscillations during the optimization process could make it difficult to find the optimal result, it is critical to choose an appropriate methodology to improve the stress field approximation. The proposed stress recovery technique was therefore adopted, which increases the complexity of the sensitivity formulation (the most intricate part of this work). Most importantly, since only enriched nodes under tension are considered into the objective function, which computes quantities in lower-dimensional manifolds, our approach is more efficient than stress-based optimization considering all solid elements as the problems increase in size asymptotically.

## 6.2. Recommendations

Here, we provide several possible directions and the corresponding recommendations to extend the work in this thesis.

- Since DE-FEM cannot get optimal convergence rates for problems with singularities, extra enrichment functions can be introduced to improve the stress field along crack fronts. Extra enriched degrees of freedom (DOFs) can be directly added to the original mesh data structure, and a specific quadrature technique could be required to integrate singular enrichments accurately.

- For simplicity, the work on topology optimization is based on the assumption that cracks can only nucleate perpendicularly to the structural boundary. However, this work could be extended to find the critical angle for each potential crack, which depends on the local stress field. Actually, this leads to a dual-optimization problem, where the most "dangerous" angle is firstly calculated and the corresponding energy release rate is then minimized. Although this approach requires an even more involved analytical sensitivity formulation and demands for more computational resources, the optimization procedure provides a more accurate result.
- The proposed optimization approach can be extended to design composite materials with enhanced fracture toughness, where its representative elementary volume (REV) is set to the design domain. Noteworthy, the effective fracture toughness is defined as the maximum energy release rate during the whole crack propagation process. Since the boundary of REV is not the same as that of the structural boundary, a different formulation based on topological derivatives would be required to evaluate energy release rates of all potential cracks, where these cracks are assumed to nucleate inside the design domain. Periodic boundary conditions should be imposed on the unit cell, and a constraint for keeping unit cells connected to each other should be satisfied.
- Design of 3-D structures with enhanced fracture resistance can be done by extending the proposed topology optimization methodology, where the key component is to obtain the formulation between SIFs and the stress field obtained from the finite element analysis of an intact domain in 3-D.
- As strength and fracture toughness are usually inverse relationships for most materials, both of them could be introduced into the same topology optimization framework as different objectives. For instance, strength-related terms can be set to the objective function, and energy release rates can be considered as constraints, or vice versa.
- Instead of improving the fracture resistance by optimizing the structural layout, an alternative is to use repairing patches bounded at the location with cracks to reduce the likelihood of fracture failure. Topology optimization could then be performed on the bounded patches for obtaining an optimized design. This procedure could use DE-FEM to predict the behavior of the structure with cracks, and IGFEM for obtaining the structural response of bounded patches. Noteworthy, it would also be necessary to predict the behavior of the glue interface that connects the cracked structure with the bounded patches.
- In regards to predicting the structural behavior, IGFEM could be used in other topology optimization techniques such as the moving morphable component-based (MMC) method. Since each MMC used to represent the structural topology is described explicitly with geometric parameters, such as length, thickness, and orientation, IGFEM can be naturally integrated into this optimization procedure.

# A

# Appendix

## A.1. Analytical fields for modes I, II, and III

For completeness, the equations of the displacement and stress fields for modes I, II, and III are provided. For displacement fields, they are expressed as

$$\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \frac{K_I}{\sqrt{2\pi r}} \begin{bmatrix} \frac{r}{4\mu}[(2\kappa - 1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}] \\ \frac{r}{4\mu}[(2\kappa + 1)\sin\frac{\theta}{2} - \sin\frac{3\theta}{2}] \\ -\frac{2\nu_1 z}{E}\cos\frac{\theta}{2} \end{bmatrix}, \tag{A.1}$$

$$\boldsymbol{u} = \frac{K_{II}}{\sqrt{2\pi r}} \begin{bmatrix} \frac{r}{4\mu}[(2\kappa + 3)\sin\frac{\theta}{2} + \sin\frac{3\theta}{2}] \\ \frac{r}{4\mu}[-(2\kappa - 3)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}] \\ \frac{2\nu_1 z}{E}\sin\frac{\theta}{2} \end{bmatrix}, \tag{A.2}$$

$$\boldsymbol{u} = \frac{K_{III}}{\mu}\sqrt{\frac{2r}{\pi}} \begin{bmatrix} 0 \\ 0 \\ \sin\frac{\theta}{2} \end{bmatrix}, \tag{A.3}$$

where $\mu$ is the shear modulus, $\kappa$ and $\nu_1$ are equal to $3 - 4\nu$ and 0 under the plane strain, or $(3 - \nu)/(1 + \nu)$ and $\nu$ under the plane stress, respectively, $r$ and $\theta$ are polar coordinates centered at the point along the crack front.

Stress fields around the crack front are given, respectively, by

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{K_I}{\sqrt{2\pi r}} \cos\frac{\theta}{2} \begin{bmatrix} 1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2} \\ 1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2} \\ 2\nu_2 \\ \sin\frac{\theta}{2}\cos\frac{3\theta}{2} \\ 0 \\ 0 \end{bmatrix}, \tag{A.4}$$

$$\boldsymbol{\sigma} = \frac{K_{II}}{\sqrt{2\pi r}} \begin{bmatrix} -\sin\frac{\theta}{2}(2 + \cos\frac{\theta}{2}\cos\frac{3\theta}{2}) \\ \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \\ -2\nu_2 \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2}(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}) \\ 0 \\ 0 \end{bmatrix}, \tag{A.5}$$

$$\boldsymbol{\sigma} = \frac{K_{III}}{\sqrt{2\pi r}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2} \end{bmatrix}, \tag{A.6}$$

where $\nu_2$ equals to $\nu$ (0) under the plane strain (plane stress).

## A.2. Expression of $\bar{\mathbf{E}}_\sigma, \mathbf{E}_\sigma$ and $\mathbf{E}_\zeta$

Interpolation matrices $\bar{\mathbf{E}}_\sigma, \mathbf{E}_\sigma$ and $\mathbf{E}_\zeta$ used in Equation (4.16) for recovering stress distributions in 2D and 3D are given by

$$\begin{aligned} \bar{\mathbf{E}}_\sigma &= \begin{bmatrix} 1 & 0 & 0 & x & y & 2xy & 0 & 0 & 0 & y^2 & 0 & x^2 \\ 0 & 1 & 0 & 0 & 0 & 0 & x & y & 2xy & 0 & x^2 & y^2 \\ 0 & 0 & 1 & -y & 0 & -y^2 & 0 & -x & -x^2 & 0 & 0 & -2xy \end{bmatrix}_{3\times 12}, \\ \mathbf{E}_\sigma &= \begin{bmatrix} 1 & x & y & xy & x^2 & y^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & y & xy & x^2 & y^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & y & xy & x^2 & y^2 \end{bmatrix}_{3\times 18}, \\ \mathbf{E}_\zeta &= \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}_{2\times 6}, \end{aligned} \tag{A.7}$$

