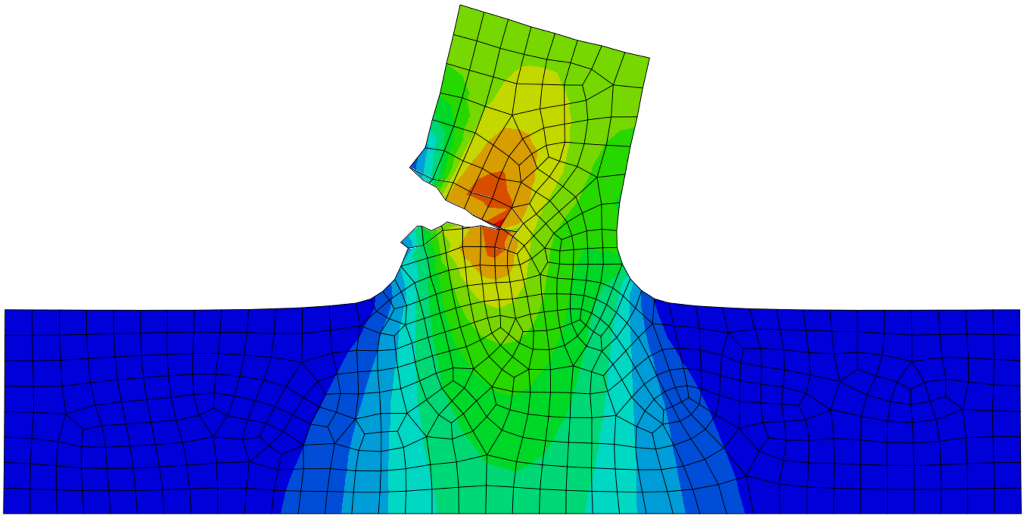
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF CIVIL ENGINEERING

The Extended Finite Element Method for Crack Propagation Problems:
Theory and Implementation Details

Serafeim Bakalakos



Master's thesis in MSc program:
Analysis and Design of Structures

Athens, October 2017

MSc THESIS

The Extended Finite Element Method for crack propagation problems

Theory and implementation details

SERAFEIM BAKALAKOS

Supervisor: Prof. Manolis Papadrakakis
Advisor: Dr. Manolis S. Georgioudakis

National Technical University of Athens
School of Civil Engineering
Department of Structural Engineering
Institute of Structural Analysis and Antiseismic Research

Athens, Greece 2017

Acknowledgements

I would like to express my deep gratitude to Prof. Manolis Papadrakakis for his encouragement, insight and helpful critique of this work, as well as his guidance throughout my academic endeavors of the last seven years. I am also indebted to Dr. Manolis Georgioudakis for the valuable time he has devoted me over the past year. Without his advice on both theoretical and technical aspects, the completion of this work would be near impossible and certainly much less entertaining. Last but not least, I wish to thank my parents for the emotional and financial support they have provided me.

Abstract

In this thesis, the eXtended Finite Element Method (XFEM) is implemented to model crack interfaces. XFEM avoids the need of conforming the finite element mesh to the crack geometry each time the crack propagates. Instead the displacement field is allowed to be discontinuous across the crack interface, by enriching the finite dimensional spaces with appropriate functions. Enrichment functions can be used to locally incorporate known behavior into the polynomial approximation of traditional FEM. For domains containing cracks, fracture mechanics provides analytic expressions of the displacement field near the crack tip, in addition to it being discontinuous across the crack body.

Predicting the crack propagation path is based on Linear Elastic Fracture Mechanics. The stress intensity factors (SIFS) are evaluated to describe the singular stress field near the crack tip and estimate the crack growth direction. A versatile and robust technique to calculate the SIFs is the J-integral method, which is suitable for computations over a finite element mesh.

This thesis is concerned with crack propagation in brittle, linear elastic materials. XFEM is applied to 2D structures under static loading. Although extending XFEM to 3D is straightforward, an accurate and efficient method to describe 3D crack geometries is the subject of ongoing research. Instead this document delves into the details of crack propagation analysis. Many algorithms and implementation details are presented for XFEM, the J-integral technique and the methods used to represent the crack geometries.

Alongside this document, C# code has been developed to implement the described theories and algorithms. This code is freely available as part of the open source software MSolve for structural analysis and design, developed by the Institute of Structural Analysis and Antiseismic Research at the National Technical University of Athens.

Table of contents

List of figures

List of tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

The Finite Element Method (FEM) is one of the most popular numerical methods for solving partial differential equations. It has been successfully applied in many areas, such as civil engineering, geomechanics, mechanical engineering, material science, electrical engineering, etc. In FEM the solution of the partial differential equation is derived from a variational form of the original boundary value problem, called the weak form. The weak form is expressed as a relationship between integrals over the whole domain. This domain is divided into non overlapping subdomains called elements. The solution is then approximated using polynomials defined over each element. This approximation works well for smooth, piecewise differentiable functions, but cannot handle more complex ones, such as discontinuities in the primary field or its gradient, singularities and high gradients.

The simulation of fracture phenomena has been a topic of growing interest over the past decades. When a structure is subjected to high or cyclic loading, the resulting stress may exceed the material strength locally and causes cracks to propagate from original imperfections. Extensive cracks lower the overall resistance of the structure and can result in failure mechanisms. FEM is ill-suited to model cracks, as they represent discontinuities in the displacement and strain fields. A number of modifications have been proposed, but they typically lack in accuracy, have high computational cost and do not yield a methodology suitable for complex constitutive models nor behavior in general three dimensions. A notable problem is the reliance on an updated mesh that conforms to the geometry of the cracks. As the crack propagates, the mesh must be recreated and refined close to the crack, such that no element is intersected by it. Not only is this computationally expensive, especially for complex geometries and 3D problems, but the mapping of the intermediate fields between the old and new mesh also results in loss of accuracy. Some element free methods have been proposed to model cracks too, such as the Element Free Galerkin method, but they come with their own set of problems.

The eXxtended Finite Element Method (XFEM) takes another approach. It enriches the original polynomial basis functions with additional ”enrichment” functions that can approximate discontinuities in the displacement field. The approximated displacement field is allowed to be discontinuous in the interior of any element, thus there is no need to generate a finite element mesh that conforms to the evolving crack’s geometry. This is of great importance, as it avoids the complexity and computational cost of regenerating and refining the mesh, while the crack propagates. Furthermore, the enrichment basis is only added locally around the crack (see Figure 1.1) and defines extra degrees of freedom. However, they are far fewer than the ones required when refining the mesh around the crack.

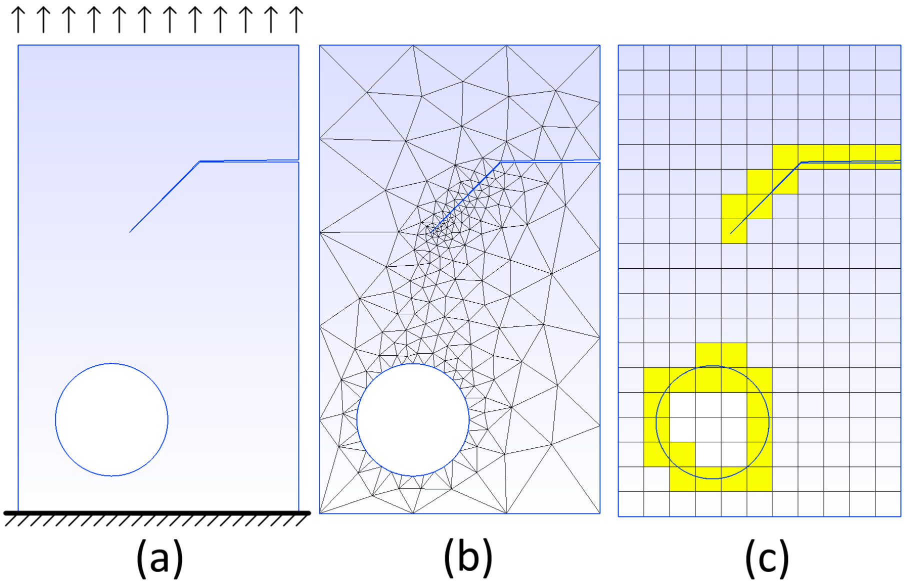


Fig. 1.1 Modeling of discontinuities and voids: a) Physical domain with a crack and a hole. b) FEM uses an adaptive mesh that must conform to the geometry of such interfaces. c) XFEM uses a uniform mesh and only elements near the interfaces are enriched.

Moreover, XFEM can be applied to various problems where the solution field is non polynomial and thus traditional FEM would fail. This is readily achievable by selecting suitable enrichment functions that can approximate the expected field and are usually derived from a priori knowledge of the solution. In crack propagation problems two classes of enrichments are used: a) discontinuous functions across the crack body, b) asymptotic functions to model a displacement field with singular gradient around the crack tips. Other enrichment functions can be used to model changes in the material properties, e.g. material interfaces and inclusions in composite materials or multiphase fluids. Voids can also be modeled by using enrichment functions that ignore the material at some regions, thus avoiding the need to conform the mesh to internal boundaries (see Figure 1.1).

1.2 Enrichment of the approximation space

Assume the well known approximation field of FEM:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i \tag{1.1}$$

where M is the set of nodes of the discretization, N_i are the basis or shape functions associated with each node and $\bar{\mathbf{u}}_i$ are the degrees of freedom (dofs), namely the nodal values of the approximated field.

The shape functions of the FEM approximation are a Partition of Unity (PU) over the domain of interest Ω. A PU can be defined as a set of global functions $f_i(\mathbf{x})$ such that that:

- they have local support, i.e. there are subdomains $\Omega_e \subseteq \Omega$ where only a subset of them has non-zero value.
- their sum is equal to 1 at each point of the solution domain

$$\sum_{i=1}^{M} f_i(\mathbf{x}) = 1 \tag{1.2}$$

Equation (1.2) expresses the ability of the PU functions to represent constants, which is crucial for the convergence of FEM. Moreover any arbitrary function ψ can be recovered as

$$\psi(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x})\psi(\mathbf{x}) \tag{1.3}$$

This provides a mathematical framework for the development of enriched solutions. In XFEM and related methods the approximation field is enriched by multiplying the problem specific enrichment functions with basis functions that define a PU. There are two basic ways to enrich the approximation function space:

1.2.1 Extrinsic enrichment

In extrinsic enrichment, the approximation field is enriched with suitable functions ψ_k:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{k=1}^{P}\sum_{j=1}^{M_k} \bar{N}_j(\mathbf{x})\psi_k(\mathbf{x})\bar{\mathbf{a}}_{jk} \tag{1.4a}$$

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{j=1}^{M_1} \bar{N}_j(\mathbf{x})\psi_1(\mathbf{x})\bar{\mathbf{a}}_{j1} + \sum_{j=1}^{M_2} \bar{N}_j(\mathbf{x})\psi_2(\mathbf{x})\bar{\mathbf{a}}_{j2} + ... \tag{1.4b}$$

where

- M is the set of nodes of the discretization, N_i are the standard shape functions and $\bar{\mathbf{u}}_i$ are the standard dofs, as in traditional FEM. In this case though, they do not necessarily coincide with the nodal displacements

- P are the unique enrichment functions and $M_k \subseteq M$ is the subset of nodes that are enriched with the function ψ_k respectively.

- $\bar{\mathbf{a}}_{kj}$ are degrees of freedom associated with the enrichment ψ_k at node j. These do not represent the nodal values of the field $\mathbf{u}(\mathbf{x})$ and will be called enriched dofs in the following.

- $\bar{N}_j$ are the shape functions that interpolate the enriched dofs over the domain. Like the standard shape function, these are usually selected from the well known Lagrange polynomials. However they do not have to be the same for a given approximation: in general $\bar{N}_j \neq N_i$ and $M_k \neq M$.

In (1.4) the first term of the right hand side is identical to the standard interpolation of the FEM. The second term, called ”enriched interpolation” in the following, is then superimposed upon the standard interpolation.

1.2.2 Intrinsic enrichment

In intrinsic enrichment, the approximation field is enhanced by including new basis functions:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{P} \hat{N}_i(\mathbf{x})\bar{\mathbf{a}}_i \tag{1.5}$$

where $\hat{\mathbf{N}}_i = \left\{\hat{N}_i^1, \hat{N}_i^1, ...\hat{N}_i^M\right\}$ is one of the P enriched bases and $\bar{\mathbf{a}}_i$ are coefficients obtained by a weighted least-squares technique, usually Moving Least Squares. Intrinsic enrichment does not introduce new dofs but has its own set of drawbacks concerning accuracy and computational complexity. It is beyond the scope of this thesis and will not be developed further.

1.3 The eXtended Finite Element Method

The eXtended Finite Element Method is a partition of unity method (PUM) that uses extrinsic enrichment to model discontinuities or other non-smooth features. The main advantage over the traditional FEM is the ability to generate the finite element mesh without considering the geometry of the discontinuity. Instead, specialized enrichment functions are used to model the jump in the approximated field and the near tip singularity of its gradient. This is particularly beneficial, when the geometry of the discontinuity moves over time. XFEM avoids the need to update the mesh and project the intermediate solution to the new mesh.

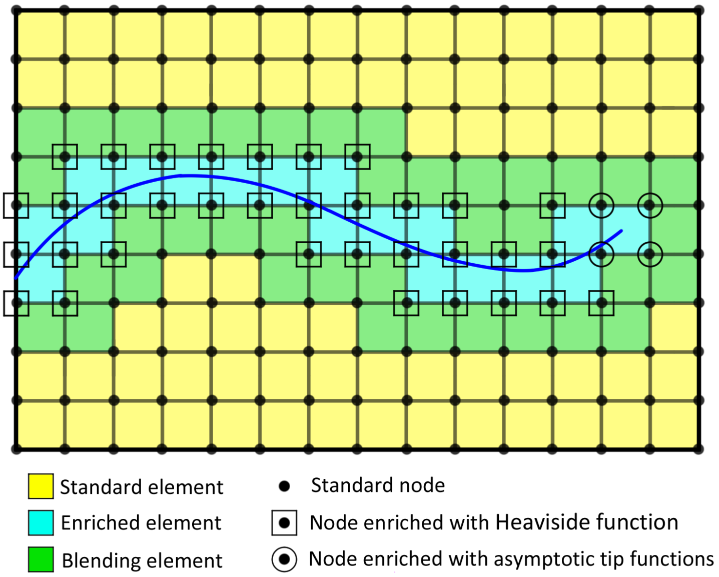

🟨 Standard element
🟦 Enriched element
🟩 Blending element
● Standard node
⊡ Node enriched with Heaviside function
⊙ Node enriched with asymptotic tip functions

Fig. 1.2 Modeling a crack in XFEM: enrichment of elements and nodes

In contrast with other PUMs, XFEM uses a local extrinsic enrichment of the approximation. Since the effects of discontinuities are generally local, the enrichment can be confined to specific zones instead of the whole domain of the solution. This approach improves the numerical solution by saving significant amounts of computing time, memory storage and avoiding other issues. As shown in Figure 1.2, only nodes around the discontinuity are enriched.

Three types of elements can be discerned: ”standard” elements where no node is enriched, ”enriched” elements where all nodes are enriched and ”blending” elements where only some nodes are enriched. Blending elements are an important aspect of XFEM and any other local PUM. The PU properties described in (1.2) and (1.3) do not hold in blending elements. As a result spurious terms are introduced into the approximation field and the convergence rate of XFEM is reduced. This difficulty is more pronounced when higher order finite elements are used and various techniques to tackle it have been developed over the years.

1.4 Basic enrichment functions

In this section some basic enrichment functions are introduced, that will then be used to model crack propagation using XFEM. First of all the signed distance function must be defined as:

$$\phi(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^*\| \operatorname{sign}\left(\mathbf{n}_{\Gamma_d} \cdot (\mathbf{x} - \mathbf{x}^*)\right) \tag{1.6}$$

where $\mathbf{x}^*$ is the closest point projection of $\mathbf{x}$ onto the discontinuity Γ_d, and $\mathbf{n}_{\Gamma_d}$ is the normal vector to the interface at point $\mathbf{x}^*$.

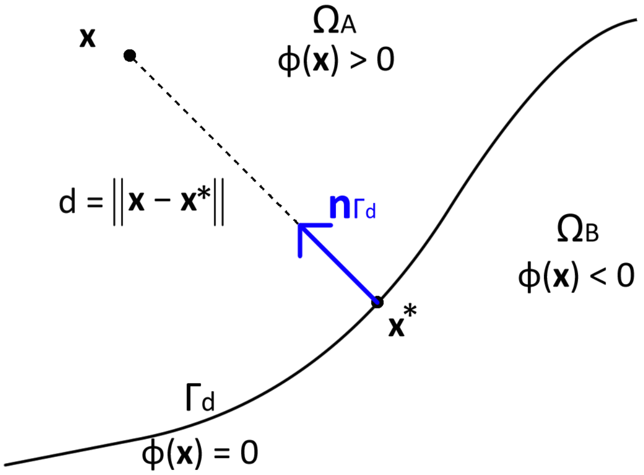


Fig. 1.3 The signed distance function.

Equation (1.6) is the general formula for calculating the signed distance of a point $\mathbf{x}$ from a curve. Depending on the method chosen to represent the discontinuity, other more efficient formulas may be available. It can be seen from (1.6) that the sign is different on the two sides of a closed interface and each subdomain can be identified by the sign of $\phi(\mathbf{x})$:

$$\phi(\mathbf{x}) \begin{cases} > 0, & \text{if } \mathbf{x} \in \Omega_A \\ = 0, & \text{if } \mathbf{x} \in \Gamma_d \\ < 0, & \text{if } \mathbf{x} \in \Omega_B \end{cases} \tag{1.7}$$

It can be shown that the norm of the gradient of the signed distance is equal to unity, that is, $\|\nabla\phi\| = 1$. Obviously, the gradient of the signed distance function at the discontinuity is indeed the unit normal $\mathbf{n}_{\Gamma_d}$ oriented to Ω_A, where $\phi(\mathbf{x}) > 0$.

1.4.1 Heaviside function

In a crack propagation problem the jump of the displacement field across the crack is referred to as a strong discontinuity. To model strong discontinuities the Heaviside function can be used as an enrichment function:

$$H(\mathbf{x}) = \begin{cases} 0, & \text{if } \phi(\mathbf{x}) < 0 \\ 1, & \text{if } \phi(\mathbf{x}) \geq 0 \end{cases} \tag{1.8}$$

or

$$H(\mathbf{x}) = \begin{cases} -1, & \text{if } \phi(\mathbf{x}) < 0 \\ 0, & \text{if } \phi(\mathbf{x}) = 0 \\ 1, & \text{if } \phi(\mathbf{x}) > 0 \end{cases} \tag{1.9}$$

where $\phi(\mathbf{x})$ is the signed distance defined in (1.6). Equation (1.8) is known as the Heaviside step function, while (1.9) as the sign function. In the following (1.9) will be used. Obviously the Heaviside enrichment function is discontinuous at the interface. Its derivative is the Dirac delta function:

$$\delta(\mathbf{x}) = \begin{cases} 0, & \text{if } \phi(\mathbf{x}) \neq 0 \\ \infty, & \text{if } \phi(\mathbf{x}) = 0 \end{cases} \tag{1.10}$$

However during numerical computations the Dirac function will only be evaluated at the integration points, which will usually be selected such that they will not lie on the discontinuity. Thus its value will be 0. In Figure 1.4 the one dimensional Heaviside function and its derivative can be seen.

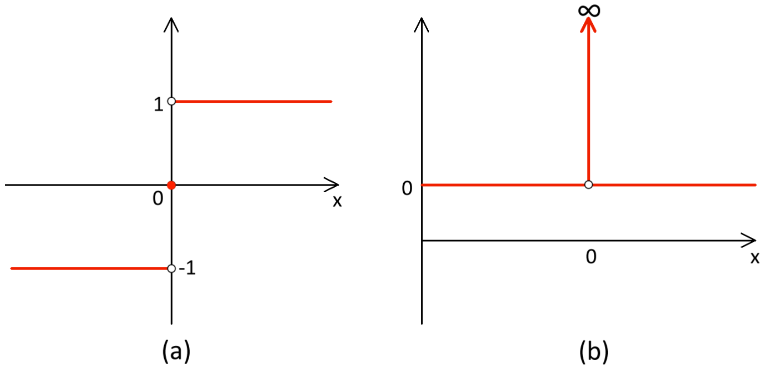


Fig. 1.4 a) The Heaviside sign function. b) The Dirac δ function.

Using the Heaviside enrichment function, the approximation field can be written as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{j=1}^{M_1} \bar{N}_j(\mathbf{x})H(\mathbf{x})\mathbf{c}_j \tag{1.11}$$

Consider the one dimensional bar with 3 linear elements as depicted in Figure 1.5a. The bar's length is equal to 1. The middle element has a strong discontinuity at the location x_c. The Heaviside sign function is used to enrich the nodes 2 and 3 according to (1.9). The signed distance is set as positive in the right side of the crack and negative at the left side. The shape functions for both the standard and enriched approximation are the well known linear shape functions $N_1(x) = 1 - x$ and $N_2(x) = x$. In Figure 1.5a the global numbering of the shape functions and the enriched basis $\bar{N}_j(x)H(x)$ are shown. By applying (1.11), the displacement value $u(x_k)$ at an enriched node k is $u(x_k) = u_k + H(\phi(x_k))c_k$. Since $H(\phi(x_k))$ is not necessarily zero, this expression is not equal to the standard degree of freedom u_k.