and

$$\bar{\mathbf{E}}_\sigma^{\mathrm{T}} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
x & 0 & 0 & -y & 0 & 0 \\
y & 0 & 0 & 0 & 0 & 0 \\
z & 0 & 0 & 0 & 0 & 0 \\
2xy & 0 & 0 & -y^2 & 0 & 0 \\
yz & 0 & 0 & 0 & 0 & 0 \\
2xz & 0 & 0 & 0 & 0 & -z^2 \\
x^2 & 0 & 0 & -xy & yz & -xz \\
0 & 2x & 0 & 0 & 0 & 0 \\
0 & y & 0 & 0 & -z & 0 \\
0 & 2z & 0 & 0 & 0 & 0 \\
0 & 2xy & 0 & -x^2 & 0 & 0 \\
0 & 2yz & 0 & 0 & -z^2 & 0 \\
0 & xz & 0 & 0 & 0 & 0 \\
0 & y^2 & 0 & -xy & -yz & xz \\
0 & 0 & 2x & 0 & 0 & 0 \\
0 & 0 & 2y & 0 & 0 & 0 \\
0 & 0 & z & 0 & -y & 0 \\
0 & 0 & xy & 0 & 0 & 0 \\
0 & 0 & 2yz & 0 & -y^2 & 0 \\
0 & 0 & 2xz & 0 & 0 & -x^2 \\
0 & 0 & z^2 & xy & -yz & -xz \\
0 & y & 0 & -x & 0 & 0 \\
0 & 0 & 0 & -z & 0 & 0 \\
0 & 0 & 0 & 2yz & 0 & -z^2 \\
0 & 0 & 0 & 2xz & -z^2 & 0 \\
0 & x^2 & 0 & 0 & 0 & 0 \\
y^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -z^2 & 0 & 0 \\
0 & 0 & 0 & 0 & -x & 0 \\
0 & 0 & 0 & 0 & 2xy & -x^2 \\
0 & 0 & 0 & -x^2 & 2xz & 0 \\
0 & 0 & 0 & 0 & -x^2 & 0 \\
0 & 0 & y^2 & 0 & 0 & 0 \\
0 & z^2 & 0 & 0 & 0 & 0 \\
0 & 0 & z & 0 & 0 & -x \\
0 & 0 & 0 & 0 & 0 & -y \\
x & 0 & 0 & 0 & 0 & -z \\
0 & 0 & 0 & 0 & -y^2 & 2xy \\
0 & 0 & 0 & -y^2 & 0 & 2yz \\
0 & 0 & x^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -y^2 \\
z^2 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}_{48\times 6}, \tag{A.8}$$

$$\mathbf{E}_\sigma = \begin{bmatrix}
\boldsymbol{e}_\sigma & & & & \mathbf{0} & \\
& \boldsymbol{e}_\sigma & & \boldsymbol{e}_\sigma & & \\
& \mathbf{0} & & & \boldsymbol{e}_\sigma & \boldsymbol{e}_\sigma
\end{bmatrix}_{6\times 60},$$

$$\mathbf{E}_\zeta = \begin{bmatrix}
1 & x & y & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x & y & z & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & y & z
\end{bmatrix}_{3\times 12},$$

respectively, where $\boldsymbol{e}_\sigma = [\ 1 \quad x \quad y \quad zxy \quad yz \quad zx \quad x^2 \quad y^2 \quad z^2\ ]$.

A

## A.3. Matrix $\boldsymbol{H}(\gamma)$ of influence coefficients

$\boldsymbol{H}(\gamma)$ is a 2×2 matrix associated with angle $\gamma$ between cracks and the inward normal of the structural boundary, and its components are defined as

$$
\begin{aligned}
h_{11}(\gamma) &= \sum_{i=1}^{6}\left\{\left[1-\tan(\gamma)^2\right]\cdot c_i^{(\mathrm{I},1)}\cos[(i-1)\gamma]-\frac{\sin(\gamma)}{\cos(\gamma)^3}\cdot c_i^{(\mathrm{I},2)}\sin(i\gamma)\right\}\\
h_{12}(\gamma) &= \sum_{i=1}^{6}\left\{2\tan(\gamma)\cdot c_i^{(\mathrm{I},1)}\cos[(i-1)\gamma]+\frac{1}{\cos(\gamma)^2}\cdot c_i^{(\mathrm{I},2)}\sin(i\gamma)\right\}\\
h_{21}(\gamma) &= \sum_{i=1}^{6}\left\{\left[1-\tan(\gamma)^2\right]\cdot c_i^{(\mathrm{II},1)}\sin(i\gamma)-\frac{\tan(\gamma)}{\cos(\gamma)^2}\cdot c_i^{(\mathrm{II},2)}\cos[(i-1)\gamma]\right\}\\
h_{22}(\gamma) &= \sum_{i=1}^{6}\left\{2\tan(\gamma)\cdot c_i^{(\mathrm{II},1)}\sin(i\gamma)+\frac{1}{\cos(\gamma)^2}\cdot c_i^{(\mathrm{II},2)}\cos[(i-1)\gamma]\right\}
\end{aligned}
\quad , \tag{A.9}
$$

where $c_i^{(\mathrm{I},1)}$, $c_i^{(\mathrm{II},1)}$, $c_i^{(\mathrm{I},2)}$, and $c_i^{(\mathrm{II},2)}$ are given by the following table:

Table A.1: Data of parameters $c_i^{(\mathrm{I},1)}$, $c_i^{(\mathrm{II},1)}$, $c_i^{(\mathrm{I},2)}$, and $c_i^{(\mathrm{II},2)}$ in matrix $\boldsymbol{H}(\gamma)$.

| i | $c_i^{(\mathrm{I},1)}$ | $c_i^{(\mathrm{II},1)}$ | $c_i^{(\mathrm{I},2)}$ | $c_i^{(\mathrm{II},2)}$ |
|---|---|---|---|---|
| 1 | -0.174856 | -0.198196 | -0.419098 | 0.478653 |
| 2 | 1.393783 | 0.681479 | -0.197271 | -0.130868 |
| 3 | -0.278259 | -0.282608 | -0.445897 | 0.663435 |
| 4 | 0.240695 | 0.136522 | -0.050066 | -0.066599 |
| 5 | -0.071883 | -0.041562 | -0.022856 | 0.183693 |
| 6 | 0.011246 | 0.006177 | 0.003281 | -0.006140 |

As cracks are assumed to nucleate perpendicularly to the structural boundary, the angle $\gamma$ is set to 0, which leads to a simplified $\boldsymbol{H}(\gamma)$ matrix:

$$
\boldsymbol{H}(0)=\sum_{i=1}^{6}\begin{bmatrix} c_i^{(\mathrm{I},1)} & 0\\ 0 & c_i^{(\mathrm{II},2)}\end{bmatrix}. \tag{A.10}
$$