As explained in the example above, the Kronecker-δ property of the approximation is lost. Consequently $u(x_k) \neq u_k$, which renders the imposition of essential boundary conditions difficult. Furthermore, the interpretation of the results is more difficult as $u(x_k)$ has to be constructed correctly by evaluating all terms in the approximation. It is therefore desirable to have enrichment terms that vanish at all nodes, thereby recovering the Kronecker-δ property of standard FE approximations. This is achieved by shifting the approximation as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{j=1}^{M_1} \bar{N}_j(\mathbf{x})(H(\mathbf{x}) - H(\mathbf{x}_j))\mathbf{c}_j \tag{1.12}$$

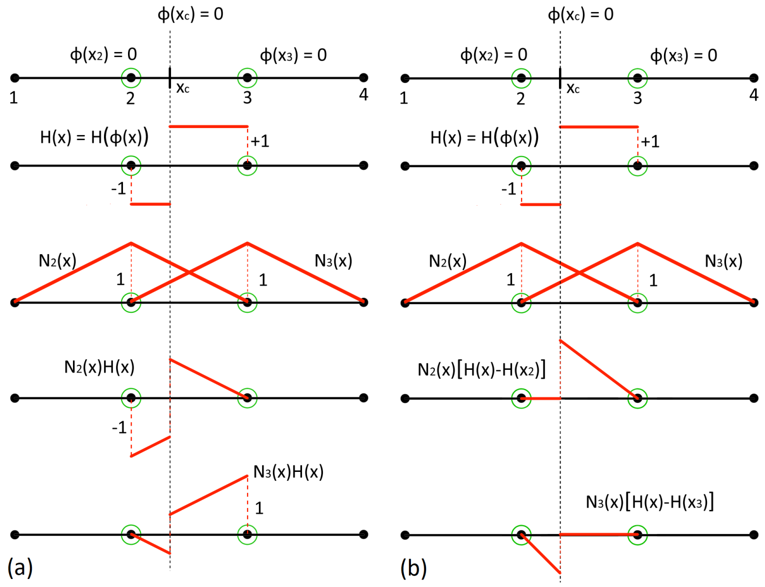


Fig. 1.5 Heaviside enrichment: a) Regular $N_j(x)\,H(x)$, b) Shifted $N_j(x)(H(x) - H(x_j))$.

Figure 1.5b depicts the shifted enrichment, where it can be seen that $u(x_k) = u_k$. This means that the standard degrees of freedom coincide with the nodal displacements. Therefore interpreting the results and more importantly applying boundary conditions on these dofs is straightforward. Moreover, it can be shown that this formulation is still able to reproduce the enrichment function $H(\mathbf{x})$ and generally any function $\psi(\mathbf{x})$ exactly [32]. Thus it is recommended to use shifted formulations and in the rest of this thesis this recommendation will be followed. Finally in Figure 1.6 the two dimensional Heaviside function and the resulting enriched basis of a quadrilateral finite element are shown.

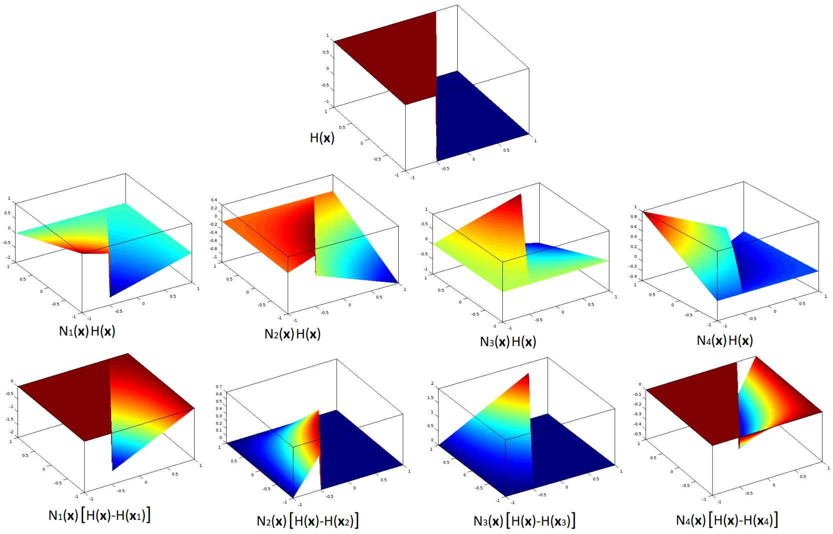


Fig. 1.6 The 2D Heaviside and the corresponding enriched basis functions.

1.4.2 Asymptotic tip functions

Cracks are open discontinuities and can be categorized into edge cracks or interior cracks (see Figure 1.7). In both crack types there is an area around the crack tip (or crack tips in case of interior cracks), where high or even singular gradients are present in the displacement field, in addition to the discontinuity. To correctly simulate the behavior around crack tips, specialized enrichment functions must be used, which model both the discontinuity across the crack and the singularity at the tip.

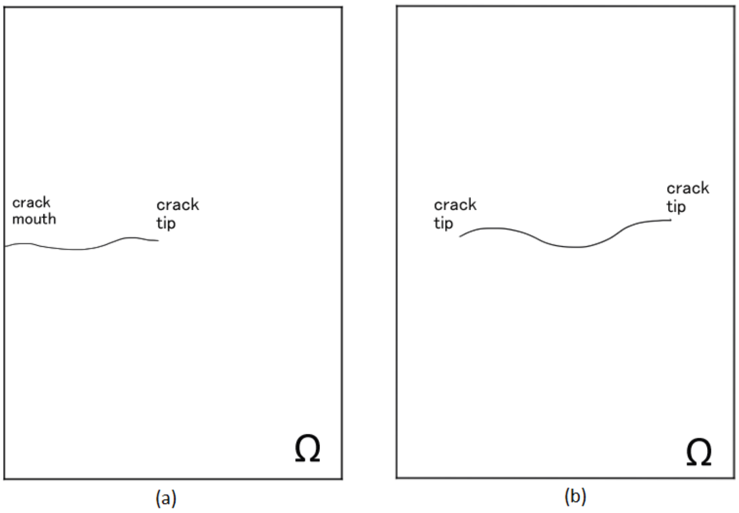


Fig. 1.7 Cracks in 2D domains: a) An edge crack. b) An interior crack

These functions are developed from analytic solutions of simple cases and depend on the particular physical model. In this thesis crack propagation in brittle, isotropic materials is considered. The appropriate enrichment functions are then extracted from Linear Elastic Fracture Mechanics (LEFM) analytic solutions. According to LEFM, the stress field is singular at the crack tip and the displacement field is contained in the span of the following four asymptotic functions:

$$\{B^a(\mathbf{x})\} = \{B^1, B^2, B^3, B^4\} = \left\{ \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right\} \tag{1.13}$$

where (r, θ) are the local polar coordinates at the crack tip as depicted in Figure 1.8.

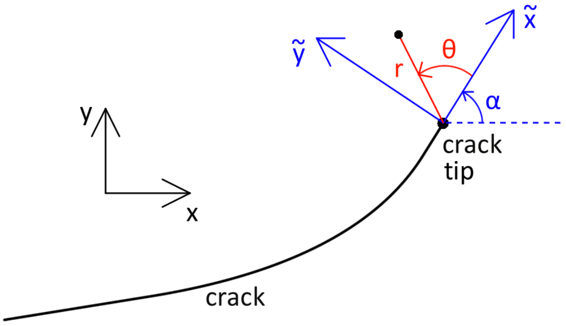


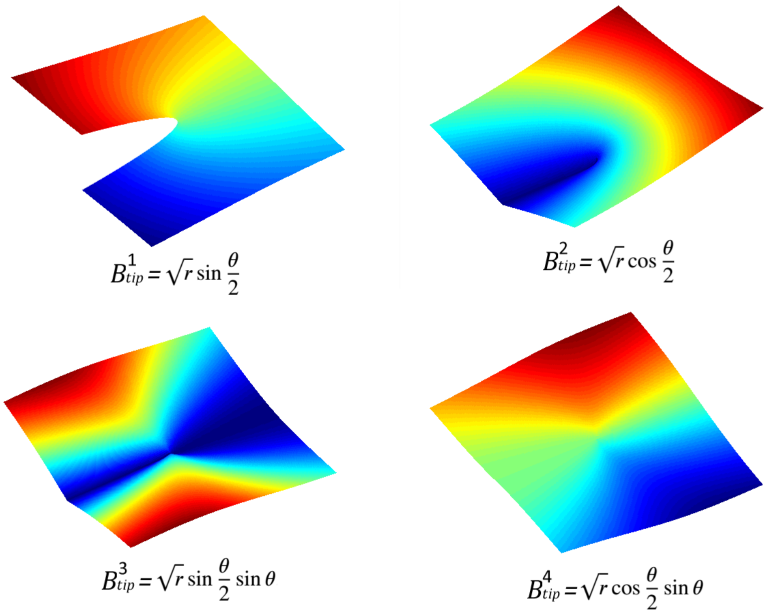
Fig. 1.8 Local cartesian and local polar coordinate system at a crack tip.

Figure 1.9 shows these four functions. They can then be used to enrich the approximation as:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{j=1}^{M_{tip}} \bar{N}_j(\mathbf{x}) \sum_{a=1}^{4} B^a(\mathbf{x})\mathbf{b}_j^a \tag{1.14}$$

where $\mathbf{b}_{aj}$ are the enriched dofs associated with each of the four asymptotic crack tip functions $B_a(\mathbf{x})$. As mentioned in Section 1.4.1 it is preferable to use a shifted enrichment:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{M} N_i(\mathbf{x})\bar{\mathbf{u}}_i + \sum_{j=1}^{M_{tip}} \bar{N}_j(\mathbf{x}) \sum_{a=1}^{4} (B^a(\mathbf{x}) - B^a(\mathbf{x}_j))\mathbf{b}_j^a \tag{1.15}$$



$B^1_{tip} = \sqrt{r}\sin\frac{\theta}{2}$

$B^2_{tip} = \sqrt{r}\cos\frac{\theta}{2}$

$B^3_{tip} = \sqrt{r}\sin\frac{\theta}{2}\sin\theta$

$B^4_{tip} = \sqrt{r}\cos\frac{\theta}{2}\sin\theta$

Fig. 1.9 Asymptotic tip functions for brittle isotropic materials.

1.5 Outline

The rest of this document is organized as follows. Chapter 2 presents the governing equations of a body containing one or more cracks and develops the XFEM formulation for the linear elastic analysis of that problem. In Chapter 3, the basics of Linear Elastic Fracture Mechanics and the J-integral method are explained. In addition, the crack propagation model for brittle materials is analyzed. Numerical integration in XFEM is not as straightforward as in FEM. The most prominent integration schemes are detailed in Chapter 4.

Chapter 5 describes in detail techniques to represent the crack geometry and its interaction with the finite element mesh, as required by XFEM. Numerical applications that demonstrate the accuracy of the proposed methods are presented in Chapter 6. Concluding remarks regarding the whole document are given in Chapter 7, while Appendix A clarifies all coordinate systems and the transformations between them that are necessary for the implementation of XFEM and the J-integral method.

Chapter 2

XFEM modeling of a fractured body

2.1 Introduction

This chapter presents how XFEM is applied to the analysis of a 2D structure that contains one or more cracks. Static loading is assumed and nonlinearities due to the constitutive law or large deformations are ignored. The purpose of this linear static analysis is to obtain the discontinuous displacement, strain and stress fields.

The calculation of these fields is repeated at each iteration of a crack propagation simulation, after determining the current configuration of the crack's geometry. Then the crack grows based on the theory covered in the next chapter. As mentioned earlier, a key advantage of XFEM is that all computations are performed over the same finite element mesh, which avoids having to conform the mesh to the crack interface each time the crack grows and projecting the displacement field between the old and new meshes.

2.2 Governing equations of the problem

2.2.1 The strong form

Consider the 2D body shown in Figure 2.1, where:

- Ω is the domain, Γ is its external boundary and $\mathbf{n}_\Gamma$ is the outward normal vector to Γ
- The external boundary is partitioned into $\Gamma = \Gamma_u \cup \Gamma_t$. $\tilde{\mathbf{u}}$ is the prescribed displacement on Γ_u and $\tilde{\mathbf{t}}$ is the prescribed traction on Γ_t.

- Γ_d is a discontinuity (edge crack) crossing the body and $\mathbf{n}_{\Gamma_d}$ is the normal vector to Γ_d. It is considered traction-free.

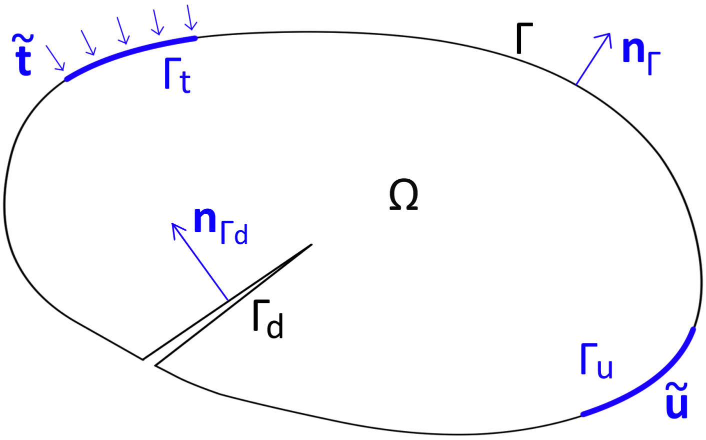


Fig. 2.1 Body with an edge crack

Assuming that the displacements remain small, the kinematic equations consist of the strain-displacement relation:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{u}) = \nabla_{sym}\mathbf{u} = \frac{\nabla\mathbf{u} + (\nabla\mathbf{u})^T}{2} \tag{2.1}$$

where $\mathbf{u}$ is the displacement field, $\boldsymbol{\varepsilon}$ is the strain tensor and $\nabla_{sym}\mathbf{u}$ denotes the symmetric part of the $\nabla\mathbf{u}$ tensor.

For a linear elastic material, the constitutive law is:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) \tag{2.2}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{C}$ is the 4th rank stiffness tensor, which is constant in this case (elastic material).

Given the kinematic equations, the constitutive law, the prescribed displacements and tractions on the external boundaries, the traction free condition on the discontinuity and the body force per unit volume $\mathbf{b} = \mathbf{b}(\mathbf{x})$ applied to Ω, the strong form of the problem can then be posed as:

Find $\mathbf{u}(\mathbf{x})$, such that the equilibrium equation holds

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \qquad in\, \Omega \tag{2.3}$$

with boundary conditions

$$\mathbf{u} = \tilde{\mathbf{u}} \qquad on\, \Gamma_u \tag{2.4a}$$
$$\boldsymbol{\sigma} \cdot \mathbf{n}_\Gamma = \mathbf{0} \qquad on\, \Gamma_u \tag{2.4b}$$
$$\boldsymbol{\sigma} \cdot \mathbf{n}_\Gamma = \tilde{\mathbf{t}} \qquad on\, \Gamma_t \tag{2.4c}$$
$$\boldsymbol{\sigma} \cdot \mathbf{n}_{\Gamma_d} = \boldsymbol{\sigma}^+ \cdot \mathbf{n}_{\Gamma_d} = \boldsymbol{\sigma}^- \cdot \mathbf{n}_{\Gamma_d} = \mathbf{0} \qquad on\, \Gamma_d \tag{2.4d}$$

Note that the divergence of the stress tensor in (2.3) is a vector defined as

$$\nabla \cdot \boldsymbol{\sigma} = \frac{\vartheta \sigma_{ij}}{\vartheta x_j} \mathbf{e}_i = \begin{bmatrix} \dfrac{\vartheta \sigma_{xx}}{\vartheta x} + \dfrac{\vartheta \sigma_{xy}}{\vartheta y} \\ \dfrac{\vartheta \sigma_{yx}}{\vartheta x} + \dfrac{\vartheta \sigma_{yy}}{\vartheta y} \end{bmatrix} \tag{2.5}$$

2.2.2 The Divergence theorem for discontinuous domains

The Divergence theorem (or Gauss-Green theorem) is necessary to derive the weak form that is the basis of any finite element formulation. Given a continuous domain Ω, with boundary Γ and a continuous vector field $\mathbf{V}$, the integration of its divergence over the domain is equivalent to the integration of the field itself over the boundary:

$$\int_\Omega \nabla \cdot \mathbf{V} d\Omega = \int_\Gamma \mathbf{V} \cdot \mathbf{n}_\Gamma d\Gamma \tag{2.6}$$

where the divergence operator is used:

$$\nabla \cdot \mathbf{V} = \mathrm{div}\mathbf{V} = \begin{bmatrix} \vartheta/\vartheta x & \vartheta/\vartheta y \end{bmatrix} \cdot \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \frac{\vartheta V_x}{\vartheta x} + \frac{\vartheta V_y}{\vartheta y} \tag{2.7}$$

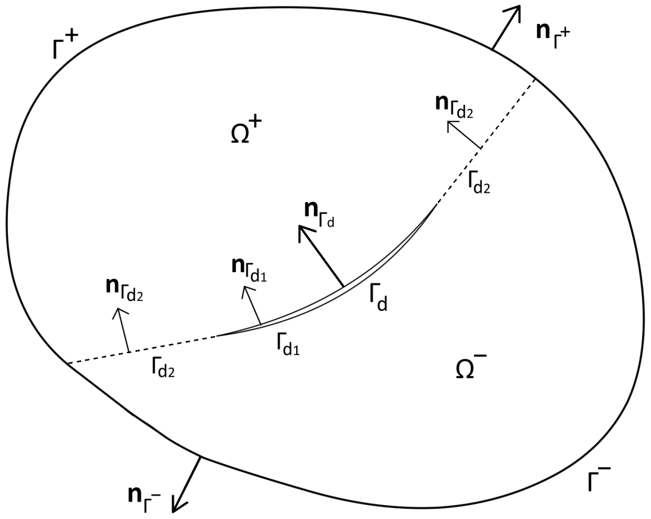


Fig. 2.2 A two-dimensional domain Ω with an interior discontinuity Γ_d

In XFEM the domain Ω is discontinuous, but we can apply the Divergence theorem by splitting Ω into continuous subdomains, as demonstrated in [34]. Figure 2.2 depicts a 2D domain Ω that is divided into two subdomains Ω^+ and Ω^- by a discontinuity Γ_d. The external boundary of domains Ω^+ and Ω^- are denoted as Γ^+ and Γ^-, with their outward unit normal vectors being $\mathbf{n}_{\Gamma^+}$ and $\mathbf{n}_{\Gamma^-}$ respectively. The curve Γ_d, with the unit normal vector $\mathbf{n}_{\Gamma d}$ oriented to Ω^+, consists of the actual discontinuity Γ_{d1} with the unit normal vector $\mathbf{n}_{\Gamma_{d1}}$ and its extension Γ_{d2} with the unit normal vector $\mathbf{n}_{\Gamma_{d2}}$, both oriented to Ω^+. We can now apply the Divergence theorem to Ω^+ and Ω^-, since they are continuous:

$$\int\limits_{\Omega^+} \nabla \cdot \mathbf{V} d\Omega = \int\limits_{\Gamma^+} \mathbf{V} \cdot \mathbf{n}_{\Gamma^+} d\Gamma + \int\limits_{\Gamma_{d1}^+} \mathbf{V}^+ \cdot \left(-\mathbf{n}_{\Gamma_{d1}}\right) d\Gamma + \int\limits_{\Gamma_{d2}^+} \mathbf{V} \cdot \left(-\mathbf{n}_{\Gamma_{d2}}\right) d\Gamma \tag{2.8a}$$

$$\int\limits_{\Omega^-} \nabla \cdot \mathbf{V} d\Omega = \int\limits_{\Gamma^-} \mathbf{V} \cdot \mathbf{n}_{\Gamma^-} d\Gamma + \int\limits_{\Gamma_{d1}^-} \mathbf{V}^- \cdot \mathbf{n}_{\Gamma_{d1}} d\Gamma + \int\limits_{\Gamma_{d2}^-} \mathbf{V} \cdot \mathbf{n}_{\Gamma_{d2}} d\Gamma \tag{2.8b}$$

where the values of $\mathbf{V}$ along the two sides of the discontinuity Γ_{d1}^+ and Γ_{d1}^- are different and denoted as $\mathbf{V}^+$ and $\mathbf{V}^-$ respectively. Since the two subdomains span the whole domain $\Omega = \Omega^+ \cup \Omega^-$ and $\Gamma = \Gamma^+ \cup \Gamma^-$ and by noticing that the contour integrals along the extension of the discontinuity Γ_{d2} in (2.8) cancel out:

$$
\begin{aligned}
\int\limits_{\Omega} \nabla \cdot \mathbf{V} d\Omega &= \int\limits_{\Omega^+} \nabla \cdot \mathbf{V} d\Omega + \int\limits_{\Omega^-} \nabla \cdot \mathbf{V} d\Omega \\
&= \int\limits_{\Gamma^+} \mathbf{V} \cdot \mathbf{n}_{\Gamma^+} d\Gamma + \int\limits_{\Gamma^-} \mathbf{V} \cdot \mathbf{n}_{\Gamma^-} d\Gamma - \int\limits_{\Gamma_{d1}^+} \mathbf{V}^+ \cdot \mathbf{n}_{\Gamma_{d1}} d\Gamma \\
&\qquad + \int\limits_{\Gamma_{d1}^-} \mathbf{V}^- \cdot \mathbf{n}_{\Gamma_{d1}} d\Gamma - \int\limits_{\Gamma_{d2}^+} \mathbf{V} \cdot \mathbf{n}_{\Gamma_{d2}} d\Gamma + \int\limits_{\Gamma_{d2}^-} \mathbf{V} \cdot \mathbf{n}_{\Gamma_{d2}} d\Gamma \\
&= \int\limits_{\Gamma} \mathbf{V} \cdot \mathbf{n}_{\Gamma} d\Gamma - \int\limits_{\Gamma_d} \left(\mathbf{V}^+ - \mathbf{V}^-\right) \cdot \mathbf{n}_{\Gamma_d} d\Gamma
\end{aligned} \tag{2.9}
$$

By defining the jump of the vector field across Γ_d as $[\![\mathbf{V}]\!] = \mathbf{V}^+ - \mathbf{V}^-$ the previous equation becomes

$$
\int\limits_{\Omega} \nabla \cdot \mathbf{V} d\Omega = \int\limits_{\Gamma} \mathbf{V} \cdot \mathbf{n}_{\Gamma} d\Gamma - \int\limits_{\Gamma_d} [\![\mathbf{V}]\!] \cdot \mathbf{n}_{\Gamma_d} d\Gamma \tag{2.10}
$$

For problems where the domain contains N_d discontinuities, the following should be used instead

$$
\int\limits_{\Omega} \nabla \cdot \mathbf{V} d\Omega = \int\limits_{\Gamma} \mathbf{V} \cdot \mathbf{n}_{\Gamma} d\Gamma - \sum_{i=1}^{N_d} \int\limits_{\Gamma_{d_i}} [\![\mathbf{V}_i]\!] \cdot \mathbf{n}_{\Gamma_{d_i}} d\Gamma \tag{2.11}
$$