## A.4. Matrix $\boldsymbol{P}_g$ about angle $\beta$

The energy release rate was given in Equations (5.25) and (5.26) both in polar and Cartesian coordinates, respectively. In fact, expanding the matrix $\boldsymbol{Q}(\gamma,\beta)$, the expression for the energy release rate takes the form

$$\begin{aligned} G(\boldsymbol{x}_i,\eta,\gamma,\beta) &= \frac{\pi\eta}{\overline{E}}\boldsymbol{\sigma}(\boldsymbol{x}_i)^{\intercal}\boldsymbol{Q}(\gamma,\beta)\boldsymbol{\sigma}(\boldsymbol{x}_i) \\ &= \frac{\pi\eta}{\overline{E}}\boldsymbol{\sigma}(\boldsymbol{x}_i)^{\intercal}\boldsymbol{P}_g(\beta)^{\intercal}\boldsymbol{P}_l(\gamma)^{\intercal}\boldsymbol{H}^{\intercal}(\gamma)\boldsymbol{H}(\gamma)\boldsymbol{P}_l(\gamma)\boldsymbol{P}_g(\beta)\boldsymbol{\sigma}(\boldsymbol{x}_i), \end{aligned} \tag{A.11}$$

where we assume that cracks always nucleate perpendicularly to the boundary, *i.e.*, $\gamma = 0$ then for all potential cracks. $\boldsymbol{P}_g(\beta)$ is a $4\times 4$ matrix defined as

$$\begin{aligned} \boldsymbol{P}_g(\beta) &= \begin{bmatrix} \cos^2\beta & \sin\beta\cos\beta & \sin\beta\cos\beta & \sin^2\beta \\ -\sin\beta\cos\beta & \cos^2\beta & -\sin^2\beta & \sin\beta\cos\beta \\ -\sin\beta\cos\beta & -\sin^2\beta & \cos^2\beta & \sin\beta\cos\beta \\ \sin^2\beta & -\sin\beta\cos\beta & -\sin\beta\cos\beta & \cos^2\beta \end{bmatrix} \\ &= \begin{bmatrix} n_x^2 & n_x n_y & n_x n_y & n_y^2 \\ -n_x n_y & n_x^2 & -n_y^2 & n_x n_y \\ -n_x n_y & -n_y^2 & n_x^2 & n_x n_y \\ n_y^2 & -n_x n_y & -n_x n_y & n_x^2 \end{bmatrix}, \end{aligned} \tag{A.12}$$

where $\boldsymbol{n} = (n_x, n_y) = (-\cos\beta, -\sin\beta)$ is the outward normal of the local boundary.

Under the discretized model, $\boldsymbol{n}$ is expressed as the inward normal $\boldsymbol{n}_i$ of node $i$ that is evaluated by averaging the normal of segments along material interfaces. As shown in Figure A.1, node $i$ is shared by two line elements $ni$ and $im$ (marked

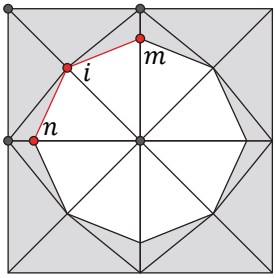



Figure A.1: Line elements $ni$ and $im$ (marked with red segments) sharing node $i$ are used to compute the inward normal $\boldsymbol{n}_i$ of node $i$.

with red segments). The average of normals of these two segments is evaluated

as

$$\boldsymbol{n}_{\text{avg}} = (n_{\text{avg}x}, n_{\text{avg}y}) = \frac{1}{2}(\boldsymbol{n}_{ni} + \boldsymbol{n}_{im}). \tag{A.13}$$

Then $\boldsymbol{n}_i$ is evaluated by normalizing $\boldsymbol{n}_{\text{avg}}$ as

$$\boldsymbol{n}_i = (n_x, n_y) = \frac{(n_{\text{avg}x}, n_{\text{avg}y})}{\sqrt{n_{\text{avg}x}^2 + n_{\text{avg}y}^2}}. \tag{A.14}$$

The other transformation matrix $\boldsymbol{P}_l(\gamma)$ is given by

$$\boldsymbol{P}_l(\gamma) = \begin{bmatrix} \sin^2(\gamma) & -\frac{\sin(2\gamma)}{2} & -\frac{\sin(2\gamma)}{2} & \cos^2(\gamma) \\ -\sin(\gamma)\cos(\gamma) & \frac{\cos^2(\gamma)-\sin^2(\gamma)}{2} & \frac{\cos^2(\gamma)-\sin^2(\gamma)}{2} & \sin(\gamma)\cos(\gamma) \end{bmatrix}. \tag{A.15}$$

Setting the angle $\gamma = 0$, $\boldsymbol{P}_l(\gamma)$ is simplified to

$$\boldsymbol{P}_l(0) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}. \tag{A.16}$$

## A.5. Sensitivity formulations

### A.5.1. Energy release rate

According to Equation (5.16), the derivative of $J$ associated with design variable $s_j$ is expressed as

$$\frac{\partial J}{\partial s_j} = \left(\frac{1}{N_{\text{node}}}\right)^{\frac{1}{p}} \frac{1}{p} \left(\sum_{i=1}^{N_{\text{node}}} G_i^p\right)^{\frac{1}{p}-1} \left(\sum_{i=1}^{N_{\text{node}}} p G_i^{p-1} \frac{\partial G_i}{\partial s_j}\right), \tag{A.17}$$

where

$$\begin{aligned} \frac{\partial G_i}{\partial s_j} &= \frac{\pi\eta}{\overline{E}_1} \frac{\partial \left(\boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{Q}(\gamma,\beta) \boldsymbol{\sigma}(\boldsymbol{x}_i)\right)}{\partial s_j} \\ &= \frac{\pi\eta}{\overline{E}_1} \left(\frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathsf{T}}}{\partial s_j} \boldsymbol{Q}(\gamma,\beta) \boldsymbol{\sigma}(\boldsymbol{x}_i) + \boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathsf{T}} \frac{\partial \boldsymbol{Q}(\gamma,\beta)}{\partial s_j} \boldsymbol{\sigma}(\boldsymbol{x}_i) + \boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{Q}(\gamma,\beta) \frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)}{\partial s_j}\right). \end{aligned} \tag{A.18}$$

According to Equation (4.17), the derivative of nodal stress $\boldsymbol{\sigma}(\boldsymbol{x}_i)$ with respect to design variable $s_j$ is defined as

$$\begin{aligned} \frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)}{\partial s_j} &= \frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_n} \frac{\partial \boldsymbol{x}_n}{\partial s_j} = \frac{1}{N_i} \frac{\partial \left(\sum \boldsymbol{E}_\sigma(\boldsymbol{x}_i) \hat{\boldsymbol{\sigma}}_e\right)}{\partial \boldsymbol{x}_n} \frac{\partial \boldsymbol{x}_n}{\partial s_j} \\ &= \frac{1}{N_i} \sum \left(\frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_n} \hat{\boldsymbol{\sigma}}_e + \boldsymbol{E}_\sigma(\boldsymbol{x}_i) \frac{\partial \hat{\boldsymbol{\sigma}}_e}{\partial \boldsymbol{x}_n}\right) \frac{\partial \boldsymbol{x}_n}{\partial s_j}. \end{aligned} \tag{A.19}$$

The first term $(\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_i)/\partial \boldsymbol{x}_n)\,\hat{\boldsymbol{\sigma}}_e$ is only non-zero, when $\boldsymbol{x}_i$ equals to enriched node $\boldsymbol{x}_n$, and derivatives of $\boldsymbol{E}_\sigma(\boldsymbol{x}_i)$ with respect to coordinates $(x_i, y_i)$ are given by

$$\begin{aligned}
\frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_i)}{\partial x_i} &= \begin{bmatrix} 0 & 1 & 0 & y_i & 2x_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & y_i & 2x_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & y_i & 2x_i & 0 \end{bmatrix}, \\
\frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_i)}{\partial y_i} &= \begin{bmatrix} 0 & 0 & 1 & x_i & 0 & 2y_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_i & 0 & 2y_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_i & 0 & 2y_i \end{bmatrix}.
\end{aligned} \tag{A.20}$$