2.2.3 The weak form

Let the space of admissible displacement fields (trial function space) be:

$$
U = \{\mathbf{u} \in U \mid \mathbf{u} = \tilde{\mathbf{u}} \text{ on } \Gamma_u \text{ , } \mathbf{u} \text{ is discontinuous on } \Gamma_d\} \tag{2.12}
$$

Also define the space of weighting functions (test function space) as:

$$
W = \{\mathbf{w} \in W \mid \mathbf{w} = 0 \text{ on } \Gamma_u \text{ , } \mathbf{w} \text{ is discontinuous on } \Gamma_d\} \tag{2.13}
$$

Given the same kinematic equations, constitutive law, body force and boundary conditions as described in Section 2.2.1 (actually (2.4a) is now embedded in the trial function), the weak form of the problem can be written as:

Find $\mathbf{u} \in U$ such that, $\forall$ $\mathbf{w} \in W$

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u})\, d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b}\, d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \tilde{\mathbf{t}}\, d\Gamma \tag{2.14}$$

It is shown in [21] that the above satisfies the traction-free conditions on the two crack faces. In fact the strong and weak form are equivalent. The derivation of the weak form follows, where $\boldsymbol{\sigma}$ is used instead of the more verbose $\boldsymbol{\sigma}(\mathbf{u})$ for simplicity. We start from (2.3), multiply with an arbitrary test function, integrate and apply the product rule of differentiation

$$\begin{aligned}
&\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \\
\Leftrightarrow &\int_{\Omega} \mathbf{w} \cdot (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b})\, d\Omega = 0 \\
\Leftrightarrow &\int_{\Omega} \mathbf{w} \cdot (\nabla \cdot \boldsymbol{\sigma})\, d\Omega + \int_{\Omega} \mathbf{w} \cdot \mathbf{b}\, d\Omega = 0 \\
\Leftrightarrow &\int_{\Omega} \nabla \cdot (\mathbf{w} \cdot \boldsymbol{\sigma})\, d\Omega - \int_{\Omega} \nabla \mathbf{w} : \boldsymbol{\sigma}\, d\Omega + \int_{\Omega} \mathbf{w} \cdot \mathbf{b}\, d\Omega = 0
\end{aligned} \tag{2.15}$$

We can now apply the discontinuous Divergence theorem (2.10) on the first integral of (2.15)

$$\begin{aligned}
\int_{\Omega} \nabla \cdot (\mathbf{w} \cdot \boldsymbol{\sigma})\, d\Omega &= \int_{\Gamma} \mathbf{w} \cdot \boldsymbol{\sigma} \cdot \mathbf{n}_{\Gamma}\, d\Gamma - \int_{\Gamma_d} [\![\mathbf{w} \cdot \boldsymbol{\sigma}]\!] \cdot \mathbf{n}_{\Gamma_d}\, d\Gamma \\
= \int_{\Gamma_t} \mathbf{w} \cdot \boldsymbol{\sigma} \cdot \mathbf{n}_{\Gamma}\, d\Gamma + \int_{\Gamma_u} \mathbf{w} \cdot \boldsymbol{\sigma} \cdot \mathbf{n}_{\Gamma}\, d\Gamma &- \int_{\Gamma_d} \left(\mathbf{w}^{+} \cdot \boldsymbol{\sigma}^{+} \cdot \mathbf{n}_{\Gamma_d} - \mathbf{w}^{-} \cdot \boldsymbol{\sigma}^{-} \cdot \mathbf{n}_{\Gamma_d} \right) d\Gamma
\end{aligned} \tag{2.16}$$

By imposing the boundary conditions (2.4b), (2.4c) and (2.4d) the last two integrals are eliminated from the previous equation

$$\int_{\Omega} \nabla \cdot (\mathbf{w} \cdot \boldsymbol{\sigma}) d\Omega = \int_{\Gamma_t} \mathbf{w} \cdot \tilde{\mathbf{t}} d\Gamma \tag{2.17}$$

The gradient of the test vector field can be written as

$$\nabla \mathbf{w} = \frac{\nabla \mathbf{w} + (\nabla \mathbf{w})^T}{2} + \frac{\nabla \mathbf{w} - (\nabla \mathbf{w})^T}{2} = \nabla_{sym} \mathbf{w} + \nabla_{ant} \mathbf{w} \tag{2.18}$$

where $\nabla_{sym}\mathbf{w}$ and $\nabla_{ant}\mathbf{w}$ are the symmetric and anti-symmetric parts of the $\nabla\mathbf{w}$ tensor. Since $\nabla_{ant}\mathbf{w}$ is anti-symmetric and $\boldsymbol{\sigma}$ is symmetric, their product is

$$\nabla_{ant}\mathbf{w} : \boldsymbol{\sigma} = 0 \tag{2.19}$$

Therefore the second integral of (2.15) becomes

$$\begin{aligned}\int_{\Omega} \nabla\mathbf{w} : \boldsymbol{\sigma}\, d\Omega &= \int_{\Omega} \nabla_{sym}\mathbf{w} : \boldsymbol{\sigma}\, d\Omega + \int_{\Omega} \nabla_{ant}\mathbf{w} : \boldsymbol{\sigma}\, d\Omega \\ &= \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u})\, d\Omega\end{aligned} \tag{2.20}$$

Substituting (2.17) and (2.20) into (2.15) results in the weak form

$$\begin{aligned}&\int_{\Gamma_t} \mathbf{w}\cdot\tilde{\mathbf{t}}\, d\Gamma - \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u})\, d\Omega + \int_{\Omega} \mathbf{w}\cdot\mathbf{b}\, d\Omega = 0 \\ \Leftrightarrow &\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u})\, d\Omega = \int_{\Omega} \mathbf{w}\cdot\mathbf{b}\, d\Omega + \int_{\Gamma_t} \mathbf{w}\cdot\tilde{\mathbf{t}}\, d\Gamma\end{aligned} \tag{2.21}$$

The inverse procedure will not be presented here as it is straightforward. In short, it involves reversing the previous steps and substituting appropriate test functions to eliminate any extra terms.

The weak form in matrix-vector notation

While developing the discretized equations, it is convenient to work with matrices instead of tensors. The matrix-vector forms of the strain and stress tensors are respectively

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{bmatrix} \tag{2.22a}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \tag{2.22b}$$

Using the the symmetric property of the strain and stress tensors, we can expand the colon product and write it in matrix notation:

$$\begin{aligned} \boldsymbol{\varepsilon} : \boldsymbol{\sigma} &= \varepsilon_{ij} \cdot \sigma_{ij} = \varepsilon_{xx}\sigma_{xx} + \varepsilon_{xy}\sigma_{xy} + \varepsilon_{yx}\sigma_{yx} + \varepsilon_{yy}\sigma_{yy} \\ &= \varepsilon_{xx}\sigma_{xx} + \varepsilon_{yy}\sigma_{yy} + 2\varepsilon_{xy}\sigma_{xy} = \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\sigma} \end{aligned} \tag{2.23}$$

The weak form can now be written in matrix-vector notation as

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w})^T \cdot \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega = \int_{\Omega} \mathbf{w}^T \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{w}^T \cdot \tilde{\mathbf{t}} \, d\Gamma \tag{2.24}$$

2.3 The extended finite element approximation

In any finite element method, the whole domain Ω is divided into smaller subdomains called finite elements:

$$\Omega = U\Omega_e \tag{2.25}$$

The vector and tensor fields implicated in the weak form, or rather their approximations, are defined over each element, by interpolating appropriate nodal values n the interior of each Ω_e. Then the continuous problem is reduced to identifying these discrete nodal values, commonly referred to as degrees of freedom (dofs).

2.3.1 The enriched displacement field

To model the discontinuity of the crack interface in XFEM, the nodes of elements cut by it are enriched with the Heaviside function, as demonstrated in Section 1.4.1. Also the asymptotic tip enrichments (see Section 1.4.2) are applied to nodes surrounding the crack tip, in order to model the local behavior of the displacement field there. By applying both, the XFEM approximation of the displacement field inside the domain of an element Ω_e becomes:

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}) = \sum_{i \in M_{std}} N_i(\mathbf{x})\bar{\mathbf{u}}_i \; &+ \sum_{j \in M_{crack}} N_j(\mathbf{x}) \left[H(\mathbf{x}) - H(\mathbf{x}_j)\right] \bar{\mathbf{c}}_j \\ &+ \sum_{k \in M_{tip}} N_k(\mathbf{x}) \cdot \sum_{a=1}^{4} \left[B^a(\mathbf{x}) - B^a(\mathbf{x}_j)\right] \bar{\mathbf{b}}_k^a \end{aligned} \tag{2.26}$$

where:

- $\mathbf{u}^h(\mathbf{x})$ is the finite element approximation of the displacement field.
- M_{std} is the set of all nodes of the finite element and $N_i(\mathbf{x})$ are the well known shape functions.
- $\bar{\mathbf{u}}_i = \begin{bmatrix} u_{ix} & u_{iy} \end{bmatrix}^T$ are the nodal displacements associated with node i. They will also be referred to as standard dofs.
- $M_{crack} \subset M_{std}$ and $M_{tip} \subset M_{std}$ are the sets of the element's nodes that are enriched with the Heaviside function $H(\mathbf{x})$ and the asymptotic tip functions $B^a(\mathbf{x})$ respectively.
- $N_j(\mathbf{x})$ and $N_k(\mathbf{x})$ are the shape functions used for interpolating the nodal Heaviside and tip enrichment values respectively. Usually they are selected from the same basis of shape functions as $N_i(\mathbf{x})$.
- $\bar{\mathbf{c}}_j = \begin{bmatrix} c_{jx} & c_{jy} \end{bmatrix}^T$ and $\bar{\mathbf{b}}_k^a = \begin{bmatrix} b^a_{kx} & c^a_{ky} \end{bmatrix}^T$ are the values associated with each node j that is enriched with the Heaviside and asymptotic tip functions respectively. Henceforth they will be referred to as artificial dofs.

Although (2.26) clearly describes the enriched displacement field, a slightly different notation, which includes all enrichments in a uniform way, is more convenient for further developing the method:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i \in M_{std}} N_i(\mathbf{x}) \bar{\mathbf{u}}_i + \sum_{e=0}^{4} \sum_{j \in M_e} N_j(\mathbf{x}) \left[\psi_e(\mathbf{x}) - \psi_e(\mathbf{x}_j) \right] \bar{\mathbf{a}}_{je} \tag{2.27}$$

where:

- $e = 0$ corresponds to the Heaviside enrichment and $e = 1...4$ to the asymptotic tip enrichments.
- M_e is the set of nodes enriched with the function ψ_e
- $\bar{\mathbf{a}}_{je} = \begin{bmatrix} a_{jex} & a_{jey} \end{bmatrix}^T$ are the artificial dofs associated with node j and enrichment function ψ_e

Equation (2.27) can be written in matrix-vector form as:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{N}_{std}(\mathbf{x}) \cdot \bar{\mathbf{u}} + \mathbf{N}_{enr}(\mathbf{x}) \cdot \bar{\mathbf{a}} \tag{2.28}$$

where:

$$\mathbf{N}_{std}(\mathbf{x}) = \begin{bmatrix} \cdots & N_i(\mathbf{x}) & 0 & \cdots \\ \cdots & 0 & N_i(\mathbf{x}) & \cdots \end{bmatrix} \qquad \bar{\mathbf{u}} = \begin{bmatrix} \vdots \\ u_{ix} \\ u_{iy} \\ \vdots \end{bmatrix}$$

$$\mathbf{N}_{enr}(\mathbf{x}) = \begin{bmatrix} \cdots & N_j(\mathbf{x})[\psi_e(\mathbf{x}) - \psi_e(\mathbf{x}_j)] & 0 & \cdots \\ \cdots & 0 & N_j(\mathbf{x})[\psi_e(\mathbf{x}) - \psi_e(\mathbf{x}_j)] & \cdots \end{bmatrix} \quad \bar{\mathbf{a}} = \begin{bmatrix} \vdots \\ a_{jex} \\ a_{jey} \\ \vdots \end{bmatrix}$$

After substituting the finite element approximation, (2.1) can be written in matrix-vector form as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}\left(\mathbf{u}^h\right) = \mathbf{B}_{std}(\mathbf{x}) \cdot \bar{\mathbf{u}} + \mathbf{B}_{enr}(\mathbf{x}) \cdot \bar{\mathbf{a}} \tag{2.29}$$

where the standard and enriched deformation matrices $B^{std}(\mathbf{x})$ and $B^{enr}(\mathbf{x})$ are defined as:

$$\mathbf{B}_{std}(\mathbf{x}) = \boldsymbol{\Delta} \cdot \mathbf{N}_{std}(\mathbf{x}) \tag{2.30a}$$

$$\mathbf{B}_{enr}(\mathbf{x}) = \boldsymbol{\Delta} \cdot \mathbf{N}_{enr}(\mathbf{x}) \tag{2.30b}$$

with $\boldsymbol{\Delta}$ denoting the matrix differential operator:

$$\boldsymbol{\Delta} = \begin{bmatrix} \vartheta/\vartheta x & 0 \\ 0 & \vartheta/\vartheta y \\ \vartheta/\vartheta y & \vartheta/\vartheta x \end{bmatrix} \tag{2.31}$$

The computation of the deformation matrices will be presented in detail in Section 2.5.3. For now it suffices to state that their dimensions are (3 × number of standard dofs) and (3 × number of artificial dofs) respectively.

So far the displacement field, dofs and matrices concern an individual finite element. In order to substitute them into the weak form, a global representation must be used. By collecting all dofs of the domain in a single global vector (and using different numbering than when examining each individual element) we have:

$$\bar{\mathbf{U}} = \begin{bmatrix} \cdots & u_{Ix} & u_{Iy} & \cdots & a_{Jex} & a_{Jey} & \cdots \end{bmatrix}^T \tag{2.32}$$

where:

- I, J are global indices of nodes, albeit different than the local indices used for enumerating element-wise vectors and matrices

- All artificial dofs are placed after all standard dofs. This convention will be followed for consistency from now on, however other numbering schemes may be used interchangeably.

To map the global dofs to the local dofs of each element, the boolean matrices $\mathbf{T}^{std}$ and $\mathbf{T}^{enr}$ are defined for each element as:

$$\bar{\mathbf{u}} = \mathbf{T}_{std} \cdot \bar{\mathbf{U}} \tag{2.33a}$$

$$\bar{\mathbf{a}} = \mathbf{T}_{enr} \cdot \bar{\mathbf{U}} \tag{2.33b}$$

Finite element approximation of the test function

Approximating the test (weighting) function is identical to approximating the trial function (displacement field):

$$\mathbf{w}^h(\mathbf{x}) = \sum_{i \in M_{std}} N_i(\mathbf{x}) \bar{\mathbf{w}}_i + \sum_{e=0}^{4} \sum_{j \in M_e} N_j(\mathbf{x}) \left[\psi_e(\mathbf{x}) - \psi_e(\mathbf{x}_j) \right] \bar{\mathbf{p}}_{je} \tag{2.34}$$

where $\bar{\mathbf{w}}_i$ and $\bar{\mathbf{p}}_{je}$ are the standard and enriched nodal values used for the interpolation of $\mathbf{w}^h(\mathbf{x})$ throughout the element Ω_e. The shape and enrichment functions as well as the node numbering are identical to before.

Using the same shape function and deformation matrices, the matrix-vector forms of the test function approximation and its corresponding strain are:

$$\mathbf{w}^h(\mathbf{x}) = \mathbf{N}_{std}(\mathbf{x}) \cdot \bar{\mathbf{w}} + \mathbf{N}_{enr}(\mathbf{x}) \cdot \bar{\mathbf{p}} \tag{2.35a}$$

$$\boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) = \mathbf{B}_{std}(\mathbf{x}) \cdot \bar{\mathbf{w}} + \mathbf{B}_{enr}(\mathbf{x}) \cdot \bar{\mathbf{p}} \tag{2.35b}$$

where:

$$\bar{\mathbf{w}} = \begin{bmatrix} \vdots \\ w_{ix} \\ w_{iy} \\ \vdots \end{bmatrix} \quad \bar{\mathbf{p}} = \begin{bmatrix} \vdots \\ p_{jex} \\ p_{jey} \\ \vdots \end{bmatrix}$$

Furthermore mapping from a global vector containing the test function dofs to the local dofs, is performed using the same boolean matrices:

$$\bar{\mathbf{W}} = \begin{bmatrix} \cdots & w_{Ix} & w_{Iy} & \cdots & p_{Jex} & p_{Jey} & \cdots \end{bmatrix}^T \tag{2.36a}$$

$$\bar{\mathbf{w}} = \mathbf{T}_{std} \cdot \bar{\mathbf{W}} \tag{2.36b}$$

$$\bar{\mathbf{p}} = \mathbf{T}_{enr} \cdot \bar{\mathbf{W}} \tag{2.36c}$$

2.3.2 Discretization of the governing equations

This section presents the derivation of the final linear system, by substituting the approximated solutions and matrix-vector forms of the related quantities from Section 2.3.1 into the weak form. In order to simplify the process, the arguments ($\mathbf{x}$) of all quantities will be dropped and (2.24) will be broken down to three integrals:

$$I_K = \int\limits_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w})^T \cdot \boldsymbol{\sigma}(\mathbf{u})\, d\Omega \tag{2.37a}$$

$$I_b = \int\limits_{\Omega} \mathbf{w}^T \cdot \mathbf{b}\, d\Omega \tag{2.37b}$$

$$I_t = \int\limits_{\Gamma_t} \mathbf{w}^T \cdot \tilde{\mathbf{t}}\, d\Gamma \tag{2.37c}$$

First we divide the domain into a set of M_{el} finite elements. By decomposing the first integral into the sum of integrals over each finite element and substituting the test function approximation

$$\begin{aligned} I_K &= \sum_{e \in M_{el}} \int\limits_{\Omega_e} \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right)^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right)\, d\Omega_e \\ &= \sum_{e \in M_{el}} \int\limits_{\Omega_e} \left(\bar{\mathbf{w}}^T \cdot \mathbf{B}_{std}^T + \bar{\mathbf{p}}^T \cdot \mathbf{B}_{enr}^T\right) \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right)\, d\Omega_e \\ &= \sum_{e \in M_{el}} \left[\bar{\mathbf{w}}^T \cdot \int\limits_{\Omega_e} \mathbf{B}_{std}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right)\, d\Omega_e + \bar{\mathbf{p}}^T \cdot \int\limits_{\Omega_e} \mathbf{B}_{enr}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right)\, d\Omega_e \right] \end{aligned} \tag{2.38}$$

Then we convert the nodal values to global dofs and pull them out of the sum

$$\begin{aligned} I_K &= \sum_{e \in M_{el}} \left[\bar{\mathbf{W}}^T \cdot \mathbf{T}_{std}^T \cdot \int_{\Omega_e} \mathbf{B}_{std}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right) \, d\Omega_e + \bar{\mathbf{W}}^T \cdot \mathbf{T}_{enr}^T \cdot \int_{\Omega_e} \mathbf{B}_{enr}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right) \, d\Omega_e \right] \\ &= \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} \left[\mathbf{T}_{std}^T \cdot \int_{\Omega_e} \mathbf{B}_{std}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right) \, d\Omega_e + \mathbf{T}_{enr}^T \cdot \int_{\Omega_e} \mathbf{B}_{enr}^T \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h\right) \, d\Omega_e \right] \end{aligned} \tag{2.39}$$

Then the stress of the trial field approximation is substituted in

$$\begin{aligned} I_K = \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} \Bigg[& \mathbf{T}_{std}^T \cdot \int_{\Omega_e} \mathbf{B}_{std}^T \cdot \mathbf{C} \cdot \left(\mathbf{B}_{std} \cdot \bar{\mathbf{u}} + \mathbf{B}_{enr} \cdot \bar{\mathbf{a}}\right) \, d\Omega_e \\ & + \mathbf{T}_{enr}^T \cdot \int_{\Omega_e} \mathbf{B}_{enr}^T \cdot \mathbf{C} \cdot \left(\mathbf{B}_{std} \cdot \bar{\mathbf{u}} + \mathbf{B}_{enr} \cdot \bar{\mathbf{a}}\right) \, d\Omega_e \Bigg] \\ = \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} \Bigg[& \mathbf{T}_{std}^T \cdot \left(\int_{\Omega_e} \mathbf{B}_{std}^T \cdot \mathbf{C} \cdot \mathbf{B}_{std} \, d\Omega_e \cdot \bar{\mathbf{u}} + \int_{\Omega_e} \mathbf{B}_{std}^T \cdot \mathbf{C} \cdot \mathbf{B}_{enr} \, d\Omega_e \cdot \bar{\mathbf{a}} \right) \\ & + \mathbf{T}_{enr}^T \cdot \left(\int_{\Omega_e} \mathbf{B}_{enr}^T \cdot \mathbf{C} \cdot \mathbf{B}_{std} \, d\Omega_e \cdot \bar{\mathbf{u}} + \int_{\Omega_e} \mathbf{B}_{enr}^T \cdot \mathbf{C} \cdot \mathbf{B}_{enr} \, d\Omega_e \cdot \bar{\mathbf{a}} \right) \Bigg] \end{aligned} \tag{2.40}$$

At this point it is beneficial to notice that the integrals that remain are submatrices of the element's stiffness matrix. We denote them as:

$$\mathbf{k}_{ss} = \int\limits_{\Omega_e} \mathbf{B}_{std}^T \cdot \mathbf{C} \cdot \mathbf{B}_{std} \, d\Omega_e \tag{2.41a}$$

$$\mathbf{k}_{se} = \int\limits_{\Omega_e} \mathbf{B}_{std}^T \cdot \mathbf{C} \cdot \mathbf{B}_{enr} \, d\Omega_e \tag{2.41b}$$

$$\mathbf{k}_{es} = \int\limits_{\Omega_e} \mathbf{B}_{enr}^T \cdot \mathbf{C} \cdot \mathbf{B}_{std} \, d\Omega_e \tag{2.41c}$$

$$\mathbf{k}_{ee} = \int\limits_{\Omega_e} \mathbf{B}_{enr}^T \cdot \mathbf{C} \cdot \mathbf{B}_{enr} \, d\Omega_e \tag{2.41d}$$

Substituting (2.41) and (2.33) into (2.40)

$$\begin{aligned} I_K = \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} \Big[& \mathbf{T}_{std}^T \cdot \left(\mathbf{k}_{ss} \cdot \mathbf{T}_{std} \cdot \bar{\mathbf{U}} + \mathbf{k}_{se} \cdot \mathbf{T}_{enr} \cdot \bar{\mathbf{U}} \right) \\ & + \mathbf{T}_{enr}^T \cdot \left(\mathbf{k}_{es} \cdot \mathbf{T}_{std} \cdot \bar{\mathbf{U}} + \mathbf{k}_{ee} \cdot \mathbf{T}_{enr} \cdot \bar{\mathbf{U}} \right) \Big] \\ = \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} & \left(\mathbf{T}_{std}^T \cdot \mathbf{k}_{ss} \cdot \mathbf{T}_{std} + \mathbf{T}_{std}^T \cdot \mathbf{k}_{se} \cdot \mathbf{T}_{enr} + \mathbf{T}_{enr}^T \cdot \mathbf{k}_{es} \cdot \mathbf{T}_{std} + \mathbf{T}_{enr}^T \cdot \mathbf{k}_{ee} \cdot \mathbf{T}_{enr} \right) \cdot \bar{\mathbf{U}} \end{aligned} \tag{2.42}$$

The sum represents the global stiffness matrix. By denoting

$$\mathbf{K} = \sum_{e \in M_{el}} \left(\mathbf{T}_{std}^T \cdot \mathbf{k}_{ss} \cdot \mathbf{T}_{std} + \mathbf{T}_{std}^T \cdot \mathbf{k}_{se} \cdot \mathbf{T}_{enr} + \mathbf{T}_{enr}^T \cdot \mathbf{k}_{es} \cdot \mathbf{T}_{std} + \mathbf{T}_{enr}^T \cdot \mathbf{k}_{ee} \cdot \mathbf{T}_{enr} \right) \tag{2.43}$$

the previous equation becomes

$$I_K = \bar{\mathbf{W}}^T \cdot \mathbf{K} \cdot \bar{\mathbf{U}} \tag{2.44}$$

Following similar steps, the ”body load integral” becomes

$$
\begin{aligned}
I_b &= \sum_{e \in M_{el}} \int_{\Omega_e} \left(\mathbf{w}^h\right)^T \cdot \mathbf{b} \, d\Omega_e \\
&= \sum_{e \in M_{el}} \int_{\Omega_e} \left(\bar{\mathbf{w}}^T \cdot \mathbf{N}_{std}^T + \bar{\mathbf{p}}^T \cdot \mathbf{N}_{enr}^T\right) \cdot \mathbf{b} \, d\Omega_e \\
&= \sum_{e \in M_{el}} \left(\bar{\mathbf{W}}^T \cdot \mathbf{T}_{std}^T \cdot \int_{\Omega_e} \mathbf{N}_{std}^T \cdot \mathbf{b} \, d\Omega_e + \bar{\mathbf{W}}^T \cdot \mathbf{T}_{enr}^T \cdot \int_{\Omega_e} \mathbf{N}_{enr}^T \cdot \mathbf{b} \, d\Omega_e \right) \qquad (2.45) \\
&= \bar{\mathbf{W}}^T \cdot \sum_{e \in M_{el}} \left(\mathbf{T}_{std}^T \cdot \int_{\Omega_e} \mathbf{N}_{std}^T \cdot \mathbf{b} \, d\Omega_e + \mathbf{T}_{enr}^T \cdot \int_{\Omega_e} \mathbf{N}_{enr}^T \cdot \mathbf{b} \, d\Omega_e \right) \\
&= \bar{\mathbf{W}}^T \cdot \mathbf{F}^b
\end{aligned}
$$

where $\mathbf{F}^b$ is the global body force vector

$$
\mathbf{F}^b = \sum_{e \in M_{el}} \left(\mathbf{T}_{std}^T \cdot \mathbf{f}_{std}^b + \mathbf{T}_{enr}^T \cdot \mathbf{f}_{enr}^b\right) \qquad (2.46)
$$

while the standard and enriched parts of the element’s body force vector are respectively

$$
\mathbf{f}_{std}^b = \int_{\Omega_e} \mathbf{N}_{std}^T \cdot \mathbf{b} \, d\Omega_e \qquad (2.47a)
$$

$$
\mathbf{f}_{enr}^b = \int_{\Omega_e} \mathbf{N}_{enr}^T \cdot \mathbf{b} \, d\Omega_e \qquad (2.47b)
$$

Similarly for the ”traction integral”:

$$
\begin{aligned}
I_t &= \sum_{e\in M_{el}} \int_{\Gamma_e} \left(\mathbf{w}^h\right)^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e \\
&= \sum_{e\in M_{el}} \int_{\Gamma_e} \left(\bar{\mathbf{w}}^T \cdot \mathbf{N}_{std}^T + \bar{\mathbf{p}}^T \cdot \mathbf{N}_{enr}^T\right) \cdot \tilde{\mathbf{t}}\, d\Omega_e \\
&= \sum_{e\in M_{el}} \left(\bar{\mathbf{W}}^T \cdot \mathbf{T}_{std}^T \cdot \int_{\Gamma_e} \mathbf{N}_{std}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e + \bar{\mathbf{W}}^T \cdot \mathbf{T}_{enr}^T \cdot \int_{\Gamma_e} \mathbf{N}_{enr}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e \right) \qquad (2.48) \\
&= \bar{\mathbf{W}}^T \cdot \sum_{e\in M_{el}} \left(\mathbf{T}_{std}^T \cdot \int_{\Gamma_e} \mathbf{N}_{std}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e + \mathbf{T}_{enr}^T \cdot \int_{\Gamma_e} \mathbf{N}_{enr}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e \right) \\
&= \bar{\mathbf{W}}^T \cdot \mathbf{F}^t
\end{aligned}
$$

where $\mathbf{F}^t$ is the global traction force vector

$$\mathbf{F}^t = \sum_{e\in M_{el}} \left(\mathbf{T}_{std}^T \cdot \mathbf{f}_{std}^t + \mathbf{T}_{enr}^T \cdot \mathbf{f}_{enr}^t\right) \qquad (2.49)$$

while the standard and enriched parts of the element’s traction force vector are respectively

$$\mathbf{f}_{std}^t = \int_{\Gamma_e} \mathbf{N}_{std}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e \qquad (2.50a)$$

$$\mathbf{f}_{enr}^t = \int_{\Gamma_e} \mathbf{N}_{enr}^T \cdot \tilde{\mathbf{t}}\, d\Gamma_e \qquad (2.50b)$$

Substituting (2.44), (2.45), (2.48) into the weak form (2.24)

$$\bar{\mathbf{W}}^T \cdot \mathbf{K} \cdot \bar{\mathbf{U}} = \bar{\mathbf{W}}^T \cdot \left(\mathbf{F}^b + \mathbf{F}^t\right) \qquad (2.51)$$

This equation must hold $\forall \mathbf{w}^h$, which means it must hold $\forall \bar{\mathbf{W}} \in \mathbb{R}^{\text{number of dofs}}$. Thus the final linear system is obtained

$$\mathbf{K} \cdot \bar{\mathbf{U}} = \mathbf{F} = \mathbf{F}^b + \mathbf{F}^t \qquad (2.52)$$

2.4 Selecting nodes for enrichment

Before diving into the construction of the elementary stiffness matrices and force vectors, it is important to explain how the nodes are selected for enrichment with the Heaviside or/and the asymptotic tip functions.

2.4.1 Enriching nodes with Heaviside function

For each node of the finite element mesh, its support is defined as the subdomain where the basis function associated with that node $N_i(\mathbf{x})$ is non-zero. In FEM related methods, this support is local and includes the elements having that node as a vertex. In XFEM we are also interested in the subdomain where the enriched basis function $N_i(\mathbf{x})\left[\psi(\mathbf{x}) - \psi(\mathbf{x}_i)\right]$ is non zero. For 1st order (linear) finite elements, if a node is enriched then those two subdomains coincide. An example of the local support of a node is given in Figure 2.3.

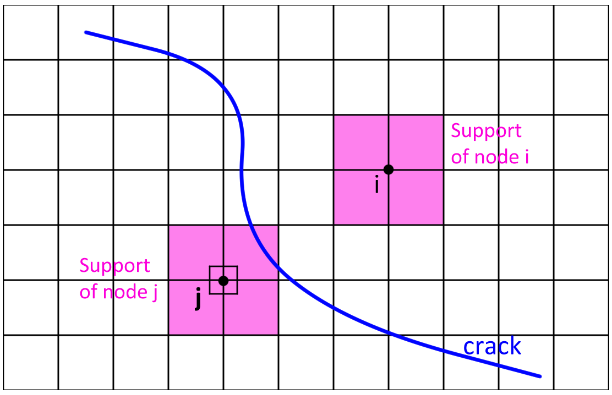


Fig. 2.3 Local support of a node

Normally a node would be enriched with the Heaviside function, if one of the elements belonging to its support is intersected by the crack, as done for node j in Figure 2.3. However this is not always optimal. Consider the two configurations depicted in Figure 2.4. In (a) the crack interface is aligned with the elements' edges and nodes 1, 2 are enriched with the Heaviside function since they lie directly on the crack. However nodes 3, 4 are not enriched, since their support is not intersected by the interface.

In (b) the interface passes in a very small band of width ε along the element's edges and nodes 1, 2 are again enriched. Based on the criterion that a node is enriched if its support is cut by the crack interface, nodes 3, 4 would also be enriched. However, this could lead to an ill-conditioned or even singular stiffness matrix.

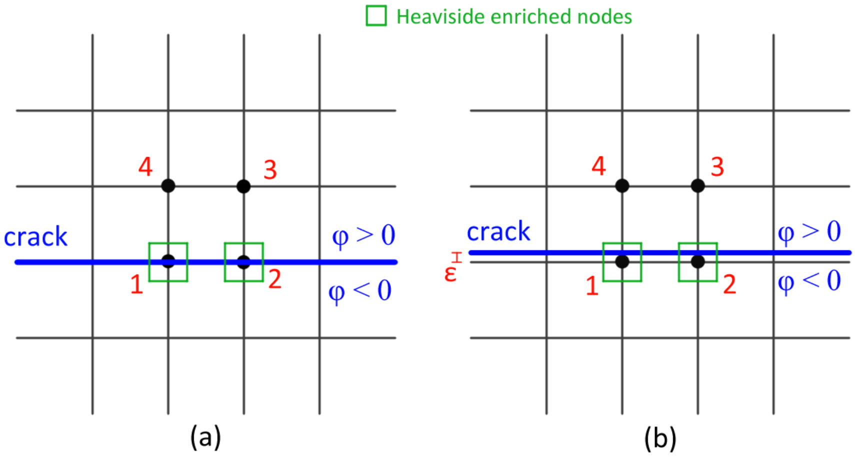


Fig. 2.4 Effects of crack near edge. a) The crack is aligned with the mesh and nodes c, d are not enriched. b) The crack is almost aligned with the mesh and nodes c, d should not be enriched to avoid a singular stiffness matrix.

To understand why, consider that the support of node 3 lies almost entirely over the crack. If an integration point $\mathbf{x}_k$ of one of the elements comprising its support is above the crack then

$$H\left(\mathbf{x}_k\right) = H\left(\varphi\left(\mathbf{x}_k\right)\right) = 1 \tag{2.53a}$$

$$N_3\left(\mathbf{x}_k\right)\left[H\left(\mathbf{x}_k\right) - H\left(\mathbf{x}_3\right)\right] = N_3\left(\mathbf{x}_k\right) \cdot (1-1) = 0 \tag{2.53b}$$

Therefore its contribution to the global stiffness matrix entries that correspond to the x and y artificial dofs at node 3 is also 0. If all the integration points in the support of node 3 are above the crack, then the whole rows (and columns) that correspond to the aforementioned dofs are zero, resulting in a singular matrix.

If very few integration points are below the crack, then linear dependence is avoided, but entries of these rows are much smaller in magnitude relative to the rest of the matrix, thus increasing its condition number. Note that without the shifted enrichment, these rows will not be zero but identical to the ones corresponding to standard dofs of node 3, since $N_3\left(\mathbf{x}_k\right) H\left(\mathbf{x}_k\right) = N_3\left(\mathbf{x}_k\right)$, so the matrix is once more singular.

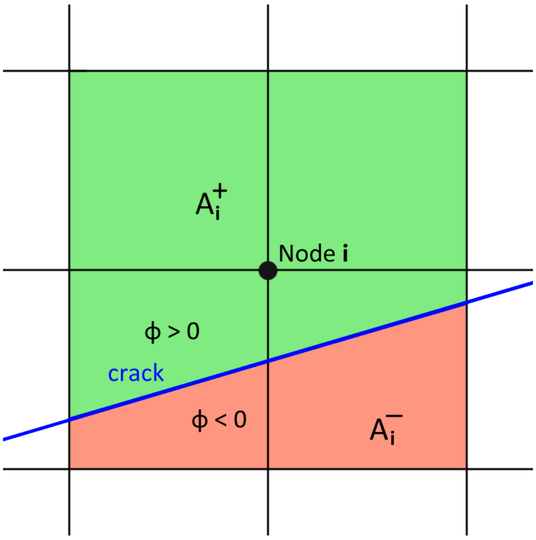


Fig. 2.5 The nodal support is partitioned into 2 areas that are used in the criterion for Heaviside enrichment.

This can be overcome by the strategy described in [22]. For each node i, whose support ω_i is intersected by the crack:

- Partition its support into the two subdomains defined by the crack: $\omega_i = \omega_i^+ \cup \omega_i^-$, with areas $A_i = A_i^+ + A_i^-$ (see Figure 2.5).
- Calculate the following ratios

$$r^+ = \frac{A_i^+}{A_i} \tag{2.54a}$$

$$r^- = \frac{A_i^-}{A_i} \tag{2.54b}$$

- Node i is not enriched with the Heaviside function if $r^+ \leq \varepsilon_{tol}$ **OR** $r^- \leq \varepsilon_{tol}$. Usually a tolerance of $\varepsilon_{tol} = 10^{-4}$ is used, as proposed in [19].

An alternative criterion is to enrich a node only if both sides (above and below the crack interface) of its support contain at least one integration point. This could be easier to implement, depending on the integration rule that is selected.

2.4.2 Enriching nodes with asymptotic tip functions

Two methods have been proposed for selecting which nodes will be enriched with the asymptotic tip functions (1.13).

- In the ”crack tip enrichment element” scheme only the nodes of the element containing the crack tip are enriched with the asymptotic tip functions, as shown in Figure 2.6a. This was the first method proposed, but the support of the tip enrichment functions vanishes, as the element size goes to zero. This hinders the convergence rate expected by using a finer mesh.

- To overcome this difficulty Laborde et al. [29] proposed the ”fixed enrichment area” scheme, which is illustrated in Figure 2.6b. According to this technique, all nodes inside a circle centered around the crack tip are enriched with the tip functions. The radius of this circle is fixed throughout the analysis, thereby guaranteeing that the support of the tip enrichment functions is independent of the element mesh size. In [29] the radius was set to 1/10th of the domain dimension, though that might be too large for practical applications. Another approach is to set the radius of the enrichment area equal to the J-integral domain radius r_J described in Section 3.5.2.

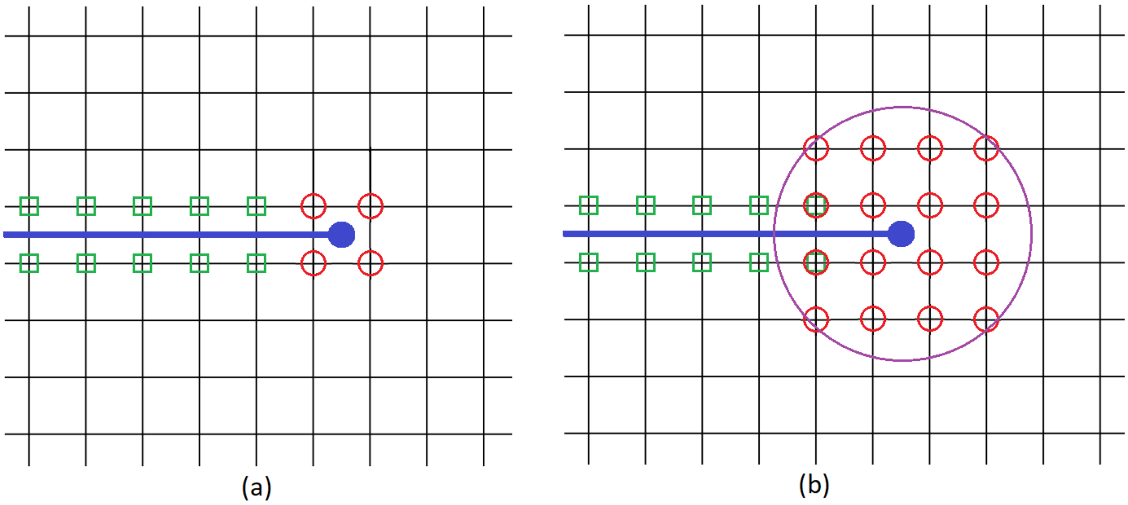


Fig. 2.6 Strategies for selecting tip enriched nodes. a) The ”crack tip enrichment element” scheme. b) The ”fixed enrichment area” scheme.

2.4.3 Enriching nodes with both Heaviside and tip functions

Figure 2.6 illustrates an interesting detail: none of the nodes of the element containing the crack tip is enriched with the Heaviside function, although their support is always intersected by the crack interface. In the fixed enrichment area case, it is possible for a node to be enriched with both the Heaviside and the asymptotic tip functions, as long as that node does not belong to the crack tip element. The first tip enrichment function introduces a discontinuity in the displacement field (see Figure 1.9). This discontinuity coincides with the crack interface ***near*** the crack tip, thus modeling it with the Heaviside enrichment function is redundant.

If the crack was a straight line throughout the domain, only the tip functions would be necessary to model it. In fact the original version of XFEM proposed in Belytschko et al. [21] did not use any Heaviside enrichments. However if a crack is modeled as a series of line segments, then the polar coordinates used in the asymptotic tip functions would have to be transformed, in order to account for the kinks of the crack. This process is computationally expensive, especially as the crack grows and includes many segments. In Moes et al. [20] the Heaviside function enrichment was introduced, which handles both kinked and curved lines elegantly and more efficiently.

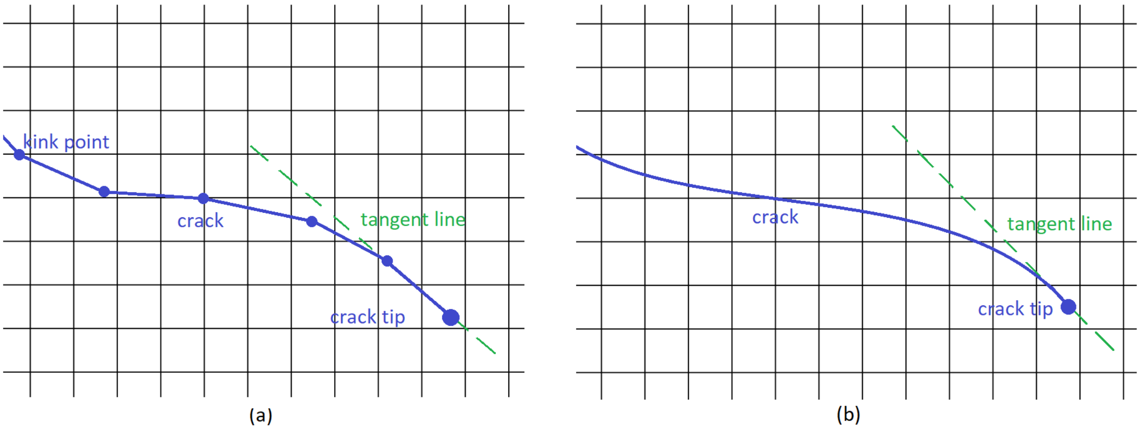


Fig. 2.7 a) Crack modeled as a series of line segments. b) Crack modeled as a curved line.

Notice that the displacement field discontinuity introduced by the first asymptotic tip function is along the tangent line shown in Figure 2.7, which always intersects at least the element containing the crack tip (”tip element”). In (a) the crack geometry is modeled as a series of line segments. The tip element is only intersected by the last crack segment, which spans beyond its boundaries and is always aligned with the tangent line. Consequently the tip element’s nodes do not have to be enriched with the Heaviside function.

If the last crack segment was completely inside the tip element, then the second to last segment would also intersect the tip element and the first asymptotic tip function's discontinuity would not suffice to model the discontinuity along the latter. Nevertheless, the length of the segments is usually chosen to be larger than the element size, therefore that configuration is unlikely. On the other hand, a curved crack geometry may be used in practice and it requires enriching the tip element's nodes with the Heaviside function for the same reason. Note that curved cracks, such as Figure 2.7(b), are typically used alongside higher order finite elements and level set methods.

2.4.4 Implementation details

This section details the main algorithms used in the numerical examples for enriching nodes with the Heaviside or asymptotic tip functions. The first algorithm listed goes through the whole procedure:

Algorithm 2.1 Enrich nodes with Heaviside and asymptotic tip functions

```
 1: Let $M_H$ be the set of nodes to be enriched with the Heaviside function.
 2: Let $M_t$ be the set of nodes to be enriched with the asymptotic tip functions.
 3: Find the elements intersected by the crack.
 4: for each of the intersected elements do
 5:     Add its nodes to $M_H$.
 6: Find the element containing the crack tip.
 7: Add the tip element's nodes to $M_t$.
 8: If possible, remove the tip element's nodes from $M_H$.
 9: if fixed area enrichment with a radius $r_t$ is employed then
10:     Find all nodes that are inside a circle centered around the crack tip, with radius $r_t$.
11:     Add these nodes to $M_t$.
12: Remove from $M_H$ nodes whose support is barely intersected by the crack.
13: Enrich all nodes in $M_H$ with the Heaviside function.
14: Enrich all nodes in $M_t$ with the asymptotic tip functions.
```

Remarks:

- The operations described in lines 3 and 6 depend on the code for the underlying representation of the crack geometry. That code is responsible for their efficient implementation, such as avoid looking at all elements of the mesh, cache intersected elements, etc. For more details see Chapter 5.