In order to evaluate $\partial \hat{\boldsymbol{\sigma}}_e/\partial \boldsymbol{x}_n$, we need to solve the derivative of Equation (4.16), which is used to obtain the recovered stress field. For simplicity, we redefine several terms in Equation (4.16) as

$$\boldsymbol{A}_e = \begin{bmatrix} \int_e \bar{\boldsymbol{E}}_\sigma^{\mathrm{T}} \boldsymbol{E}_\sigma \mathrm{d}e \\ \int_e \boldsymbol{E}_\zeta^{\mathrm{T}} \partial^{\mathrm{T}} \boldsymbol{E}_\sigma \mathrm{d}e \end{bmatrix}, \boldsymbol{B}_{e1} = \int_e \bar{\boldsymbol{E}}_\sigma^{\mathrm{T}} \boldsymbol{\sigma}_e^h \mathrm{d}e, \boldsymbol{B}_{e2} = \int_e \boldsymbol{E}_\zeta^{\mathrm{T}} \boldsymbol{b} \mathrm{d}e. \tag{A.21}$$

Then, the corresponding derivative associated with enriched node $\boldsymbol{x}_n$ is given by

$$\left(\sum \frac{\partial \boldsymbol{A}_e}{\partial \boldsymbol{x}_n}\right)\hat{\boldsymbol{\sigma}}_e + \left(\sum \boldsymbol{A}_e\right)\frac{\partial \hat{\boldsymbol{\sigma}}_e}{\partial \boldsymbol{x}_n} = \sum \begin{bmatrix} \frac{\partial \boldsymbol{B}_{e1}}{\partial \boldsymbol{x}_n} \\ \frac{\partial \boldsymbol{B}_{e2}}{\partial \boldsymbol{x}_n} \end{bmatrix}. \tag{A.22}$$

After solving the equation above, the derivative of $\hat{\boldsymbol{\sigma}}_e$ corresponding to enriched node $\boldsymbol{x}_n$is expressed as

$$\frac{\partial \hat{\boldsymbol{\sigma}}_e}{\partial \boldsymbol{x}_n} = \left(\sum \boldsymbol{A}_e\right)^{-1} \left( \sum \begin{bmatrix} \frac{\partial \boldsymbol{B}_{e1}}{\partial \boldsymbol{x}_n} \\ \frac{\partial \boldsymbol{B}_{e2}}{\partial \boldsymbol{x}_n} \end{bmatrix} - \left(\sum \frac{\partial \boldsymbol{A}_e}{\partial \boldsymbol{x}_n}\right)\hat{\boldsymbol{\sigma}}_e \right). \tag{A.23}$$

According to Equation (A.21), $\partial \boldsymbol{A}_e/\partial \boldsymbol{x}_n$ and $\partial \boldsymbol{B}_{ei}/\partial \boldsymbol{x}_n, i = 1,2$ can be obtained directly. As enriched nodes are created at intersections between element edges and material interfaces, their locations are evaluated based on level set values and coordinates of background mesh nodes. As shown in Figure A.2, enriched node $\boldsymbol{x}_n$

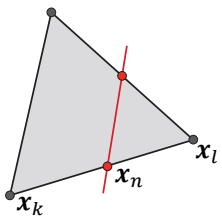



Figure A.2: Enriched node $\boldsymbol{x}_n$ is created at the intersection between a material interface (marked with a red segment) and an element edge with $(\boldsymbol{x}_k, \boldsymbol{x}_l)$.

is related to original mesh nodes $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$, and it is expressed as

$$\boldsymbol{x}_n = \boldsymbol{x}_l - \frac{\phi_l}{\phi_k - \phi_l}(\boldsymbol{x}_k - \boldsymbol{x}_l), \tag{A.24}$$

where $\phi_k$ and $\phi_l$ are level set values of $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$, respectively. The derivative of $\boldsymbol{x}_n$ with respect to design variable $s_j$ is then given by

$$\frac{\partial \boldsymbol{x}_n}{\partial s_j} = \frac{\partial \boldsymbol{x}_n}{\partial \phi_l}\frac{\partial \phi_l}{\partial s_j} + \frac{\partial \boldsymbol{x}_n}{\partial \phi_k}\frac{\partial \phi_k}{\partial s_j}, \tag{A.25}$$

where

$$\frac{\partial \boldsymbol{x}_n}{\partial \phi_l} = -\frac{\phi_k}{(\phi_k - \phi_l)^2}(\boldsymbol{x}_k - \boldsymbol{x}_l), \tag{A.26}$$

and

$$\frac{\partial \boldsymbol{x}_n}{\partial \phi_k} = -\frac{\phi_l}{(\phi_k - \phi_l)^2}(\boldsymbol{x}_k - \boldsymbol{x}_l). \tag{A.27}$$

Since level set function $\phi$ is constructed as compactly supported RBFs (shown in Equation (5.18)), $\partial\phi_l/\partial s_j$ and $\partial\phi_k/\partial s_j$ equal to $\zeta_j(\boldsymbol{x}_l)$ and $\zeta_j(\boldsymbol{x}_k)$, respectively. The derivative of $\boldsymbol{Q}(\gamma,\beta)$ with respect to design variable $s_j$ in Equation (A.18) is given in the following Chapter A.5.4. For the derivative of $J$ associated with the displacement $\tilde{\boldsymbol{U}}$, it is shown as

$$\frac{\partial J}{\partial \tilde{\boldsymbol{U}}} = \left(\frac{1}{N_{\text{node}}}\right)^{\frac{1}{p}} \frac{1}{p}\left(\sum_{i=1}^{N_{\text{node}}} G_i^p\right)^{\frac{1}{p}-1}\left(\sum_{i=1}^{N_{\text{node}}} p G_i^{p-1}\frac{\partial G_i}{\partial \tilde{\boldsymbol{U}}}\right), \tag{A.28}$$

where

$$\begin{aligned}\frac{\partial G_i}{\partial \tilde{\boldsymbol{U}}} &= \frac{\pi\varepsilon}{\overline{\overline{E}}_1}\frac{\partial\left(\boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{Q}(\gamma,\beta)\boldsymbol{\sigma}(\boldsymbol{x}_i)\right)}{\partial \tilde{\boldsymbol{U}}} \\ &= \frac{\pi\varepsilon}{\overline{\overline{E}}_1}\left(\frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathrm{T}}}{\partial \tilde{\boldsymbol{U}}}\boldsymbol{Q}(\gamma,\beta)\boldsymbol{\sigma}(\boldsymbol{x}_i) + \boldsymbol{\sigma}(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{Q}(\gamma,\beta)\frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)}{\partial \tilde{\boldsymbol{U}}}\right).\end{aligned} \tag{A.29}$$