- Similarly the operation described in line 10 should be efficiently implemented by the code representing the finite element mesh.
- The condition mentioned in 8 for removing the tip element's nodes from the set of Heaviside enriched nodes is explained in Section 2.4.3. In short, nodes of the tip element do not need to be enriched with Heaviside, if the crack geometry is piecewise linear and the tip element is only intersected by the last line segment.
- Line 12 is further developed in Algorithm 2.2.
- In the case of interior cracks, there are two crack tips and the relevant steps should be repeated for both.

Algorithm 2.2 Remove enrichments that cause singular stiffness matrix

```
Let $M_H$ be the set of nodes to be enriched with the Heaviside function, obtained by the previous steps of Algorithm 2.1.
Let $M_{el}$ be a dictionary that maps elements to the areas $A_{el}^+$ and $A_{el}^-$ of their subdomains above and below the crack interface respectively.
for each node $n$ in $M_H$ do
    $A_n^+ \leftarrow 0$
    $A_n^- \leftarrow 0$
    Find the elements to which this node belongs.
    for each of these elements do
        if $M_{el}$ does not contain this element yet then
            Calculate the areas $A_{el}^+$, $A_{el}^-$ of this element's subdomains.
            Store the element, $A_{el}^+$ and $A_{el}^-$ in $M_{el}$.
        Retrieve the elements areas $A_{el}^+$ and $A_{el}^-$ from $M_{el}$.
        $A_n^+ \leftarrow A_n^+ + A_{el}^+$
        $A_n^- \leftarrow A_n^- + A_{el}^-$
    $r^+ \leftarrow \frac{A_n^+}{A_n^+ + A_n^-}$
    $r^- \leftarrow \frac{A_n^-}{A_n^+ + A_n^-}$
    if $r^+ < \varepsilon_{tol}$ OR $r^- < \varepsilon_{tol}$ then
        Remove this node from $M_H$.
```

Algorithm 2.2 lists the steps taken to ensure that nodes, whose support is barely intersected by the crack interface, are not enriched with the Heaviside function. The importance of this and any relevant formulas are explained in Section 2.4.1. This algorithm is called by Algorithm 2.1 at line 12.

Remarks:

- Since most of the elements will be investigated for multiple nodes, the computational cost can be reduced by caching the areas of their subdomains in the dictionary described in line 2. The areas are only computed once per element, stored and then retrieved from M_{el} whenever necessary.
- The operation in line 6 should be implemented by the code representing the finite element mesh efficiently (e.g. the mesh structure stores elements connected to each node, in addition to storing nodes connected to each element as usual).
- Line 9 is further developed in Algorithm 2.3.
- The tolerance ε_{tol} used in line 16 is typically set to 10^{-4} as mentioned in Section 2.4.1.

In the numerical examples, the area of an element's subdomains above and below the crack interface is calculated by dividing the element into triangles, as illustrated in Figure 2.8, such that:

- No triangle is intersected by the crack interface.
- The union of all triangles is the element and the areas of its subdomains are $A^+_{el} = \sum\limits_i A^+_i$ and $A^-_{el} = \sum\limits_j A^-_j$ respectively.

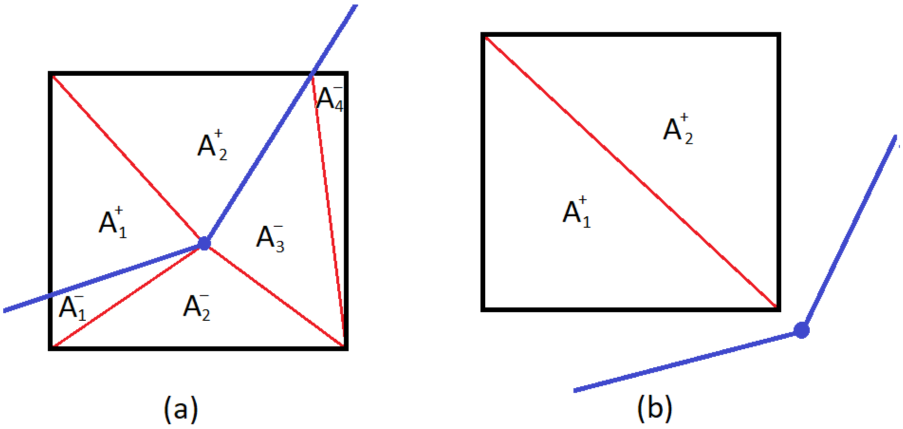


Fig. 2.8 Triangulation of a quadrilateral element that is is intersected by the crack interface (a) or not (b).

To obtain the decomposition into triangles, a constrained Dealuny algorithm may be applied (see [16]), although the special properties of the Delauny triangles are not useful here. Therefore any alternative algorithm can be used, as long as it can mesh the element's domain into subcells that conform to the crack interface. If an element is not intersected by the crack, the Delauny algorithm will still divide it into triangles (see Figure 2.8b), even though it would be more efficient to calculate the area of its polygonal boundary directly. The next listing describes the algorithm implemented for calculating the area of an element's subdomains, after decomposing it into conforming triangles:

Algorithm 2.3 Calculate the areas of an element above and below the crack interface

```
 1: $A^+_{el} \leftarrow 0$
 2: $A^-_{el} \leftarrow 0$
 3: Divide the element into triangles as depicted in Figure 2.8.
 4: for each of these triangles do
 5:     Calculate its area $A_{triangle}$ according to (2.55).
 6:     for each of the triangle's vertices do
 7:         Find its sign distance to the crack interface $\varphi$.
 8:         if $\varphi \neq 0$ then                ▷ If the node lies on the crack then $\varphi = 0$
 9:             break
10:     if $\varphi > 0$ then
11:         $A^+_{el} \leftarrow A^+_{el} + A_{triangle}$
12:     else
13:         $A^-_{el} \leftarrow A^-_{el} + A_{triangle}$
14: return $A^+_{el}, A^-_{el}$
```

The area of a triangle with vertices 1, 2, 3 can be obtained by

$$A_{triangle} = \frac{|x_1 (y_2 - y_3) + x_2 (y_3 - y_1) + x_3 (y_1 - y_2)|}{2} \tag{2.55}$$

Remarks:

- The operation in 7 is implemented by the method chosen to represent the crack geometry. For more details see Chapter 5.
- In line 12 it is assumed that $\varphi < 0$. If the element contains a kink point of a piecewise linear crack, it is possible that the triangulation will produce triangles whose vertices all lie on the crack interface. This case should be checked and handled appropriately (e.g. by using the signed distance of the triangle's centroid).

2.5 Implementation of the discretized governing equations

This section presents the derivation of the stiffness matrix of an enriched isoparametric finite element. We will denote the coordinates of the global cartesian system as $\mathbf{x}$ and the coordinates of the element's isoparametric system as $\boldsymbol{\xi}$, with $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$. It is recommended to read Appendix A first about details on the isoparametric formulation, the derivation of the formulas used here and other aspects of the coordinate systems involved. Let us define:

- A standard basis/shape function $N_i(\boldsymbol{\xi})$ associated with node i. These are selected from the usual Lagrange polynomials. According to the isoparametric formulation, the standard shape functions $N_i(\boldsymbol{\xi})$ are used to map from the natural coordinate system of the element to the global cartesian coordinate system. However, none of the following functions are involved in this mapping.

- An enriched basis function $N_{je}^{enr}(\boldsymbol{\xi})$ associated with node j and enrichment function e

$$N_{je}^{enr}(\boldsymbol{\xi}) = N_j(\boldsymbol{\xi}) \left[\psi_e(\mathbf{x}(\boldsymbol{\xi})) - \psi_e(\mathbf{x}_j)\right] \tag{2.56}$$

- An enriched shape function $N_j(\boldsymbol{\xi})$ associated with node j, used in the above enriched basis function $N_{je}^{enr}(\boldsymbol{\xi})$. These are also regular Lagrange polynomials, although they are not necessarily identical to the standard shape functions $N_i(\boldsymbol{\xi})$. When using higher order finite elements, it is preferable to enrich only a subset of the element's nodes to speed convergence, leading to different standard and enriched shape functions. This follows from the fact that the error introduced by blending elements increases with higher order elements. For linear finite elements the shape functions are identical. An example is shown in Figure 2.9.

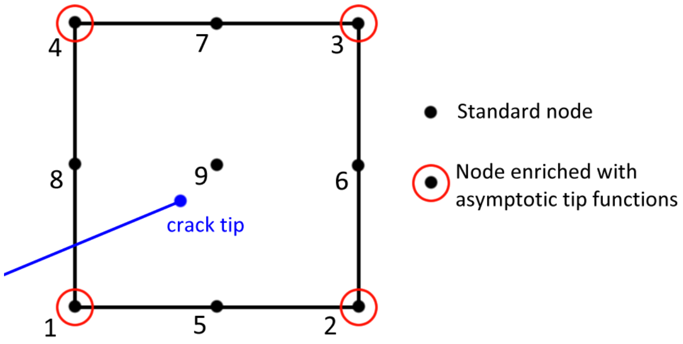


Fig. 2.9 A 9-nodes quadrilateral element enriched with asymptotic tip functions. It is preferable to only enrich the nodes used in the linear shape functions.

2.5.1 Enumerating degrees of freedom

The dof enumeration scheme that will be used for vectors and matrices in the rest of the thesis is the following:

- First number all standard dofs based on the node, then axis.
- Afterwards number all enriched dofs based on the node, then enrichment, then axis.

Thus the nodal displacement vector is of the form: $\bar{\mathbf{u}} = \begin{bmatrix} \cdots & u_{ix} & u_{iy} & \cdots & a_{jex} & a_{jey} & \cdots \end{bmatrix}^T$ where

- i, j are node indices.
- e is the enrichment index. $e = 0$ corresponds to the Heaviside function enrichment, while $e = 1, 2, 3, 4$ correspond to the asymptotic tip function enrichments.
- The vector's length is $n_{dofs}^{std} + n_{dofs}^{enr}$, with n_{dofs}^{std} and n_{dofs}^{enr} being the number of standard and artificial dofs respectively.

Note that it is not necessary to include both standard and enriched parts in the same elementary vector or matrix. As can be seen in (2.43), (2.46) and (2.49), constructing the global vectors and matrix considers the corresponding elementary subvectors or submatrices separately.

Example

Consider the quadrilateral element depicted in Figure 2.10, where node 4 is enriched with the Heaviside function and node 2 is enriched with both the Heaviside and asymptotic tip functions. The order of dofs can be seen in the nodal displacement vector:

$$\begin{aligned} \bar{\mathbf{u}} = \big[& u_{1x} \quad u_{1y} \quad u_{2x} \quad u_{2y} \quad u_{3x} \quad u_{3y} \quad u_{4x} \quad u_{4y} \quad \cdots \\ \cdots \quad & a_{20x} \quad a_{20y} \quad a_{21x} \quad a_{21y} \quad a_{22x} \quad a_{22y} \quad a_{23x} \quad a_{23y} \quad a_{24x} \quad a_{24y} \quad a_{40x} \quad a_{40y} \big]^T \end{aligned}$$

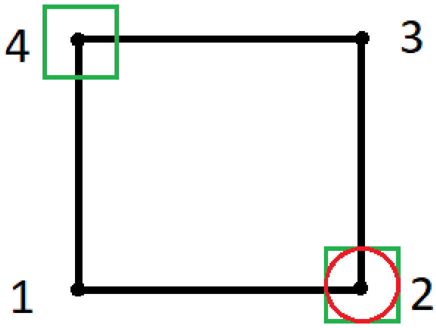


Fig. 2.10 A quadrilateral element. Node 2 is enriched with both Heaviside and asymptotic tip functions. Node 4 is enriched with only the Heaviside function.

2.5.2 Evaluating the basis functions and their derivatives

Shape functions

For the standard and enriched shape functions, there exist analytic formulas only with respect to natural coordinates. Thus to differentiate with respect to global coordinates, the scalar gradient transformations described in Appendix A.1.3 are needed. Using (A.10):

$$\begin{bmatrix} N_{i},_{x}(\xi) & N_{i},_{y}(\xi) \end{bmatrix} = \begin{bmatrix} N_{i},_{\xi}(\xi) & N_{i},_{\eta}(\xi) \end{bmatrix} \cdot J_{NG}^{-1} = \begin{bmatrix} N_{i},_{\xi}(\xi) & N_{i},_{\eta}(\xi) \end{bmatrix} \cdot \begin{bmatrix} \xi,_{x} & \xi,_{y} \\ \eta,_{x} & \eta,_{y} \end{bmatrix} \tag{2.57a}$$

$$\begin{bmatrix} N_{j},_{x}(\xi) & N_{j},_{y}(\xi) \end{bmatrix} = \begin{bmatrix} N_{j},_{\xi}(\xi) & N_{j},_{\eta}(\xi) \end{bmatrix} \cdot J_{NG}^{-1} = \begin{bmatrix} N_{j},_{\xi}(\xi) & N_{j},_{\eta}(\xi) \end{bmatrix} \cdot \begin{bmatrix} \xi,_{x} & \xi,_{y} \\ \eta,_{x} & \eta,_{y} \end{bmatrix} \tag{2.57b}$$

where

- $N_{i},_{\xi}(\xi)$, $N_{i},_{\eta}(\xi)$, $N_{j},_{\xi}(\xi)$, $N_{j},_{\eta}(\xi)$ can be easily computed from the analytic shape function formulas.
- The inverse Jacobian matrix of the isoparametric mapping J_{NG}^{-1} can be computed by applying (A.4), (A.5) and (A.6)

Heaviside enrichment

To calculate $\psi_0(\mathbf{x}) = H(\mathbf{x})$:

1. First find the global coordinates using the isoparametric mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{nodes}} N_i(\boldsymbol{\xi})\mathbf{x}_i$. Note that only the standard shape functions are used for the mapping.
2. Apply the definition of $H(\mathbf{x})$ from (1.9).

Differentiating the Heaviside function with respect to global coordinates results in the dirac function

$$H,_x(\mathbf{x}) = H,_y(\mathbf{x}) = \delta(\mathbf{x}) = \begin{cases} \infty & \text{on the crack interface} \\ 0 & \text{everywhere else} \end{cases} \tag{2.58}$$

However, using ∞ (or very large) values in the calculations will cause numerical problems. Furthermore, all quantities containing the above derivatives will be integrated numerically. It is highly unlikely that the coordinates of the integration points lie exactly on the crack interface. In fact some integration rules guarantee that it will not happen (see Section 4.3). Thus, it is reasonable to use the following simplification for the whole integration domain:

$$\begin{bmatrix} \psi_0,_x(\mathbf{x}) & \psi_0,_y(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} H,_x(\mathbf{x}) & H,_y(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{2.59}$$

Evaluating asymptotic tip enrichments

For each crack tip a local cartesian and a corresponding local polar coordinate systems are defined. Details about these coordinate systems are presented in Appendix A.2. To evaluate the asymptotic tip functions and their derivatives with respect to (w.r.t in short) global cartesian coordinates:

1. The asymptotic tip enrichments are functions of the local polar coordinates, thus converting the natural coordinates (ξ, η) to local polar coordinates (r, θ) must be done first of all:

   (a) Convert natural coordinates (of an integration point) to global cartesian coordinates, using the isoparametric mapping (A.1).

   (b) Convert the global cartesian coordinates to local cartesian coordinates, using the mapping (A.21).

   (c) Convert the local cartesian coordinates to local polar coordinates, using the mapping (A.27).

2. Then evaluate the asymptotic tip functions and their derivatives with respect to local polar coordinates

   - $$\psi_1(\mathbf{r}) = B^1(\mathbf{r}) = \sqrt{r} \cdot \sin\frac{\theta}{2} \tag{2.60a}$$
     $$\psi_{1,r}(\mathbf{r}) = \frac{1}{2\sqrt{r}} \cdot \sin\frac{\theta}{2} \tag{2.60b}$$
     $$\psi_{1,\theta}(\mathbf{r}) = \frac{1}{2} \cdot \sqrt{r} \cdot \cos\frac{\theta}{2} \tag{2.60c}$$

   - $$\psi_2(\mathbf{r}) = B^2(\mathbf{r}) = \sqrt{r} \cdot \cos\frac{\theta}{2} \tag{2.61a}$$
     $$\psi_{2,r}(\mathbf{r}) = \frac{1}{2\sqrt{r}} \cdot \cos\frac{\theta}{2} \tag{2.61b}$$
     $$\psi_{2,\theta}(\mathbf{r}) = -\frac{1}{2} \cdot \sqrt{r} \cdot \sin\frac{\theta}{2} \tag{2.61c}$$

   - $$\psi_3(\mathbf{r}) = B^3(\mathbf{r}) = \sqrt{r} \cdot \sin\frac{\theta}{2} \cdot \sin\theta \tag{2.62a}$$
     $$\psi_{3,r}(\mathbf{r}) = \frac{1}{2\sqrt{r}} \cdot \sin\frac{\theta}{2} \cdot \sin\theta \tag{2.62b}$$
     $$\psi_{3,\theta}(\mathbf{r}) = \sqrt{r}\left(\frac{1}{2} \cdot \cos\frac{\theta}{2} \cdot \sin\theta + \sin\frac{\theta}{2} \cdot \cos\theta\right) \tag{2.62c}$$

   - $$\psi_4(\mathbf{r}) = B^4(\mathbf{r}) = \sqrt{r} \cdot \cos\frac{\theta}{2} \cdot \sin\theta \tag{2.63a}$$
     $$\psi_{4,r}(\mathbf{r}) = \frac{1}{2\sqrt{r}} \cdot \cos\frac{\theta}{2} \cdot \sin\theta \tag{2.63b}$$
     $$\psi_{4,\theta}(\mathbf{r}) = \sqrt{r}\left(-\frac{1}{2} \cdot \sin\frac{\theta}{2} \cdot \sin\theta + \cos\frac{\theta}{2} \cdot \cos\theta\right) \tag{2.63c}$$

3. Finally convert the above derivatives to derivatives with respect to global coordinates using (A.38)

$$\begin{bmatrix}\psi_{e,x}(\mathbf{x}) & \psi_{e,y}(\mathbf{x})\end{bmatrix} = \begin{bmatrix}\psi_{e,r}(\mathbf{r}) & \psi_{e,\theta}(\mathbf{r})\end{bmatrix} \cdot \begin{bmatrix} cos\alpha \cdot cos\theta - sin\alpha \cdot sin\theta & sin\alpha \cdot cos\theta + cos\alpha \cdot sin\theta \\ -cos\alpha\frac{sin\theta}{r} - sin\alpha\frac{cos\theta}{r} & -sin\alpha\frac{sin\theta}{r} + cos\alpha\frac{cos\theta}{r} \end{bmatrix} \tag{2.64}$$

where α is the counter-clockwise angle from the global x axis to the local cartesian $\tilde{x}$ axis.

Alternatively the following formulas can be used to calculate the derivatives w.r.t. local cartesian coordinates $(\tilde{x}, \tilde{y})$ directly

$$\begin{aligned}
\psi_{1,\tilde{x}}(\mathbf{r}) &= -\frac{1}{2\sqrt{r}} \cdot \sin\frac{\theta}{2} & \psi_{1,\tilde{y}}(\mathbf{r}) &= \frac{1}{2\sqrt{r}} \cdot \cos\frac{\theta}{2} \\
\psi_{2,\tilde{x}}(\mathbf{r}) &= \frac{1}{2\sqrt{r}} \cdot \cos\frac{\theta}{2} & \psi_{2,\tilde{y}}(\mathbf{r}) &= \frac{1}{2\sqrt{r}} \cdot \sin\frac{\theta}{2} \\
\psi_{3,\tilde{x}}(\mathbf{r}) &= -\frac{1}{2\sqrt{r}} \cdot \sin\frac{3\theta}{2} \cdot \sin\theta & \psi_{3,\tilde{y}}(\mathbf{r}) &= \frac{1}{2\sqrt{r}} \left(\sin\frac{\theta}{2} + \sin\frac{3\theta}{2} \cdot cos\theta\right) \\
\psi_{4,\tilde{x}}(\mathbf{r}) &= -\frac{1}{2\sqrt{r}} \cdot \cos\frac{3\theta}{2} \cdot \sin\theta & \psi_{4,\tilde{y}}(\mathbf{r}) &= \frac{1}{2\sqrt{r}} \left(\cos\frac{\theta}{2} + \cos\frac{3\theta}{2} \cdot cos\theta\right)
\end{aligned} \tag{2.65}$$

Equations (2.65) incorporate the conversion from derivatives w.r.t. local polar coordinates to derivatives w.r.t local cartesian coordinates. To transform the latter into derivatives w.r.t global cartesian coordinates apply (A.36):

$$\begin{bmatrix}\psi_{e,x}(\mathbf{x}) & \psi_{e,y}(\mathbf{x})\end{bmatrix} = \begin{bmatrix}\psi_{e,\tilde{x}}(\mathbf{r}) & \psi_{e,\tilde{y}}(\mathbf{r})\end{bmatrix} \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \tag{2.66}$$

instead of using (2.64)

2.5.3 Deformation matrices

The standard and enriched basis matrices (also see (2.28)) are defined as

$$\mathbf{N}_{std}(\boldsymbol{\xi}) = \begin{bmatrix} \cdots & N_i(\boldsymbol{\xi}) & 0 & \cdots \\ \cdots & 0 & N_i(\boldsymbol{\xi}) & \cdots \end{bmatrix} \qquad \text{dimensions: } \left(2 \times n^{std}_{dofs}\right) \tag{2.67a}$$

$$\mathbf{N}_{enr}(\boldsymbol{\xi}) = \begin{bmatrix} \cdots & N^{enr}_{je}(\boldsymbol{\xi}) & 0 & \cdots \\ \cdots & 0 & N^{enr}_{je}(\boldsymbol{\xi}) & \cdots \end{bmatrix} \qquad \text{dimensions: } \left(2 \times n^{enr}_{dofs}\right) \tag{2.67b}$$

where only the submatrix corresponding to node i is explicitly written. Similarly the standard and enriched deformation matrices (also see (2.30)) and their respective dimensions are

$$\mathbf{B}_{std}(\boldsymbol{\xi}) = \boldsymbol{\Delta} \cdot \mathbf{N}_{std}(\boldsymbol{\xi}) = \begin{bmatrix} \cdots & N_i,_x(\boldsymbol{\xi}) & 0 & \cdots \\ \cdots & 0 & N_i,_y(\boldsymbol{\xi}) & \cdots \\ \cdots & N_i,_y(\boldsymbol{\xi}) & N_i,x(\boldsymbol{\xi}) & \cdots \end{bmatrix} \qquad \left(3 \times n^{std}_{dofs}\right) \tag{2.68a}$$

$$\mathbf{B}_{enr}(\boldsymbol{\xi}) = \boldsymbol{\Delta} \cdot \mathbf{N}_{enr}(\boldsymbol{\xi}) = \begin{bmatrix} \cdots & N^{enr}_{je},_x(\boldsymbol{\xi}) & 0 & \cdots \\ \cdots & 0 & N^{enr}_{je},_y(\boldsymbol{\xi}) & \cdots \\ \cdots & N^{enr}_{je},_y(\boldsymbol{\xi}) & N^{enr}_{je},_x(\boldsymbol{\xi}) & \cdots \end{bmatrix} \qquad \left(3 \times n^{enr}_{dofs}\right) \tag{2.68b}$$

where $f,_x(\boldsymbol{\xi})$ and $f,_y(\boldsymbol{\xi})$ denote differentiation with respect to x and y accordingly and

$$N^{enr}_{je},_x(\boldsymbol{\xi}) = \frac{\vartheta N_j(\boldsymbol{\xi})\left[\psi_e(\mathbf{x}(\boldsymbol{\xi})) - \psi_e(\mathbf{x}_j)\right]}{\vartheta x} = N_j,_x(\boldsymbol{\xi})\left[\psi_e(\mathbf{x}(\boldsymbol{\xi})) - \psi_e(\mathbf{x}_j)\right] + N_j(\boldsymbol{\xi})\psi_e,_x(\mathbf{x}(\boldsymbol{\xi})) \tag{2.69a}$$

$$N^{enr}_{je},_y(\boldsymbol{\xi}) = \frac{\vartheta N_j(\boldsymbol{\xi})\left[\psi_e(\mathbf{x}(\boldsymbol{\xi})) - \psi_e(\mathbf{x}_j)\right]}{\vartheta y} = N_j,_y(\boldsymbol{\xi})\left[\psi_e(\mathbf{x}(\boldsymbol{\xi})) - \psi_e(\mathbf{x}_j)\right] + N_j(\boldsymbol{\xi})\psi_e,_y(\mathbf{x}(\boldsymbol{\xi})) \tag{2.69b}$$

2.5.4 Stiffness matrices

To integrate matrices, vectors or any other function $f(\mathbf{x})$ over a domain Ω_e of an isoparametric element in the global cartesian system, we need to apply the well known ”change of variables” technique:

- The global cartesian coordinates $\mathbf{x} = (x, y)$ are replaced by the natural coordinates $\boldsymbol{\xi} = (\xi, \eta)$ using the isoparametric mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{nodes}} N_i(\boldsymbol{\xi})\mathbf{x}_i$
- The original integral I becomes

$$I_{\Omega_e} = \int_{\Omega_e} f(\mathbf{x})dxdy = \int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{\xi}) \cdot |\mathbf{J}_{NG}(\boldsymbol{\xi})| \, d\xi d\eta \tag{2.70}$$

where $|\mathbf{J}_{NG}(\boldsymbol{\xi})|$ is the determinant of the direct isoparametric mapping’s Jacobian matrix. It is also a function of the natural coordinates.