Similar to Equation (A.19), the derivative of $\boldsymbol{\sigma}(\boldsymbol{x}_i)$ associated with displacement $\tilde{\boldsymbol{U}}$ is given by

$$\frac{\partial \boldsymbol{\sigma}(\boldsymbol{x}_i)}{\partial \tilde{\boldsymbol{U}}} = \frac{\partial\left(\sum \boldsymbol{E}_\sigma(\boldsymbol{x}_i)\hat{\boldsymbol{\sigma}}_e\right)}{\partial \tilde{\boldsymbol{U}}} = \boldsymbol{E}_\sigma(\boldsymbol{x}_i)\sum \frac{\partial \hat{\boldsymbol{\sigma}}_e}{\partial \tilde{\boldsymbol{U}}}. \tag{A.30}$$

In order to evaluate $\partial\hat{\boldsymbol{\sigma}}_e/\partial\tilde{\boldsymbol{U}}$, we need to solve the derivative of Equation (4.16) associated with $\tilde{\boldsymbol{U}}$. Then, the corresponding derivative is expressed as

$$\frac{\partial \hat{\boldsymbol{\sigma}}_e}{\partial \tilde{\boldsymbol{U}}} = \left(\sum \boldsymbol{A}_e\right)^{-1}\left(\sum\begin{bmatrix}\frac{\partial \boldsymbol{B}_{e1}}{\partial \tilde{\boldsymbol{U}}} \\ \frac{\partial \boldsymbol{B}_{e2}}{\partial \tilde{\boldsymbol{U}}}\end{bmatrix} - \left(\sum \frac{\partial \boldsymbol{A}_e}{\partial \tilde{\boldsymbol{U}}}\right)\hat{\boldsymbol{\sigma}}_e\right). \tag{A.31}$$

As shown in Equation (A.21), $\boldsymbol{A}_e$ is only related to coordinates, $\partial\boldsymbol{A}_e/\partial\tilde{\boldsymbol{U}}$ the equals to 0.

### A.5.2. Stiffness matrix and force vector

According to Equations (5.13) and (5.14), derivatives of $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{F}}$ associated to design variable $s_j$ can be expressed as

$$\frac{\partial \tilde{\boldsymbol{K}}}{\partial s_j} = \frac{\partial \boldsymbol{T}^{\intercal}}{\partial s_j}\boldsymbol{K}\boldsymbol{T}\tilde{\boldsymbol{U}} + \boldsymbol{T}^{\intercal}\frac{\partial \boldsymbol{K}}{\partial s_j}\boldsymbol{T}\tilde{\boldsymbol{U}} + \boldsymbol{T}^{\intercal}\boldsymbol{K}\frac{\partial \boldsymbol{T}}{\partial s_j}\tilde{\boldsymbol{U}}, \tag{A.32}$$

and

$$\frac{\partial \tilde{\boldsymbol{F}}}{\partial s_j} = \frac{\partial \boldsymbol{T}^{\intercal}}{\partial s_j}(\boldsymbol{F} - \boldsymbol{K}\boldsymbol{g}) + \boldsymbol{T}^{\intercal}\left(\frac{\partial \boldsymbol{F}}{\partial s_j} - \frac{\partial \boldsymbol{K}}{\partial s_j}\boldsymbol{g} - \boldsymbol{K}\frac{\partial \boldsymbol{g}}{\partial s_j}\right). \tag{A.33}$$

According to Equation (5.7), $\partial \boldsymbol{K}/\partial s_j$ and $\partial \boldsymbol{F}/\partial s_j$ can be evaluated directly. $\partial \boldsymbol{g}/\partial s_j$ is a zero vector as prescribed displacements are not related to design variables. As shown in Equation (5.12), $\boldsymbol{T}$ matrix is constructed with shape function $N_i, i = 1, 2, 3$, it can be expressed as a $10 \times 10$ matrix:

$$\boldsymbol{T} = \left[\begin{array}{cccccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hdashline -N_1 & 0 & -N_2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -N_1 & 0 & -N_2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -N_2 & 0 & -N_3 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -N_2 & 0 & -N_3 & 0 & 0 & 0 & 1 \end{array}\right]. \tag{A.34}$$

Its sensitivity with respect to design variable $s_j$ can be computed as

$$\frac{\partial \boldsymbol{T}}{\partial s_j} = \left[\begin{array}{cccccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hdashline -\frac{\partial N_1}{\partial s_j} & 0 & -\frac{\partial N_2}{\partial s_j} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\partial N_1}{\partial s_j} & 0 & -\frac{\partial N_2}{\partial s_j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\partial N_2}{\partial s_j} & 0 & -\frac{\partial N_3}{\partial s_j} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\partial N_2}{\partial s_j} & 0 & -\frac{\partial N_3}{\partial s_j} & 0 & 0 & 0 & 0 \end{array}\right], \tag{A.35}$$

where the derivative of shape function $N_i, i = 1, 2, 3$ associated with design variable $s_j$ is given by

$$\frac{\partial N_i}{\partial s_j} = \frac{\partial N_i}{\partial \boldsymbol{x}_n}\frac{\partial \boldsymbol{x}_n}{\partial s_j}. \tag{A.36}$$

### A.5.3. von Mises stress

Considering the objective function $J_{\sigma_{\text{vM}}}$ associated with von Mises stress, it is given by

$$J_{\sigma_{\text{vM}}} = \frac{1}{N_{\text{e}}} \sum_{i=1}^{N_{\text{e}}} \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)^q . \tag{A.37}$$

Its derivative associated to design variable $s_j$ is then shown as

$$\begin{aligned} \frac{\partial J_{\sigma_{\text{vM}}}}{\partial s_j} &= \frac{q}{N_{\text{e}}} \sum_{i=1}^{N_{\text{e}}} \left( \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)^{q-1} \frac{\partial \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)}{\partial s_j} \right) \\ &= \frac{q}{N_{\text{e}}} \sum_{i=1}^{N_{\text{e}}} \left( \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)^{q-1} \frac{\partial \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)}{\partial \boldsymbol{x}_n} \frac{\partial \boldsymbol{x}_n}{\partial s_j} \right). \end{aligned} \tag{A.38}$$

As von Mises stress $\sigma_{\text{vM}}$ defined in 2-D is expressed as

$$\sigma_{\text{vM}} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 - \sigma_{xx}\sigma_{yy} + 3\sigma_{xy}^2} = \sqrt{\boldsymbol{\sigma}^{\mathsf{T}} \boldsymbol{R} \boldsymbol{\sigma}}, \tag{A.39}$$

where

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0 \\ -0.5 & 0 & 0 & 1 \end{bmatrix}, \tag{A.40}$$

its derivative corresponding to design variable $s_j$ is defined as

$$\frac{\partial \sigma_{\text{vM}}}{\partial s_j} = \frac{\partial \sqrt{\boldsymbol{\sigma}^{\mathsf{T}} \boldsymbol{R} \boldsymbol{\sigma}}}{\partial s_j} = \frac{1}{2\sigma_{\text{vM}}} \left( \frac{\partial \boldsymbol{\sigma}^{\mathsf{T}}}{\partial s_j} \boldsymbol{R} \boldsymbol{\sigma} + \boldsymbol{\sigma}^{\mathsf{T}} \boldsymbol{R} \frac{\partial \boldsymbol{\sigma}}{\partial s_j} \right). \tag{A.41}$$

Then the sensitivity term $\left(\partial \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)\right) / \partial s_j$ in Equation (A.38) can be expressed as

$$\frac{\partial \left(\sigma_{\text{vM}i} - \bar{\sigma}_{\text{vM}}\right)}{\partial s_j} = \frac{1}{2\sigma_{\text{vM}i}} \left( \frac{\partial \boldsymbol{\sigma}_i^{\mathsf{T}}}{\partial s_j} \boldsymbol{R} \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_i^{\mathsf{T}} \boldsymbol{R} \frac{\partial \boldsymbol{\sigma}_i}{\partial s_j} \right) - \frac{1}{N_{\text{e}}} \sum_{l=1}^{N_{\text{e}}} \frac{\partial \sigma_{\text{vM}l}}{\partial s_j}. \tag{A.42}$$

For the derivative of recovered elemental stress $\boldsymbol{\sigma}_i$ corresponding to design variable $s_j$, it is given by

$$\frac{\partial \boldsymbol{\sigma}_i}{\partial s_j} = \frac{\partial \left( \boldsymbol{E}_\sigma(\boldsymbol{x}_g) \hat{\boldsymbol{\sigma}}_i \right)}{\partial s_j} = \left( \frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_g)}{\partial \boldsymbol{x}_n} \hat{\boldsymbol{\sigma}}_i + \boldsymbol{E}_\sigma(\boldsymbol{x}_g) \frac{\partial \hat{\boldsymbol{\sigma}}_i}{\partial \boldsymbol{x}_n} \right) \frac{\partial \boldsymbol{x}_n}{\partial s_j}, \tag{A.43}$$

where $\boldsymbol{x}_g$ is the global coordinate of gauss points in the $i$th element, $\partial\hat{\boldsymbol{\sigma}}_i/\partial\boldsymbol{x}_n$ can be obtained by solving Equation (A.23), and $\hat{\boldsymbol{\sigma}}_i$ is a vector derived from Equation (4.16). The derivative of interpolation matrix $\boldsymbol{E}_\sigma(\boldsymbol{x}_g)$ associated to $\boldsymbol{x}_n$ is defined as

$$\frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_g)}{\partial \boldsymbol{x}_n} = \frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_g)}{\partial \boldsymbol{x}_g}\frac{\partial \boldsymbol{x}_g}{\partial \boldsymbol{x}_n} = \frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_g)}{\partial \boldsymbol{x}_g}\frac{\partial \boldsymbol{x}_e^{\mathrm{T}}}{\partial \boldsymbol{x}_n}\boldsymbol{N}(\boldsymbol{x}_i), \tag{A.44}$$

where $\boldsymbol{x}_e$ is nodal coordinates of element $i$, $\boldsymbol{x}_i$ is the local coordinate of gauss point, and $\boldsymbol{N}$ is a vector of shape functions.

For the derivative of $J_{\sigma_{\mathrm{vM}}}$ associated with the displacement $\tilde{\boldsymbol{U}}$, it is shown as

$$\frac{\partial J_{\sigma_{\mathrm{vM}}}}{\partial \tilde{\boldsymbol{U}}} = \frac{q}{N_{\mathrm{e}}}\sum_{i=1}^{N_{\mathrm{e}}}\left(\left(\sigma_{\mathrm{vM}i} - \bar{\sigma}_{\mathrm{vM}}\right)^{q-1}\frac{\partial\left(\sigma_{\mathrm{vM}i} - \bar{\sigma}_{\mathrm{vM}}\right)}{\partial \tilde{\boldsymbol{U}}}\right), \tag{A.45}$$

where

$$\frac{\partial\left(\sigma_{\mathrm{vM}i} - \bar{\sigma}_{\mathrm{vM}}\right)}{\partial \tilde{\boldsymbol{U}}} = \frac{1}{2\sigma_{\mathrm{vM}i}}\left(\frac{\partial \boldsymbol{\sigma}_i^{\mathrm{T}}}{\partial \tilde{\boldsymbol{U}}}\boldsymbol{R}\boldsymbol{\sigma}_i + \boldsymbol{\sigma}_i^{\mathrm{T}}\boldsymbol{R}\frac{\partial \boldsymbol{\sigma}_i}{\partial \tilde{\boldsymbol{U}}}\right) - \frac{1}{N_{\mathrm{e}}}\sum_{l=1}^{N_{\mathrm{e}}}\frac{\partial \sigma_{\mathrm{vM}l}}{\partial \tilde{\boldsymbol{U}}}. \tag{A.46}$$

The derivative of recovered elemental stress $\boldsymbol{\sigma}_i$ associated with the displacement $\tilde{\boldsymbol{U}}$ is defined as

$$\frac{\partial \boldsymbol{\sigma}_i}{\partial \tilde{\boldsymbol{U}}} = \frac{\partial\left(\boldsymbol{E}_\sigma(\boldsymbol{x}_g)\hat{\boldsymbol{\sigma}}_i\right)}{\partial \tilde{\boldsymbol{U}}} = \frac{\partial \boldsymbol{E}_\sigma(\boldsymbol{x}_g)}{\partial \tilde{\boldsymbol{U}}}\hat{\boldsymbol{\sigma}}_i + \boldsymbol{E}_\sigma(\boldsymbol{x}_g)\frac{\partial \hat{\boldsymbol{\sigma}}_i}{\partial \tilde{\boldsymbol{U}}} = \boldsymbol{E}_\sigma(\boldsymbol{x}_g)\frac{\partial \hat{\boldsymbol{\sigma}}_i}{\partial \tilde{\boldsymbol{U}}}, \tag{A.47}$$

where $\partial\hat{\boldsymbol{\sigma}}_i/\partial\tilde{\boldsymbol{U}}$ can be obtained by solving Equation (A.31).

### A.5.4. Matrix $\boldsymbol{Q}$ about angles $\gamma$ and $\beta$

According to Equation (5.26), $\boldsymbol{Q}(\gamma,\beta)$ is a matrix used to transform stress $\boldsymbol{\sigma}$ defined in the global coordinate system to stress $\boldsymbol{\sigma}_\theta$ under the polar coordinate system located along the boundary, and it is expressed as

$$\boldsymbol{Q}(\gamma,\beta) = \boldsymbol{P}_g(\beta)^{\mathrm{T}}\boldsymbol{P}_l(\gamma)^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\gamma)\boldsymbol{H}(\gamma)\boldsymbol{P}_l(\gamma)\boldsymbol{P}_g(\beta). \tag{A.48}$$

Since cracks are assumed to nucleate perpendicular to the structural boundary, the angle $\gamma$ is set as $0$. It means that matrices $\boldsymbol{P}_l(\gamma)$ and $\boldsymbol{H}(\gamma)$ are not affected by design variables. Derivative of matrix $\boldsymbol{Q}(\gamma,\beta)$ with respect to design variable $s_j$ is then expressed as

$$\begin{aligned}\frac{\partial \boldsymbol{Q}(\gamma,\beta)}{\partial s_j} = {} & \frac{\partial \boldsymbol{P}_g(\beta)^{\mathrm{T}}}{\partial s_j}\boldsymbol{P}_l(\gamma)^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\gamma)\boldsymbol{H}(\gamma)\boldsymbol{P}_l(\gamma)\boldsymbol{P}_g(\beta) \\ & + \boldsymbol{P}_g(\beta)^{\mathrm{T}}\boldsymbol{P}_l(\gamma)^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\gamma)\boldsymbol{H}(\gamma)\boldsymbol{P}_l(\gamma)\frac{\partial \boldsymbol{P}_g(\beta)}{\partial s_j}.\end{aligned} \tag{A.49}$$