- Then numerical integration is applied

$$I_{\Omega_e} = \int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{\xi}) \cdot |J_{NG}(\boldsymbol{\xi})|\ d\xi d\eta = \sum_{k\in M_{GP}} f\left(\boldsymbol{\xi}_k\right) \cdot |\mathbf{J}_{NG}\left(\boldsymbol{\xi}_k\right)| \cdot w_k \tag{2.71}$$

where M_{GP} is the set of integration points defined by their natural coordinates $\boldsymbol{\xi}_k$ and their weight w_k. For more details on numerical integration in the context of XFEM refer to Chapter 4.In actual applications, we are concerned with 3D problems that can be simplified to 2D plain stress or plain strain problems. The formulation presented so far is accurate, with one exception. We are actually integrating over finite volumes, that is $\Omega_e = V_e = dxdydz$, where the third coordinate is the thickness of the domain $dz = t(\mathbf{x})$. Therefore the above equations are modified as

$$\begin{aligned} I_{\Omega_e} = \int_{\Omega_e} f(\mathbf{x}) d\Omega_e = \int_{\Omega_e} f(\mathbf{x}) \cdot t(\mathbf{x}) dxdy = \int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{\xi}) \cdot t(\boldsymbol{\xi}) \cdot |\mathbf{J}_{NG}(\boldsymbol{\xi})|\ d\xi d\eta \\ \Leftrightarrow I_{\Omega_e} = \sum_{k\in M_{GP}} f\left(\boldsymbol{\xi}_k\right) \cdot t\left(\boldsymbol{\xi}_k\right) \cdot |\mathbf{J}_{NG}\left(\boldsymbol{\xi}_k\right)| \cdot w_k \end{aligned} \tag{2.72}$$

The element's stiffness matrix can thus be written as

$$\Leftrightarrow \mathbf{k}^{el} = \begin{bmatrix} \mathbf{k}_{ss} & \mathbf{k}_{se} \\ \mathbf{k}_{es} & \mathbf{k}_{ee} \end{bmatrix} \tag{2.73}$$

where the submatrices $\mathbf{k}_{ss}$, $\mathbf{k}_{se}$, $\mathbf{k}_{es}$, $\mathbf{k}_{ee}$ are repeated here for convenience:

$$\mathbf{k}_{ss} = \int_{\Omega_e} \mathbf{B}^T_{std}\left(\mathbf{x}_k\right) \cdot \mathbf{C}\left(\mathbf{x}_k\right) \cdot \mathbf{B}_{std}\left(\mathbf{x}_k\right)\ d\Omega_e \tag{2.74a}$$

$$\mathbf{k}_{se} = \int_{\Omega_e} \mathbf{B}^T_{std}\left(\mathbf{x}_k\right) \cdot \mathbf{C}\left(\mathbf{x}_k\right) \cdot \mathbf{B}_{enr}\left(\mathbf{x}_k\right)\ d\Omega_e \tag{2.74b}$$

$$\mathbf{k}_{es} = \int_{\Omega_e} \mathbf{B}^T_{enr}\left(\mathbf{x}_k\right) \cdot \mathbf{C}\left(\mathbf{x}_k\right) \cdot \mathbf{B}_{std}\left(\mathbf{x}_k\right)\ d\Omega_e \tag{2.74c}$$

$$\mathbf{k}_{ee} = \int_{\Omega_e} \mathbf{B}^T_{enr}\left(\mathbf{x}_k\right) \cdot \mathbf{C}\left(\mathbf{x}_k\right) \cdot \mathbf{B}_{enr}\left(\mathbf{x}_k\right)\ d\Omega_e \tag{2.74d}$$

and by applying (2.72)

$$\mathbf{k}_{ss} = \sum_{k \in M_{GP}} \mathbf{B}_{std}^{T}\left(\xi_k\right) \cdot \mathbf{C}\left(\xi_k\right) \cdot \mathbf{B}_{std}\left(\xi_k\right) \cdot t\left(\xi_k\right) \cdot |\mathbf{J}_{NG}\left(\xi_k\right)| w_k \tag{2.75a}$$

$$\mathbf{k}_{se} = \sum_{k \in M_{GP}} \mathbf{B}_{std}^{T}\left(\xi_k\right) \cdot \mathbf{C}\left(\xi_k\right) \cdot \mathbf{B}_{enr}\left(\xi_k\right) \cdot t\left(\xi_k\right) \cdot |\mathbf{J}_{NG}\left(\xi_k\right)| w_k \tag{2.75b}$$

$$\mathbf{k}_{es} = \sum_{k \in M_{GP}} \mathbf{B}_{enr}^{T}\left(\xi_k\right) \cdot \mathbf{C}\left(\xi_k\right) \cdot \mathbf{B}_{std}\left(\xi_k\right) \cdot t\left(\xi_k\right) \cdot |\mathbf{J}_{NG}\left(\xi_k\right)| w_k \tag{2.75c}$$

$$\mathbf{k}_{ee} = \sum_{k \in M_{GP}} \mathbf{B}_{enr}^{T}\left(\xi_k\right) \cdot \mathbf{C}\left(\xi_k\right) \cdot \mathbf{B}_{enr}\left(\xi_k\right) \cdot t\left(\xi_k\right) \cdot |\mathbf{J}_{NG}\left(\xi_k\right)| w_k \tag{2.75d}$$

Remarks:

- There is no actual need to assemble the whole stiffness matrix of the element $\mathbf{k}^{el}$. As seen in (2.43) we can work directly with the submatrices.

- The stiffness matrix of each element is symmetric, because we are concerned with brittle, traction-free cracks. Hence only one of $\mathbf{k}_{es}$ and $\mathbf{k}_{se}$ needs to be computed, while the other is obtained by

$$\mathbf{k}_{se} = \mathbf{k}_{es}^{T} \tag{2.76}$$

  or vice versa, in order to reduce computation time.

2.5.5 Linear system assembly and solution

Assembling the stiffness matrix of each element into a global stiffness matrix and choosing the right solver for the the linear system depend on each other. Assembling the body and traction force vectors into a global right hand side vector is quite straightforward, since it only has to follow the same dof numbering as the global stiffness matrix. Two common strategies that are popular in FEM are:

- Assemble the elementary matrices into a global stiffness matrix in symmetric Skyline format and use a direct solver. Stiffness matrices resulting from the problem described in Section 2.2 are symmetric and positive definite, so one can use Cholesky factorization. Factorization typically introduces a lot of fill-in, thus appropriate dof numbering schemes and even the application of a renumbering step must be considered.

- Assemble the elementary matrices into a global stiffness matrix in Compressed Sparse Rows (CSR) format and use an iterative solver, such as the Preconditioned Conjugate Gradient algorithm. In iterative methods, one also has to take special care of the preconditioner matrix used.

These will not be further explained here, as a detailed presentation of either approach is out of the scope of this thesis. In the code accompanying the text the first approach has been implemented, albeit without any special numbering techniques. It is worthwhile to consider the global stiffness and right hands side vector assembly though.

One approach is to use the boolean matrices mapping global dofs to local (element) dofs, as presented in (2.33). Note that the composition of $\mathbf{T}_{std}$ and $\mathbf{T}_{enr}$ might be different, since it depends on the dof numbering schemes of both global and local dofs. With these boolean matrices defined, the assembly of the global stiffness matrix, global body force vector and global traction force vector from their elementary counterparts are described in equations (2.43), (2.46) and (2.49) respectively.

However these formulas are inefficient. The matrix-vector and especially matrix-vector multiplications have very high complexity, though this can be alleviated by storing the boolean matrices in a sparse storage format. Nevertheless the intermediate matrices and vectors that are ultimately summed in equations (2.43), (2.46) and (2.49) are still too large, having dimensions:

- (number of global dofs × number of global dofs) for the matrices
- (number of global dofs × 1) for the vectors

Consequently they require significant memory and the summation part of the formulas has high complexity:

- $O\left((\text{number of global dofs})^2\right)$ for the matrix additions
- $O(\text{number of global dofs})$ for the vector additions

A more efficent approach is to use a dictionary that maps each local dof to a global dof for each element. During assembly, after allocating memory for the global matrix and right hand side vector, each entry of the element's stiffness matrix, body force vector and traction force vector is added to the appropriate position specified by global row and column indices, which are in turn obtained by applying the dictionary to the local row and column of the entry. The process can also be parallelized while dealing with an individual element, since the global positions of the same element's entries do not overlap.

Chapter 3

Crack propagation with LEFM

3.1 Introduction

This chapter explores a model that predicts the propagation of an existing crack based on Linear Elastic Fracture Mechanics (LEFM). LEFM is appropriate for brittle materials, where the nonlinear zone ahead of the crack tip is negligible. The most notable contributions to the field are Inglis' stress analysis for bodies with elliptic holes, Griffith's fracture energy theory based on the first law of thermodynamics, the introduction of the energy release rate and the stress intensity factors (SIFs) by Irwin and Eshelby's J-integral method, which provides a robust computational framework for computing the SIFs. The following sections will elaborate on these theories and detail the computational steps required for their implementation.

3.2 Basics of Linear Elastic Fracture Mechanics

3.2.1 The stress intensity factors

Consider the infinitely spanning plate shown in Figure 3.1, which contains a crack of length $2a$ and is subjected to equalbiaxial tension. In [3] Westergard used Airy stress functions and rectangular coordinates expressed as complex numbers to find the stress distribution near the crack tip, which is illustrated in Figure 3.2. Westergaard's solution doesn't apply to uniaxial stress and requires significant work for evaluating the stress at points not on the local $\tilde{x}$ axis. Nevertheless, it provides a lot of insight about the stress field in the vicinity of the crack tip. This solution assumes that the material is elastic and the crack faces traction-free. Otherwise the singularity dominated zone around the crack tip is replaced by a plastic zone.

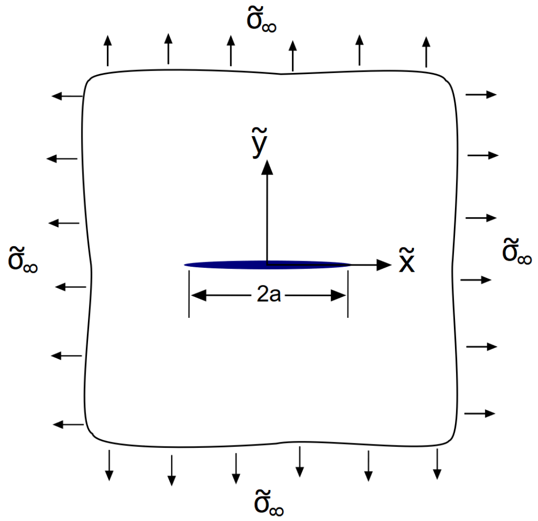


Fig. 3.1 Infinite plate with a crack under tension.

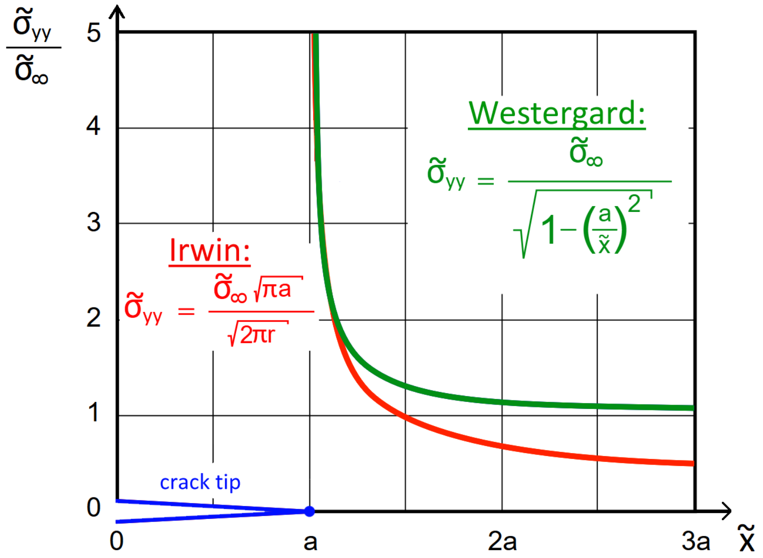


Fig. 3.2 The stress distribution near the crack tip: Westergard's exact solution is drawn in green, while Irwin's approximate solution is drawn in red.

In [5] Irwin used polar coordinates to greatly simplify the previous solution. This approximate solution diverges from Westergard's further away from the crack tip. Nevertheless, for crack propagation only the near-tip conditions are necessary and Irwin's formulas with polar coordinates not only are much easier to use, but also incorporate the stress intensity factor.

Along the $\tilde{x}$ axis of the local cartesian system defined at the crack tip, stress components in the *local cartesian system* are obtained as

$$\tilde{\sigma}_{xx}(\mathbf{r}) = \frac{\tilde{\sigma}_\infty \sqrt{\pi a}}{\sqrt{2\pi r}} \cdot \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.1a}$$

$$\tilde{\sigma}_{yy}(\mathbf{r}) = \frac{\tilde{\sigma}_\infty \sqrt{\pi a}}{\sqrt{2\pi r}} \cdot \cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.1b}$$

$$\tilde{\sigma}_{xy}(\mathbf{r}) = \frac{\tilde{\sigma}_\infty \sqrt{\pi a}}{\sqrt{2\pi r}} \cdot \cos\frac{\theta}{2}\sin\frac{\theta}{2}\sin\frac{3\theta}{2} \tag{3.1c}$$

In the equations above note that

- All three expressions include the term $\frac{1}{\sqrt{r}}$ which dictates the dependence of stress on the distance from the crack tip and reflects the singularity at $r = 0$.
- Each expressions contains a function of θ alone, separated from r and a.
- All numerators contain the expression $\tilde{\sigma}_\infty \sqrt{\pi a}$. This combination of $\tilde{\sigma}_\infty$ and a completely describes the severity of the stress state at the crack tip. Irwin [5] recognized this and introduced the term stress intensity factor (SIF) to describe the expression:

$$K = \tilde{\sigma}_\infty \sqrt{\pi a} \tag{3.2}$$

3.2.2 Strain energy release rate

During his experiments on the theoretical strength of glass rods of various diameters, Griffith observed that the tensile strength of the glass rods decreased as their diameter increased. This size dependency is caused by the presence of internal flaws in the material as proposed by Inglis in [1]. In order to correctly describe the failure of solid materials, Griffith [2] introduced a thermodynamic criterion for fracture, instead of comparing the stress field to the tensile strength. This criterion considers the total change in energy of a cracked body in terms of the crack length increase. Consider a continuum body with an existing internal crack under arbitrary loading. The first law of thermodynamics states that the change in total energy is proportional to the amount of performed work and the change of heat content:

$$\frac{d}{dt}\left(U_k + U_s + U_\Gamma\right) = \frac{d}{dt}(W + Q) \tag{3.3}$$

where U_k is the kinetic energy, U_s is the total internal strain energy, U_Γ is the surface energy, W is the external work and Q is the heat provided to the system. For an adiabatic quasi-static system $\frac{dQ}{dt} = \frac{dU_k}{dt} = 0$ and (3.3) can be simplified to

$$\frac{d}{dt}\left(U_s + U_\Gamma\right) = \frac{d}{dt}(W) \tag{3.4}$$

As all the changes with respect to time are caused by change in the flaw's length, (3.4) can be rewritten in terms of its half-length a

$$\frac{\vartheta W}{\vartheta a} = \frac{\vartheta U_s}{\vartheta a} + \frac{\vartheta U_\Gamma}{\vartheta a} \tag{3.5}$$

which represents the energy balance between the work supplied to the body by the external load, the surface energy dissipated due to the crack growth and the strain energy. The latter can be decomposed into an elastic U_s^e and a plastic U_p^e part

$$U_s = U_s^e + U_s^p \tag{3.6}$$

The potential energy of the system is defined as

$$\Pi = U_s^e - W \tag{3.7}$$

In [6] Irwin introduced the strain energy release rate, defined as

$$G = -\frac{\vartheta \Pi}{\vartheta a} \tag{3.8}$$

Using (3.5) the energy release rate can be expressed as

$$\begin{aligned} G &= -\frac{\vartheta \Pi}{\vartheta a} = \frac{\vartheta W}{\vartheta a} - \frac{\vartheta U_s^e}{\vartheta a} \\ &= \frac{\vartheta U_\Gamma}{\vartheta a} + \frac{\vartheta U_s}{\vartheta a} - \frac{\vartheta U_s^e}{\vartheta a} \\ &= \frac{\vartheta U_\Gamma}{\vartheta a} + \frac{\vartheta U_s^p}{\vartheta a} \end{aligned} \tag{3.9}$$

For brittle materials $U_s^p = 0$ and the previous reduces to

$$G = -\frac{\vartheta \Pi}{\vartheta a} = \frac{\vartheta U_\Gamma}{\vartheta a} = 2\gamma_s \tag{3.10}$$

where γ_s represents the energy required to break atomic bonds per unit surface area. Using Inglis' solution for the near tip stress field of the plate shown in Figure 3.1, the potential energy is

$$\Pi = \Pi_o - \frac{\tilde{\sigma}_\infty^2 \pi a^2}{2\mathrm{E}^*} \tag{3.11}$$

where Π_o is the potential energy of an uncracked plate and E^* is the effective Young's modulus. The energy release rate can now be calculated as

$$G = -\frac{\vartheta \Pi}{\vartheta a} = \frac{\tilde{\sigma}_\infty^2 \pi a}{\mathrm{E}^*} \tag{3.12}$$

Introducing Irwin's definition of the stress intensity factor $K = \tilde{\sigma}_\infty \sqrt{\pi a}$, the relationship between the energy release rate and the SIF is obtained

$$G = \frac{K^2}{E^*} \tag{3.13}$$

3.2.3 Fracture toughness

Griffith noticed that after a critical stress σ_f is reached, the energy balance becomes unstable, which can be observed by the crack suddenly propagating throughout the structure. Using (3.12) this failure stress can be expressed as

$$\sigma_f = \frac{\sqrt{2\gamma_s E^*}}{\pi a} = \frac{\sqrt{G_c E^*}}{\pi a} \tag{3.14}$$

where G_c is the critical energy release rate. Expressing the previous equation with respect to the stress intensity factor, Griffith's failure criterion can be written as

$$G_c = \frac{K_c^2}{E^*} \tag{3.15}$$

where

$$K_c = \sigma_f \sqrt{\pi a} \tag{3.16}$$

is Irwin's critical SIF [5]. K_c is an important material property called ”fracture toughness” and it controls how the crack grows:

- If $G < G_c \Leftrightarrow \sigma < \sigma_f \Leftrightarrow K < K_c$, the crack can propagate if the system is supplied with additional external work. This is called stable fracture.

- If $G > G_c \Leftrightarrow \sigma > \sigma_f \Leftrightarrow K > K_c$, the crack propagates without additional external work. This situation is called unstable fracture and it can lead to catastrophic failure, as a crack suddenly propagates completely through a part. In technical lingo, it is said that the crack grows ”spontaneously”.