According to Equation (A.12), the derivative of $\boldsymbol{P}_g(\beta)$ with respect to design variable $s_j$ is expressed as

$$\frac{\partial \boldsymbol{P}_g(\beta)}{\partial s_j} = \begin{bmatrix} 2n_x\frac{\partial n_x}{\partial s_j} & \frac{\partial n_x}{\partial s_j}n_y + n_x\frac{\partial n_y}{\partial s_j} & \frac{\partial n_x}{\partial s_j}n_y + n_x\frac{\partial n_y}{\partial s_j} & 2n_y\frac{\partial n_y}{\partial s_j} \\ -\frac{\partial n_x}{\partial s_j}n_y - n_x\frac{\partial n_y}{\partial s_j} & 2n_x\frac{\partial n_x}{\partial s_j} & -2n_y\frac{\partial n_y}{\partial s_j} & \frac{\partial n_x}{\partial s_j}n_y + n_x\frac{\partial n_y}{\partial s_j} \\ -\frac{\partial n_x}{\partial s_j}n_y - n_x\frac{\partial n_y}{\partial s_j} & -2n_y\frac{\partial n_y}{\partial s_j} & 2n_x\frac{\partial n_x}{\partial s_j} & \frac{\partial n_x}{\partial s_j}n_y + n_x\frac{\partial n_y}{\partial \boldsymbol{s}_j} \\ 2n_y\frac{\partial n_y}{\partial s_j} & -\frac{\partial n_x}{\partial s_j}n_y - n_x\frac{\partial n_y}{\partial s_j} & -\frac{\partial n_x}{\partial s_j}n_y - n_x\frac{\partial n_y}{\partial s_j} & 2n_x\frac{\partial n_x}{\partial s_j} \end{bmatrix}. \tag{A.50}$$

As shown in Equation (A.14), sensitivities of $n_x$ and $n_y$ associated with design variable $s_j$ are calculated as

$$\frac{\partial n_x}{\partial s_j} = \frac{\partial \frac{n_{\text{avg}x}}{\sqrt{n_{\text{avg}x}^2+n_{\text{avg}y}^2}}}{\partial s_j} = \frac{n_{\text{avg}y}^2\frac{\partial n_{\text{avg}x}}{\partial s_j} - n_{\text{avg}x}n_{\text{avg}y}\frac{\partial n_{\text{avg}y}}{\partial s_j}}{(n_{\text{avg}x}^2 + n_{\text{avg}y}^2)\sqrt{n_{\text{avg}x}^2 + n_{\text{avg}y}^2}}, \tag{A.51}$$

and

$$\frac{\partial n_y}{\partial s_j} = \frac{\partial \frac{n_{\text{avg}y}}{\sqrt{n_{\text{avg}x}^2+n_{\text{avg}y}^2}}}{\partial s_j} = \frac{n_{\text{avg}x}^2\frac{\partial n_{\text{avg}y}}{\partial s_j} - n_{\text{avg}x}n_{\text{avg}y}\frac{\partial n_{\text{avg}x}}{\partial s_j}}{(n_{\text{avg}x}^2 + n_{\text{avg}y}^2)\sqrt{n_{\text{avg}x}^2 + n_{\text{avg}y}^2}}. \tag{A.52}$$

As the average normal $(n_{\text{avg}x}, n_{\text{avg}y})$ is evaluated by the normal of segments sharing node $i$, we take segment $im$ as an example to illustrate the procedure for obtaining the derivative of segment normal $\boldsymbol{n}_{im}$. Figure A.1 shows that segment $im$ is defined by two enrich nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_m$, and its direction and normal then can be expressed as

$$\boldsymbol{d}_{im} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_m}{\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_m)\cdot(\boldsymbol{x}_i - \boldsymbol{x}_m)}} = (\boldsymbol{d}_{imx}, \boldsymbol{d}_{imy}), \tag{A.53}$$

and

$$\boldsymbol{n}_{im} = (-\boldsymbol{d}_{imy}, \boldsymbol{d}_{imx}). \tag{A.54}$$

Then the derivative of the direction $\boldsymbol{d}_{im}$ associated with design variable $s_j$ can be derived as

$$\frac{\partial \boldsymbol{d}_{im}}{\partial s_j} = \frac{\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_m)\cdot(\boldsymbol{x}_i - \boldsymbol{x}_m)}\cdot\left(\frac{\partial \boldsymbol{x}_i}{\partial s_j} - \frac{\partial \boldsymbol{x}_m}{\partial s_j}\right) - (\boldsymbol{x}_i - \boldsymbol{x}_m)\cdot\frac{(\frac{\partial \boldsymbol{x}_i}{\partial s_j} - \frac{\partial \boldsymbol{x}_m}{\partial s_j})\cdot(\boldsymbol{x}_i - \boldsymbol{x}_m)}{\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_m)\cdot(\boldsymbol{x}_i - \boldsymbol{x}_m)}}}{(\boldsymbol{x}_i - \boldsymbol{x}_m)\cdot(\boldsymbol{x}_i - \boldsymbol{x}_m)} \tag{A.55}$$

where $\partial \boldsymbol{x}_i/\partial s_j$ and $\partial \boldsymbol{x}_m/\partial s_j$ are extracted from Equation (A.25). The derivative of $\boldsymbol{n}_{im}$ corresponding to $s_j$ can be obtained via Equation (A.54).

# Acknowledgements

On the 5th of September 2016, I took the flight CZ345 to Amsterdam, which not only opens the door for my new world view, but also makes a fresh start in my life. The Netherlands is an inclusive country, and Delft is a peaceful and lovely place, where I spent more than five years living and studying. During this period, my promotor, supervisor, friends, family and colleagues gave me their guidance, encouragement, support, and help, when I struggled with my research and life. I believe that this moment is the best opportunity to express my great gratitude to all of them.

First of all, I would like to express my deepest appreciation to my promotor, Prof. Fred van Keulen. The discussions with him are always helpful and inspiring. His valuable comments lead my papers and dissertation at a higher level, and his professional knowledge and experience in topology optimization helped me complete the most challenging part of my research. At the same time, my sincerest gratitude goes to my daily supervisor Dr. Alejandro M. Aragón, who always gave his patient guidance and immediate feedback whenever I need. He provided me with the valuable suggestions and encouragement throughout every stage of my PhD life. He taught me the significance to think critically and write logically. In my mind, he is the guy with the endless energy and enthusiasm. I learned a lot from him.

I also would like to acknowledge my Master supervisors Prof. Xu Guo (郭旭) and Prof. Weisheng Zhang (张维声), who smoothened my path towards topology optimization. I deeply appreciate that they sent me to the 11th World congress on structural and multidisciplinary optimization (WCSMO-11) in Sydney. That is the first time for me to go abroad, which firms my determination to do the PhD study overseas. I gratefully acknowledge China Scholarship Council (CSC) without which my PhD research and thesis would not have been possible.