3.2.4 Fracture modes of an existing crack

Figure 3.3 depicts the three basic ways an existing crack can be loaded and subsequently extended. Assuming the crack plane coincides with the plane $(\tilde{x}, \tilde{y})$ and the crack front coincides with the plane $(\tilde{x}, \tilde{z})$, the three modes are:

- Mode I or *opening mode* corresponds to an opening of the crack faces normal to each other, that is parallel to $\tilde{y}$ axis, under the action of tensile load. The theory presented in in the previous sections applies to this fracture mode.

- Mode II or *(in-plane) sliding mode* corresponds to a relative displacement of the crack faces parallel to $\tilde{x}$ axis, under shear stress acting parallel to the crack plane and perpendicular to the crack front.

- Mode III or *(out-of-plane) tearing mode* corresponds to a relative displacement of the crack faces parallel to $\tilde{z}$ axis, under shear stress acting parallel to the crack plane and parallel to the crack front.

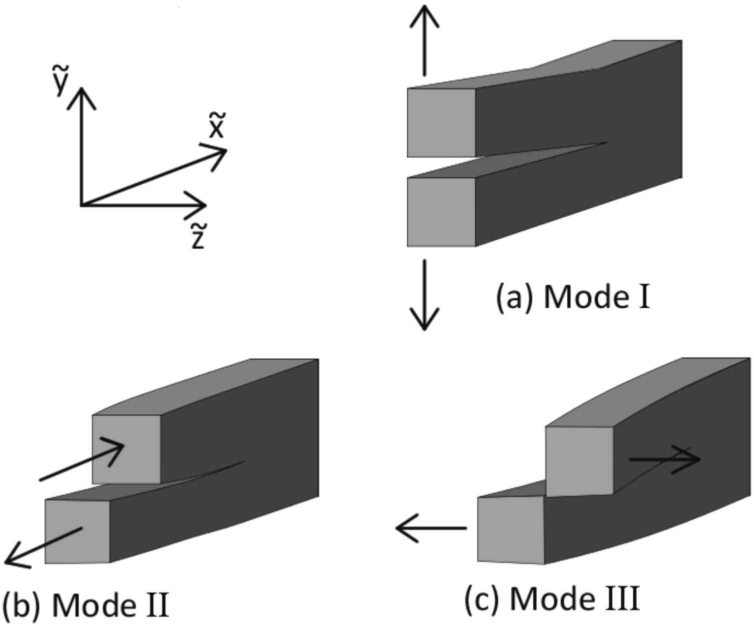


Fig. 3.3 The three basic modes of crack extension. a) Mode I: opening, b) Mode II: sliding, c) Mode III: tearing

Each fracture mode can be examined separately to obtain analytic solutions of the displacement, strain and stress fields around the crack tip. Before presenting these, the stress intensity factors for the three fracture modes must be defined as

$$K_{\mathrm{I}} = \lim_{r\to 0} \tilde{\sigma}_{yy}(r,0)\sqrt{2\pi r} \tag{3.17a}$$

$$K_{\mathrm{II}} = \lim_{r\to 0} \tilde{\sigma}_{xy}(r,0)\sqrt{2\pi r} \tag{3.17b}$$

$$K_{\mathrm{III}} = \lim_{r\to 0} \tilde{\sigma}_{yz}(r,0)\sqrt{2\pi r} \tag{3.17c}$$

The displacement and stress fields for all three modes are reviewed in [33]. In the following equations, the vector and tensor components refer to the local cartesian system defined at the crack tip, but they are expressed as functions of local polar coordinates (see Figure 3.4):

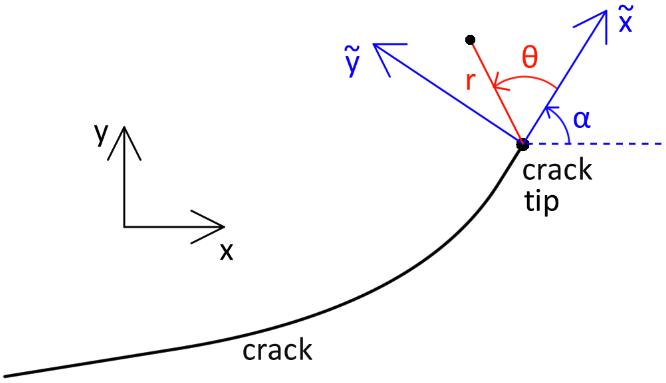


Fig. 3.4 The local cartesian (blue) and local polar (red) coordinate systems defined at the crack tip.

- In mode I loading the displacement field around the crack tip is

$$\tilde{u}_x = \frac{K_{\mathrm{I}}}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\frac{\theta}{2}\left(\kappa - 1 + 2\sin^2\frac{\theta}{2}\right) \tag{3.18a}$$

$$\tilde{u}_y = \frac{K_{\mathrm{I}}}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2}\left(\kappa + 1 - 2\cos^2\frac{\theta}{2}\right) \tag{3.18b}$$

$$\tilde{u}_z = \begin{cases} 0 & \text{plain strain} \\ -\frac{\nu\tilde{z}}{E}\left(\tilde{\sigma}_{xx} + \tilde{\sigma}_{yy}\right) & \text{plane stress} \end{cases} \tag{3.18c}$$

and the stress field is obtained as

$$\tilde{\sigma}_{xx} = \frac{K_{\mathrm{I}}}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.19a}$$

$$\tilde{\sigma}_{yy} = \frac{K_{\mathrm{I}}}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.19b}$$

$$\tilde{\sigma}_{zz} = \begin{cases} \nu\left(\tilde{\sigma}_{xx} + \tilde{\sigma}_{yy}\right) & \text{plane strain} \\ 0 & \text{plane stress} \end{cases} \tag{3.19c}$$

$$\tilde{\sigma}_{xy} = \frac{K_{\mathrm{I}}}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{3.19d}$$

$$\tilde{\sigma}_{xz} = \tilde{\sigma}_{yz} = 0 \tag{3.19e}$$

- In mode II loading the displacement field around the crack tip is

$$\tilde{u}_x = \frac{K_{\mathrm{II}}}{2\mu}\sqrt{\frac{r}{2\pi}} \sin\frac{\theta}{2}\left(\kappa + 1 + 2\cos^2\frac{\theta}{2}\right) \tag{3.20a}$$

$$\tilde{u}_y = -\frac{K_{\mathrm{II}}}{2\mu}\sqrt{\frac{r}{2\pi}} \cos\frac{\theta}{2}\left(\kappa - 1 - 2\sin^2\frac{\theta}{2}\right) \tag{3.20b}$$

$$\tilde{u}_z = \begin{cases} 0 & \text{plane strain} \\ -\frac{\nu\tilde{z}}{E}\left(\tilde{\sigma}_{xx} + \tilde{\sigma}_{yy}\right) & \text{plane stress} \end{cases} \tag{3.20c}$$

and the stress field is obtained as

$$\tilde{\sigma}_{xx} = -\frac{K_{\mathrm{II}}}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\left(2 + \cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right) \tag{3.21a}$$

$$\tilde{\sigma}_{yy} = \frac{K_{\mathrm{II}}}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{3.21b}$$

$$\tilde{\sigma}_{zz} = \begin{cases} \nu\left(\tilde{\sigma}_{xx} + \tilde{\sigma}_{yy}\right) & \text{plane strain} \\ 0 & \text{plane stress} \end{cases} \tag{3.21c}$$

$$\tilde{\sigma}_{xy} = \frac{K_{\mathrm{II}}}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.21d}$$

$$\tilde{\sigma}_{xz} = \tilde{\sigma}_{yz} = 0 \tag{3.21e}$$

- In mode III loading the displacement field around the crack tip is

$$\tilde{u}_x = \tilde{u}_y = 0 \tag{3.22a}$$

$$\tilde{u}_z = \frac{K_{\mathrm{III}}}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2} \tag{3.22b}$$

and the stress field is obtained as

$$\tilde{\sigma}_{xx} = \tilde{\sigma}_{yy} = \tilde{\sigma}_{zz} = \tilde{\sigma}_{xy} = 0 \tag{3.23a}$$

$$\tilde{\sigma}_{xz} = -\frac{K_{\mathrm{III}}}{\sqrt{2\pi r}}\sin\frac{\theta}{2} \tag{3.23b}$$

$$\tilde{\sigma}_{yz} = \frac{K_{\mathrm{III}}}{\sqrt{2\pi r}}\cos\frac{\theta}{2} \tag{3.23c}$$

where the following material properties are used:

- E is the Young's modulus and ν is the Poisson's ratio of the material.
- μ is the shear modulus of the material defined as

$$\mu = \frac{E}{2(1+\nu)} \tag{3.24}$$

- κ is Kolosov's constant defined as

$$\kappa = \begin{cases} 3 - 4\nu & \text{for plane strain problems} \\ \dfrac{3-\nu}{1+\nu} & \text{for plane stress problems} \end{cases} \tag{3.25}$$

Usually the crack is not loaded in one of the three modes, but as a combination of them, which is referred to as mixed-mode loading. In mixed-mode loading the displacement, strain and stress fields are calculated by combining the respective fields of the three modes. In LEFM this is a simple superimposition:

$$u_{ij}^{tot} = u_{ij}^{K_{\mathrm{I}}} + u_{ij}^{K_{\mathrm{II}}} + u_{ij}^{K_{\mathrm{III}}} \tag{3.26a}$$

$$\sigma_{ij}^{tot} = \sigma_{ij}^{K_{\mathrm{I}}} + \sigma_{ij}^{K_{\mathrm{II}}} + \sigma_{ij}^{K_{\mathrm{III}}} \tag{3.26b}$$

It can also be seen as linear combination with the stress intensity factors of each mode acting as coefficients. This means that the relative SIF magnitudes determine how displacements and stresses develop around an existing crack tip. Therefore, the SIFs play an important role in determining the direction of the crack growth, which will be elaborated in Section 3.4.2.

Note that the asymptotic tip functions used for enrichment in XFEM are derived from (3.18), (3.20) and (3.22) by extracting the necessary basis functions of the function space where the displacement field components belong to. Mode III loading is ignored in 2D problems, so its contribution will not be present in what follows.

3.3 The J-integral method for crack propagation

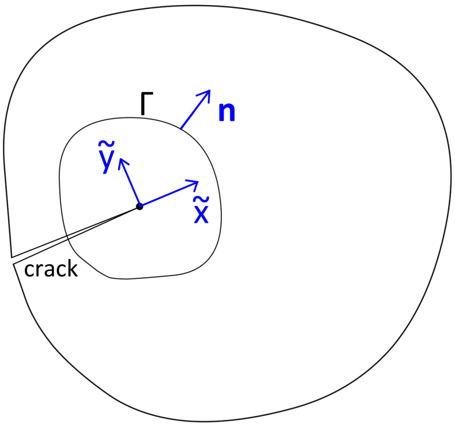


Fig. 3.5 J-integral contour around a crack tip

From Section 3.2.2 it is evident that the energy release rate and the stress intensity factors are inextricably linked to each other. Computing the SIFs is a necessary part of crack propagation analysis and there are many numerical techniques that provide solutions. One of the most flexible and popular ones is the J-integral method, which identifies the SIFs after calculating the energy release rate.

3.3.1 Original form

In [4] Eshelby defined a number of contour integrals that were path independent, based on the theorem of energy conservation. Originally they were applied for elastic analysis, but Rice [9] recognized their value in calculating the energy release rate of fractured structures. For the contour Γ depicted in Figure 3.5, the two dimensional form of the contour J-integral can be written in vector and component form respectively

$$J = \int_{\Gamma} w\,dy - \mathbf{t} \cdot \frac{\vartheta \tilde{\mathbf{u}}}{\vartheta \tilde{x}} d\Gamma \tag{3.27a}$$

$$J = \int_{\Gamma} \left(w\,\delta_{1j} - \tilde{\sigma}_{ij} \frac{\vartheta \tilde{u}_i}{\vartheta \tilde{x}} \right) \tilde{n}_j d\Gamma \tag{3.27b}$$

where

- Γ is an arbitrary closed counter-clockwise contour.
- w is the strain energy defined as

$$w = \int_{0}^{\tilde{\varepsilon}_{ij}} \tilde{\sigma}_{ij} d\tilde{\varepsilon}_{ij} = \frac{1}{2}\tilde{\sigma}_{ij}\tilde{\varepsilon}_{ij} \tag{3.28}$$

- $\tilde{u_j}$ and $\tilde{\sigma_{ij}}$ are the local cartesian components of the displacement and stress field respectively.
- $\mathbf{n}$ is the outward normal vector to the contour Γ.
- $\mathbf{t} = \tilde{\boldsymbol{\sigma}} \cdot \mathbf{n}$ is the traction vector on the ”plane” normal to $\mathbf{n}$.
- δ is the Kroenecker delta

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.29}$$

The J-integral defined in (3.27) is path independent. If it is applied on a contour around the crack tip, then it represents the variation in potential energy for an infinitesimal virtual crack extension da, which is equivalent to the fracture energy release for linear elastic materials [9]

$$J = -\frac{\vartheta \Pi}{\vartheta a} = G \tag{3.30}$$

However that equivalence only holds under the following assumptions:

- The body is subjected to monotonic loading, that is no unloading takes place
- There is no body force $\mathbf{b} = 0$
- The crack interface is traction-free

- Small deformations and isothermal conditions

In addition to the elastic energy release, the path-independence of the J-integral allows the evaluation of the nonlinear elastic energy release rate and the elastoplastic work far from the crack tip. Furthermore, the contour J-integral has been extended to dynamic loading, cohesive cracks and the presence of body forces. In [12] Atluri develops a modified J-integral formulation for the presence of body force $\tilde{\mathbf{b}}$ (expressed in local cartesian system)

$$J = \int_{\Gamma} \left(w\,dy - \mathbf{t}\frac{\vartheta \tilde{\mathbf{u}}}{\vartheta \tilde{x}} d\Gamma \right) - \int_{\Omega} \tilde{\mathbf{b}} \frac{\vartheta \tilde{\mathbf{u}}}{\vartheta \tilde{x}} d\Omega \tag{3.31}$$

3.3.2 Interaction integrals

In 2D mixed-mode crack problems there are two SIFs, thus calculating them from the J-integral requires some extra work. The interaction integral method provides a solution to this problem. Consider two states of the cracked body:

- State (1) is the real state of the body. Its displacement, strain and stress fields will be denoted as $\tilde{u}_i^{(1)}$, $\tilde{\varepsilon}_{ij}^{(1)}$ and $\tilde{\sigma}_{ij}^{(1)}$ in the following. Note that this state corresponds to the *current* configuration of the system, meaning it will change every time the crack propagates.

- State (2) is an auxiliary state, whose fields will be denoted as $\tilde{u}_i^{(2)}$, $\tilde{\varepsilon}_{ij}^{(2)}$ and $\tilde{\sigma}_{ij}^{(2)}$ in the following. Since they are virtual states, they can be selected appropriately to find a relationship between the mixed-mode SIFs and the interaction integrals. However there is a constraint: their fields must satisfy both the equilibrium equation and the traction-free boundary condition on the crack surface.

By applying (3.27) for the superposition of these two states, the J-integral is

$$J^{(1+2)} = \int_{\Gamma} \left[\frac{1}{2} \left(\tilde{\sigma}_{ij}^{(1)} + \tilde{\sigma}_{ij}^{(2)} \right) \left(\tilde{\varepsilon}_{ij}^{(1)} + \tilde{\varepsilon}_{ij}^{(2)} \right) \delta_{1j} - \left(\tilde{\sigma}_{ij}^{(1)} + \tilde{\sigma}_{ij}^{(2)} \right) \frac{\vartheta}{\vartheta \tilde{x}} \left(\tilde{u}_i^{(1)} + \tilde{u}_i^{(2)} \right) \right] \tilde{n}_j d\Gamma \tag{3.32}$$

Expanding the previous expression we get

$$J^{(1+2)} = J^{(1)} + J^{(2)} + I^{(1,2)} \tag{3.33}$$

where $J^{(1)}$, $J^{(2)}$ are the J-integrals for the real and auxiliary state respectively, while $I^{(1,2)}$ is called the interaction integral of the two states and is defined as

$$I^{(1,2)} = \int_{\Gamma} \left(w^{(1,2)} \delta_{1j} - \tilde{\sigma}_{ij}^{(1)} \frac{\vartheta \tilde{u}_i^{(2)}}{\vartheta \tilde{x}} - \tilde{\sigma}_{ij}^{(2)} \frac{\vartheta \tilde{u}_i^{(1)}}{\vartheta \tilde{x}} \right) \tilde{n}_j d\Gamma \tag{3.34}$$

where $w^{(1,2)}$ is the interaction strain energy defined as

$$w^{(1,2)} = \frac{1}{2} \left(\tilde{\sigma}_{ij}^{(1)} \tilde{\varepsilon}_{ij}^{(2)} + \tilde{\sigma}_{ij}^{(2)} \tilde{\varepsilon}_{ij}^{(1)} \right) \tag{3.35}$$

For linear elastic materials

$$\tilde{\boldsymbol{\varepsilon}}^{(1)} \cdot \tilde{\boldsymbol{\sigma}}^{(2)} = \tilde{\boldsymbol{\varepsilon}}^{(1)} \cdot \mathbf{C} \cdot \tilde{\boldsymbol{\varepsilon}}^{(2)} = \tilde{\boldsymbol{\varepsilon}}^{(2)} \cdot \mathbf{C} \cdot \tilde{\boldsymbol{\varepsilon}}^{(1)} = \tilde{\boldsymbol{\varepsilon}}^{(2)} \cdot \tilde{\boldsymbol{\sigma}}^{(1)} \tag{3.36}$$

and (3.35) can be written as

$$w^{(1,2)} = \tilde{\sigma}_{ij}^{(1)} \tilde{\varepsilon}_{ij}^{(2)} = \tilde{\sigma}_{ij}^{(2)} \tilde{\varepsilon}_{ij}^{(1)} \tag{3.37}$$

For 2D mixed-mode problems the energy release rate is linked to the stress intensity factors K_{I} and K_{II} according to

$$J = G = \frac{K_{\mathrm{I}}^2 + K_{\mathrm{II}}^2}{E^*} \tag{3.38}$$

where E^* is the effective Young's modulus

$$E^* = \begin{cases} \dfrac{E}{1 - \nu^2} & \text{for plane strain problems} \\ E & \text{for plane stress problems} \end{cases} \tag{3.39}$$

The J-integral of the superposition of the two states can also be written as [9]

$$J^{(1+2)} = J^{(1)} + J^{(2)} + \frac{2}{E^*} \left(K_{\mathrm{I}}^{(1)} K_{\mathrm{I}}^{(2)} + K_{\mathrm{II}}^{(1)} K_{\mathrm{II}}^{(2)} \right) \tag{3.40}$$

Comparing (3.33) and (3.40) the interaction integral can be expressed as

$$I^{(1,2)} = \frac{2}{E^*} \left(K_{\mathrm{I}}^{(1)} K_{\mathrm{I}}^{(2)} + K_{\mathrm{II}}^{(1)} K_{\mathrm{II}}^{(2)} \right) \tag{3.41}$$

It was mentioned previously that we can arbitrarily choose the auxiliary states. Fields that result in $K_{\mathrm{I}}^{(2)} = 0$ XOR $K_{\mathrm{II}}^{(2)} = 0$ are good candidates, since the remaining non-zero SIF can then be directly obtained by the interaction integral using (3.41):

1. First assume that auxiliary state (2) is in pure Mode I. This leads to

$$K_{\mathrm{I}}^{(2)} = 1 \tag{3.42a}$$

$$K_{\mathrm{II}}^{(2)} = 0 \tag{3.42b}$$

   and the real Mode I SIF can be obtained by the interaction integral $I^{(1,\,\mathrm{I})}$

$$K_{\mathrm{I}}^{(1)} = \frac{E^*}{2} I^{(1,\,\mathrm{I})} \tag{3.43}$$

2. Then assume that auxiliary state (2) is in pure Mode II. This leads to

$$K_{\mathrm{I}}^{(2)} = 0 \tag{3.44a}$$

$$K_{\mathrm{II}}^{(2)} = 1 \tag{3.44b}$$

   and the real Mode II SIF can be obtained by the interaction integral $I^{(1,\,\mathrm{II})}$

$$K_{\mathrm{II}}^{(1)} = \frac{E^*}{2} I^{(1,\,\mathrm{II})} \tag{3.45}$$

It should be noted that evaluation of the interaction integral requires careful attention as the real fields (state 1) are usually obtained from the finite element solution in the global cartesian or isoparametric coordinate system, while the auxiliary fields (state 2) are defined in the local crack tip polar system. Therefore, necessary transformations are required to use a unified coordinate system.

3.3.3 Equivalent Domain Integral

The countour J-integral is not optimal for calculations over a finite element mesh. Although the contour can be selected to pass through the finite elements' integration points, in practice this technique rarely exhibits path independence and the result becomes mesh dependent. A more accurate approach, which is also more suitable for FE methods, is the Equivalent Domain Integral (EDI) presented in [13]. The EDI method transforms the contour integral into an area integral using the divergence theorem. Consider the crack tip and the contour integrals shown in Figure 3.6, where

- $\tilde{x}$ and $\tilde{y}$ are the axes of the local cartesian system defined at the crack tip.
- Γ_1 and Γ_3 are two contours defined *around* the crack tip.

- **n** are unit normal vectors orientated outwards.
- A_Γ is the area between the two contours.

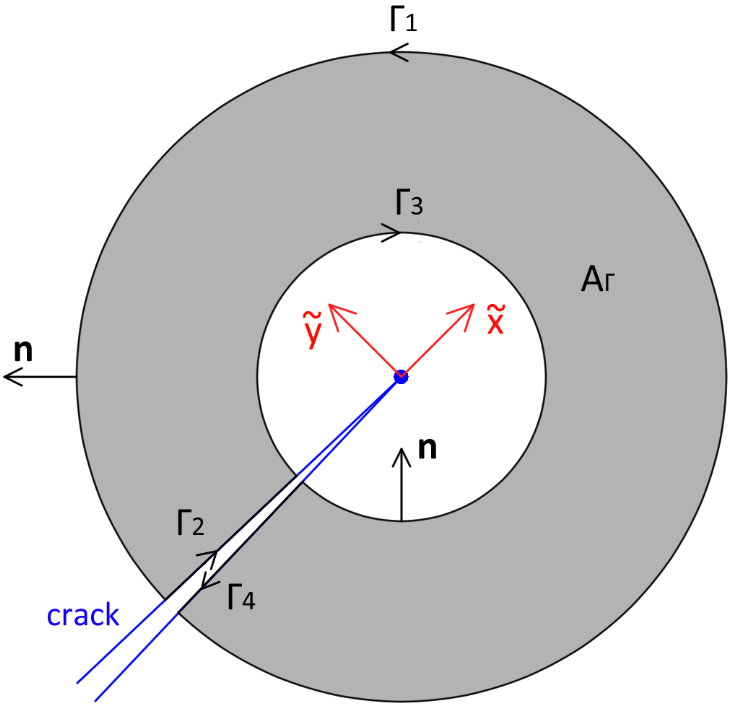


Fig. 3.6 J-integral contours around a crack tip, their normal vectors and the area between them.