I am very grateful to my colleagues from Structural Optimization and Mechanics section for their help and support in work and life. In no particular order: many thanks to Dr. Matthijs Langelaar for discussing topological derivatives used in topology optimization, Dr. Hans Goosen for giving a presentation about the bio-inspired materials when I started my PhD, Dr. Can Ayas for explaining the fracture mechanics concepts, Sanne van den Boom for her kind assistance with debugging and writing papers, Max van der Kolk for giving me a detailed instruction about using the cluster, Dr. Qi Wang (王奇) for his invaluable suggestions when I faced difficulties, Dr. Emiel van de Ven for offering me a C++ code based on parallel computation, Stijn Koppen for correcting my Dutch in writing and speaking, Sabiju Valiya Valappil for spending the whole day preparing traditional Indian food for me, Dr. Yong Zhang (张勇) for playing badminton and football, Arnoud Delissen for discussing functions for computing the condition number in Python package, Dongyu Liu (刘东宇) for implementing the parallel computation and finite element methods, Dr. Long Wu (吴龙) for helping me with Go/No Go report, Vibhas Mishra

for finding a nice Indian restaurant when we went to the summer school in Denmark, Rajit Ranjan for the nice discussion about Lego and 3D printing, Lidan Zhang (张力丹) for her help with topology optimization, Yuheng Yan (延昱衡) for accessing the cluster for helping me download the simulation results. I warmly acknowledge the secretariats and managers in our department for their kind help with the daily administration affairs. I also would like to thank other fellows from our department: Dr. Yueting Liu (刘跃庭), Dr. Tomas Manzaneque Garcia, Kai Wu (吴开), Zichao Li(李子超), Hanqing Liu (刘翰青), Xianfeng Chen (陈先锋), Zhichao Liu (刘志超), and Minxing Xu (徐敏行).

I appreciate the wonderful time that I have spent together with all my friends in the Netherlands. Special thanks to Dr. Zongchen Li (李宗琛) and Bing Huang (黄冰) for helping me decorate my studio and playing basketball together, Dr. Yageng Li (李亚庚) for accompanying me to go to the hospital, Dr. Yuxin Liu (刘宇鑫) for guiding and training me in the gym, Dr. Françoise Bobbert for helping me to translate the summary, Dr. Yuewei Ma (马跃伟) for giving me suggestions when looking for PhD position and helping me lay wood flooring, Ran Xue (薛然) and Jiayu Li (李佳禹) for inviting me to your apartment, cycling, barbecue, and visiting museums, Dr. Bowei Fan (范博文) for playing Switch games and football, Pan Zhang (张攀) for playing tennis and badminton, Yunlong Guo (郭云龙) for helping me move from Delft to Eindhoven, Dr. Jin Chang (常进) and Yun Wan (万韵) for inviting me to your home for petting a cat, Boran Jia (贾博然) for helping me move, Yuze Yang (杨宇泽) for bringing the stuff from China, Dr. Yandong Wang (王彦栋) for visiting the market in the Hague on weekends together, Dr. DianKun Pan (潘殿坤) for helping me paint my studio, Jianing Zhu (朱家宁) for being my partner in English courses, Zhu Le (朱乐) for downloading research papers when I do not have access, Liqi Cao (曹立奇), Dr. Baozhou Zhu (朱保周), Rui Yan (延睿), Mingyan Fu (付明岩), Chao Ma (马超) for playing basketball and cooking Chinese dishes. Thanks again all of them for giving me kind help and enjoyable time. I feel fortunate that I meet your guys who make my life rich and colorful.

I will not forget my good friends in China. They made me feel warm and safe when the coronavirus COVID-19 outbreaking in Europe. When I went back to China for vacation, my friends always brought me to nice restaurants for local food. They usually say that you are so poor that you can not eat real Chinese dishes for a long time. They are Xu Gao (高旭), Dongqi Ru (卢冬奇), Yang Guo (郭洋), Mingchen Sun (孙铭晨), Jiahao Han (韩佳浩), ChuanXiao Lei (雷传霄), and Jun Wang (王俊). Special thanks to Yubo Zhang (张雨波) and Nan Zhang (张南) for sending me the local specialty from their cities. Particularly, I am grateful to a friend that I bothered a lot, Jiang Fan (蒋帆). Thanks for the considerate arrangement when I visited you every time.

Last but not least, I would like to thank my parents for their endless love and encouragement. They are always trying to offer me a better life and get a better education. Their love is in return for nothing. Their support makes me confident whenever I face failure. Their encouragement leads me overcome difficulties one after another. My gratitude also goes to all other relatives in Zhang family. No matter what happens, you will always stand firm behind me.

# Curriculum Vitæ

## Jian ZHANG

11-11-1990 Born in Anshan, China.

## Education

2009–2013 Bachelor of Science in Engineering Mechanics
China University of Mining and Technology, China

2013–2016 Master of Engineering in Engineering Mechanics
Dalian University of Technology, China
*Thesis:* A new topology optimization method based on moving morphable components
*Supervisors:* Prof. Xu Guo and Prof. Weisheng Zhang

2016-Present PhD candidate
Delft University of Technology, Netherlands
Structural Optimization and Mechanics group (SOM)
*Promotor:* Prof. Fred van Keulen
*Supervisor:* Dr. Alejandro M. Aragón
*This PhD research project was funded by China Scholarship Council (CSC).*

# List of Publications

9. **J. Zhang**, S. J. van den Boom, F. van Keulen, and A. M. Aragón, *A stable discontinuity-enriched finite element method for 3-D problems containing weak and strong discontinuities*, Computer Methods in Applied Mechanics and Engineering **355**, 1097 1123 (2019).

8. **J. Zhang**, E. Zhebel, S. J. van den Boom, D. Liu, and A. M. Aragón, *An object-oriented geometric engine design for handling discontinuities in unfitted/immersed/enriched methods*, International Journal for Numerical Methods in Engineering, Submitted, (2022).

7. **J. Zhang** and A. M. Aragón, *An improved stress recovery technique for the unfitted finite element analysis of discontinuous gradient fields*, International Journal for Numerical Methods in Engineering, doi:10.1002/nme.6825, (2021).

6. **J. Zhang**, F. van Keulen, and A. M. Aragón, *On Tailoring Fracture Resistance of Brittle Structures: A Level set Interface-enriched Topology Optimization Approach*, Computer Methods in Applied Mechanics and Engineering **388**, 114189 (2022).

5. S. J. van den Boom, **J. Zhang**, F. van Keulen, and A. M. Aragón, *An interface-enriched generalized finite element method for level set-based topology optimization*, Structural and Multidisciplinary Optimization **63**, 1 20 (2021).

4. S. J. van den Boom, **J. Zhang**, F. van Keulen, and A. M. Aragón, *A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions*, International Journal for Numerical Methods in Engineering **120**, 1163-1183 (2019).

3. R. Sharma, **J. Zhang**, M. Langelaar, F. van Keulen, and A. M. Aragón, *An improved stress recovery technique for low-order 3D finite elements*, International Journal for Numerical Methods in Engineering **114**, 88-103 (2018).

2. A. Souto, **J. Zhang**, A. M. Aragón, K. Velikov, and C. Coulais, arXiv, (2021).

1. E. De Lazzari, S. J. van den Boom, **J. Zhang**, F. van Keulen, and A. M. Aragón, *A critical view on the use of NonUniform Rational B-Splines to improve geometry representation in enriched finite element methods*, International Journal for Numerical Methods in Engineering **122**, 1195-1216 (2021).