By applying the divergence theorem to (3.27), the area form of the J-integral is obtained as

$$J = \int_{\mathrm{A}_\Gamma} \left(\tilde{\sigma}_{ij} \frac{\vartheta \tilde{u}_i}{\vartheta \tilde{x}} - w \, \delta_{1j} \right) \frac{\vartheta q}{\vartheta \tilde{x}_j} dA \tag{3.46}$$

where $q(\mathbf{x})$ is a sufficiently smooth scalar field that is unity on Γ_3 and vanishes on Γ_1. As stated in Section 3.3.2, the interaction integrals are the ones actually used for the calculation of the SIFs. Their area form is

$$I^{(1,2)} = \int_{\mathrm{A}_\Gamma} \left(\tilde{\sigma}_{ij}^{(1)} \frac{\vartheta \tilde{u}_i^{(2)}}{\vartheta \tilde{x}} + \tilde{\sigma}_{ij}^{(2)} \frac{\vartheta \tilde{u}_i^{(1)}}{\vartheta \tilde{x}} - w^{(1,2)} \, \delta_{1j} \right) \frac{\vartheta q}{\vartheta \tilde{x}_j} dA \tag{3.47}$$

3.4 Crack propagation criteria

At each time step of a crack propagation analysis, after evaluating the stress intensity factors, it must be determined if the crack propagates, in which direction it propagates and what is the propagation length. After these requirements are met, the analysis can progress to the next iteration or terminate, if necessary.

3.4.1 Failure criteria

The crack propagation analysis stops if one of the following criteria is met:

- The crack tip has reached the boundary of the domain, causing it to snap into two pieces. In this case the body can no longer bear loads without deforming infinitely and the new stiffness matrix would be singular. Typically, the crack propagation direction and length are estimated (see Sections 3.4.2 and 3.4.3) and then it is determined whether the crack extension exceeds the domain's boundaries.

- The strain energy release rate has exceeded a critical value, leading to failure of the material. After this point, the crack would grow uncontrollably (”spontaneously”) without requiring additional work to be input into the system.

For the latter criterion consider the ideal case of an infinite plate in uniaxial stress discussed in Section 3.2.2. Recall that failure occurs when the actual stress intensity factor K exceeds the critical value defined in (3.16). In mixed-mode loading conditions the crack propagates as long as it is supplied by external work and there is no failure, which is represented by a combination of the mixed-mode SIFS being lower than a critical value. During a crack propagation analysis, the crack will continue to grow as long as

$$K_{\text{eff}} < K_{\text{Ic}} \tag{3.48}$$

where

- K_{Ic} is a material property called fracture toughness. The linear elastic fracture toughness of a material is usually measured as the Mode I critical stress intensity factor, hence the notation K_{Ic}. Its units are $\mathbf{Pa}\sqrt{\mathbf{m}}$.

- K_{eff} is a characteristic measure that incorporates all stress intensity factors. For 2D problems the following expressions can be used

$$K_{\text{eff}} = \sqrt{K_{\text{I}}^2 + K_{\text{II}}^2} \tag{3.49a}$$

$$K_{\text{eff}} = \left(K_{\text{I}}^4 + 8K_{\text{II}}^4\right)^{1/4} \qquad \text{(see [11])} \tag{3.49b}$$

$$K_{\text{eff}} = K_{\text{I}} \cos^3\frac{\theta_c}{2} - \frac{3}{2}K_{\text{II}}\cos\frac{\theta_c}{2}\sin\theta_c \qquad \text{(see [31])} \tag{3.49c}$$

where θ_c is the crack propagation angle, which will be described in the next section.

3.4.2 Crack propagation direction

This section describes alternative theories to predict the crack propagation angle θ_c, which is defined as the angle from the line segment containing the current crack tip to a new line segment that will contain the new crack tip, as illustrated in Figure 3.7.

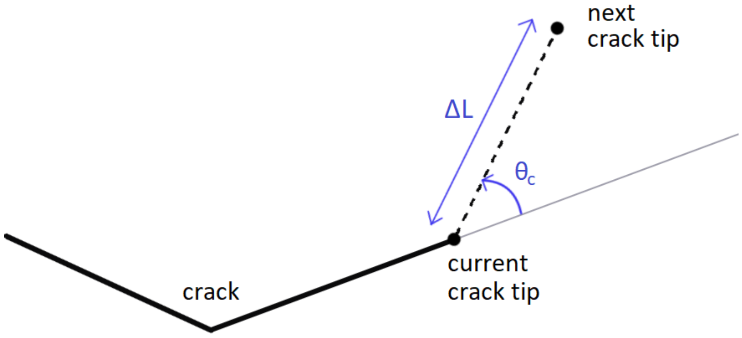


Fig. 3.7 Graphical representation of the crack propagation angle θ_c and crack propagation length $\Delta\alpha$.

Maximum circumferential tensile stress

The maximum circumferential (or hoop) stress criterion was first developed by Erdogan and Sih [7] based on the stress state near a crack tip. According to it, the crack propagates in a radial direction perpendicular to the direction of maximum tension, that is the propagation direction is perpendicular to the plane for which the circumferential stress $\sigma_{\theta\theta}$ is maximum. It can be identified as the plane where the circumferential stress $\sigma_{\theta\theta}$ is principal and the shear stress stress vanishes $\sigma_{r\theta} = 0$, with these two stress components being expressed in the local polar coordinate system defined at the crack tip. In the case of brittle materials, the plastic zone around the crack tip is negligible, which permits the use of the asymptotic stress expressions presented in Section 3.2.4. However note that the following procedure cannot be used for ductile materials, where the plastic zone is significant.

For mixed-mode loading, the circumferential and shear stress can be obtained by transforming (3.19) and (3.21) to the local polar coordinate system defined at the crack tip

$$\sigma_{\theta\theta} = \frac{K_{\mathrm{I}}}{4\sqrt{2\pi r}}\left(3\cos\frac{\theta}{2} + \cos\frac{3\theta}{2}\right) + \frac{K_{\mathrm{II}}}{4\sqrt{2\pi r}}\left(-3\sin\frac{\theta}{2} - 3\sin\frac{3\theta}{2}\right) \tag{3.50a}$$

$$\sigma_{r\theta} = \frac{K_{\mathrm{I}}}{4\sqrt{2\pi r}}\left(\sin\frac{\theta}{2} + \sin\frac{3\theta}{2}\right) + \frac{K_{\mathrm{II}}}{4\sqrt{2\pi r}}\left(\cos\frac{\theta}{2} + 3\cos\frac{3\theta}{2}\right) \tag{3.50b}$$

Setting the shear stress $\sigma_{r\theta} = 0$ results in

$$\frac{1}{2\pi r}\cos\frac{\theta}{2}\left[\frac{1}{2}K_{\mathrm{I}}\sin\theta + \frac{1}{2}K_{\mathrm{II}}\left(3\cos\theta - 1\right)\right] = 0 \tag{3.51}$$

The above can be solved to obtain the crack propagation angle θ_c

$$\theta_c = \begin{cases} 2\cdot\operatorname{atan}\frac{1}{4}\left[\frac{K_{\mathrm{I}}}{K_{\mathrm{II}}} - \operatorname{sign}(K_{\mathrm{II}})\sqrt{\left(\frac{K_{\mathrm{I}}}{K_{\mathrm{II}}}\right)^2 + 8}\right] & K_{\mathrm{II}} \neq 0 \\ 0 & K_{\mathrm{II}} = 0 \end{cases} \tag{3.52}$$

which means that

- $\theta_c < 0$ if $K_{\mathrm{II}} > 0$.
- $\theta_c > 0$ if $K_{\mathrm{II}} < 0$.
- $\theta_c = 0$ if $K_{\mathrm{II}} = 0$, that is pure Mode I conditions.

An efficient expression of the crack propagation angle is presented in [27]

$$\theta_c = 2\cdot\operatorname{atan}\frac{-2K_{\mathrm{II}}/K_{\mathrm{I}}}{1 + \sqrt{1 + 8(K_{\mathrm{II}}/K_{\mathrm{I}})^2}} \tag{3.53}$$

Note that θ_c is defined as a counter-clockwise angle from the tangent line at the crack tip, that is from the local $\tilde{x}$ axis, to the direction of the crack growth. Applying the previous equations, results in $\theta_c \in (-70.5288^o, 70.5288^o)$ as shown in Figure 3.8.

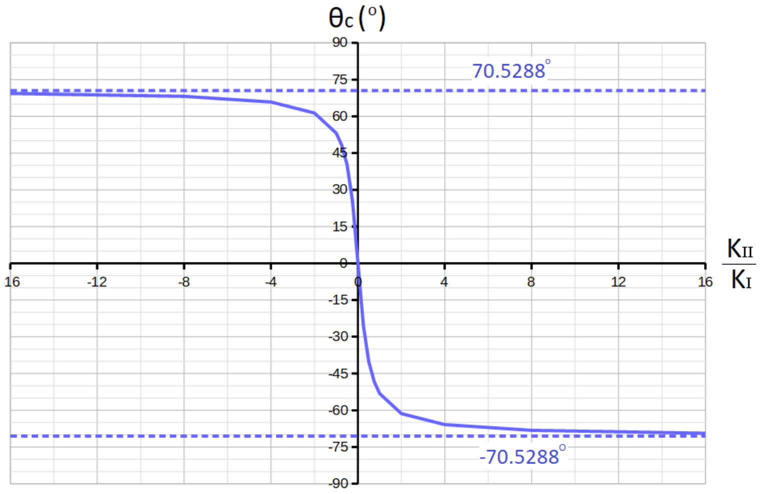


Fig. 3.8 The range of values of the crack propagation angle θ_c, obtained by the maximum circumferential stress criterion.

Minimum strain energy

The minimum strain energy density criterion was first proposed by Sih [10]. According to this criterion, the crack propagates in the direction where the material has its minimum strength. The crack propagates from the tip in a direction θ_c, along which the strain energy density at a critical distance is a minimum. The strain energy density function is defined as

$$\frac{dW}{dA} = \frac{1}{2}\left[\sigma_{rr}\frac{\vartheta u_r}{\vartheta r} + \sigma_{\theta\theta}\left(\frac{u_r}{r} + \frac{1}{r}\frac{\vartheta u_\theta}{\vartheta\theta}\right) + \sigma_{r\theta}\left(\frac{1}{r}\frac{\vartheta u_r}{\vartheta\theta} + \frac{\vartheta u_\theta}{\vartheta r} - \frac{u_\theta}{r}\right)\right] \tag{3.54}$$

where $dA = r\,dr\,d\theta$. By substituting the polar expressions of the asymptotic displacement and stress fields around the crack tip and after considerable manipulations, the strain energy density can be expressed in a quadratic form:

$$\frac{dW}{dA} = \frac{1}{r}\left(a_{11}K_{\mathrm{I}}^2 + 2a_{12}K_{\mathrm{I}}K_{\mathrm{II}} + a_{22}K_{\mathrm{II}}^2\right) \tag{3.55}$$

where the coefficients are functions of θ:

$$a_{11} = \frac{1}{16\mu}(1+\cos\theta)(\kappa-\cos\theta) \tag{3.56a}$$

$$a_{12} = \frac{1}{16\mu}\sin\theta\,(2\cos\theta-\kappa+1) \tag{3.56b}$$

$$a_{22} = \frac{1}{16\mu}[(\kappa+1)(1-\cos\theta)+(1+\cos\theta)(3\cos\theta-1)] \tag{3.56c}$$

Note that (3.55) ceases to be valid if r becomes indefinitely small and a critical value r_o that designates a core region is defined. In any case, the magnitude of the the strain density energy is denoted by S and called the *stress energy density factor*:

$$S = a_{11}K_{\text{I}}^2 + 2a_{12}K_{\text{I}}K_{\text{II}} + a_{22}K_{\text{II}}^2 \tag{3.57}$$

This factor depends on θ through the coefficients a_{ij} and therefore gives a description of the local energy density on any radial plane intersecting the crack tip. The crack propagation angle θ_c can be obtained by minimizing S with respect to θ. As in Section 3.4.2, θ_c is the counter-clockwise angle from the tangent line at the crack tip to the direction of the crack growth.

3.4.3 Crack propagation length

This section is concerned with the selection of the crack propagation length ΔL. Obtaining both θ_c and ΔL is necessary for the determination of the crack extension.

Constant propagation length

The simplest approach is to use a constant value as the crack propagation length in each iteration. In [20] the crack growth length was chosen as *ΔL = 0.5 * initial crack length* or *ΔL = 1 * initial crack length*. Some constraints to take into consideration when selecting the propagation length are:

- The crack propagation length must be greater than the mesh size:

  If the propagation length is smaller than the dimensions of the element containing the crack tip, it is possible that the new crack tip will also lie within the same element. For a piece-wise linear representation of the crack (successive line segments), this means that during the next iteration there will be an element containing the crack tip and a kink point. After two iterations that element will contain two kink points.

  Although there is no problem in theory or in the XFEM formulation, these configurations pose a significant challenge for the accurate representation of the crack's geometry. For explicit crack geometry representations (see Section 5.2), more complicated computational geometry algorithms must be employed, which are inevitably less efficient. For implicit crack geometry representations (see Section 5.3), like the Level Set

Method, the accuracy of the representation and thus of the whole analysis is reduced.

Therefore it is recommended that

$$\Delta L > \text{max element dimension} \tag{3.58}$$

- The crack propagation length must be greater than the J-integral domain radius r_J:

  Since J-integral is path independent only for straight cracks [23], the presence of non-collinear crack segments tends to deteriorate the accuracy in computing the stress intensity factors. Therefore it is recommended that

  $$\Delta L > r_J \tag{3.59}$$

- The crack propagation length should be as small as possible:

  As long as the previous two constraints are satisfied, smaller propagation lengths mean that the piece-wise linear approximation of the crack's path is more accurate. To improve the accuracy in general, use a finer mesh, shorter crack growth increments and also reduce the J-integral domain radius to match the new mesh size and propagation length.

Paris law

In fatigue crack problems the crack propagation length is determined by the Paris' law, which was introduced in [8]. Paris' law gives the advancement dL of fatigue crack per loading cycle dN as a function of the amplitude of stress intensity factor $\Delta K = K_{max} - K_{min}$:

$$\frac{dL}{dN} = C \cdot \Delta K^m \qquad \text{for} \;\; \Delta K_{th} < \Delta K < K_{Ic} \tag{3.60}$$

where

- C, m are material properties called Paris constants.
- K_{Ic} is the fracture toughness as defined in Section 3.4.1.
- ΔK_{th} is called fatigue threshold and signifies the value below which the crack does not propagate.

Figure 3.9 illustrates the development of a fatigue crack in typical alloys:

1. Region I begins at ΔK_{th} and ends when the slope of the curve $\frac{dL}{dN}$ becomes linear. In this region crack propagation is difficult to predict since it depends on microstructure and flow properties of the material.

2. In region II ($\Delta K_{th} < \Delta K < K_{Ic}$) the fatigue crack growth is governed by Paris' law. If region II includes the dominating part of the fatigue life, which is reasonable for most engineering structures, then the crack propagation length can be estimated using Paris' law.

3. In region III the fracture toughness K_{Ic} is exceeded, the crack growth rate accelerates and finally failure occurs.

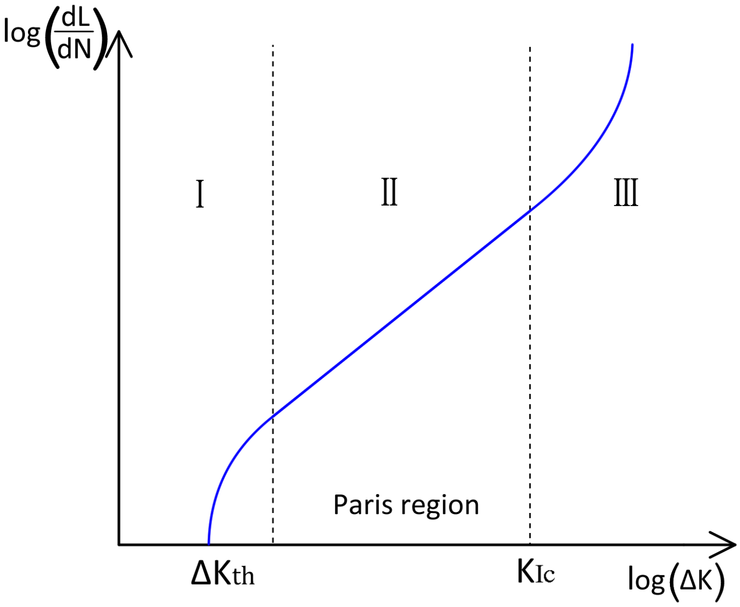


Fig. 3.9 Fatigue crack growth: the Paris law applies to region B.

By applying Paris' law for mixed-mode problems, the crack propagation length after N cycles is obtained by

$$\Delta L = N \cdot C \cdot \Delta K_{\text{eff}}^m \tag{3.61}$$

where ΔK_{eff} can be defined similarly to (3.49) as

$$\Delta K_{\mathrm{eff}} = \sqrt{\Delta K_{\mathrm{I}}^2 + \Delta K_{\mathrm{II}}^2} \tag{3.62a}$$

$$\Delta K_{\mathrm{eff}} = \left(\Delta K_{\mathrm{I}}^4 + 8\Delta K_{\mathrm{II}}^4\right)^{1/4} \quad \text{(see [11])} \tag{3.62b}$$

$$\Delta K_{\mathrm{eff}} = \Delta K_{\mathrm{I}} \cos^3 \frac{\theta_c}{2} - \frac{3}{2}\Delta K_{\mathrm{II}} \cos \frac{\theta_c}{2} \sin\theta_c \quad \text{(see [31])} \tag{3.62c}$$

3.5 Implementation of crack propagation in XFEM

This section contains details about the implementation of the crack propagation phase of the analysis. The main objective is to calculate the stress intensity factors. Once they are obtained, the crack propagates according to the criteria presented in Section 3.4.

3.5.1 Numerical form of the interaction integrals

To obtain the stress intensity factors one has to first compute the interaction integrals and then apply (3.43) and (3.45). The Equivalent Domain Integral method (see Section 3.3.3) is more suited to integrating within a FE method framework. Based on it, the interaction integrals can be written as

$$\begin{aligned} I^{(1,2)} &= \int\limits_{\mathrm{A}_\Gamma} \left(\tilde{\sigma}_{ij}^{(1)} \frac{\vartheta \tilde{u}_i^{(2)}}{\vartheta \tilde{x}} + \tilde{\sigma}_{ij}^{(2)} \frac{\vartheta \tilde{u}_i^{(1)}}{\vartheta \tilde{x}} - w^{(1,2)}\,\delta_{1j} \right) \frac{\vartheta q}{\vartheta \tilde{x}_j}\, dA \\ &= \sum_{e \in M_{el}} \left[\int\limits_{\tilde{\Omega}_e} \left(\tilde{\sigma}_{ij}^{(1)} \frac{\vartheta \tilde{u}_i^{(2)}}{\vartheta \tilde{x}} + \tilde{\sigma}_{ij}^{(2)} \frac{\vartheta \tilde{u}_i^{(1)}}{\vartheta \tilde{x}} - w^{(1,2)}\,\delta_{1j} \right) \frac{\vartheta q}{\vartheta \tilde{x}_j}\, d\tilde{x}\, d\tilde{y} \right] \\ &= \sum_{e \in M_{el}} I_e^{(1,2)} \end{aligned} \tag{3.63}$$

where

- M_{el} is the set of elements comprising the integration domain A_Γ. Details about the determination of the integration domain are given in Section 3.5.2.

- $I_e^{(1,2)}$ is the contribution of element e to the global interaction integral $I^{(1,2)}$:

$$I_e^{(1,2)} = \int\limits_{\tilde{\Omega}_e} f_J(\tilde{\mathbf{x}})\, d\tilde{x} d\tilde{y} \tag{3.64}$$

- $f_J(\tilde{\mathbf{x}})$ is used for convenience in place of the integrand, which in this case is a scalar field.

The integration subdomains $\tilde{\Omega}_e$ are defined in the the crack tip's local cartesian coordinate system. However, the integrals can only be computed in the natural coordinate system of each isoparametric element. Therefore, the integration domains need to be transformed from the tip's local cartesian to the global cartesian and then to each element's natural coordinate system. These transformations are done by applying the "change of variables" technique:

$$\begin{aligned} I_e^{(1,2)} &= \int\limits_{\tilde{\Omega}_e} f_J(\tilde{\mathbf{x}}) d\tilde{x} d\tilde{y} = \int\limits_{\Omega_e} f_J(\mathbf{x}) |\mathbf{J}_{GL}| dx dy \\ \Leftrightarrow I_e^{(1,2)} &= \int\limits_{-1}^{1} \int\limits_{-1}^{1} f_J(\boldsymbol{\xi})\ |\mathbf{J}_{GL}|\ |\mathbf{J}_{NG}(\boldsymbol{\xi})|\, d\xi\, d\eta \end{aligned} \tag{3.65}$$

where

- $|\mathbf{J}_{NG}(\boldsymbol{\xi})|$ is the determinant of the mapping from each element's natural system to the global cartesian system. It is also a (non-constant) function of the natural coordinates and its calculation is presented in Appendix A.1.2.

- $|\mathbf{J}_{GL}|$ is the determinant of the mapping from the global cartesian to the crack tip's local cartesian system. In Appendix A.2.2 it is shown that

$$|\mathbf{J}_{GL}| = |\mathbf{Q}| = \begin{vmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{vmatrix} = 1 \tag{3.66}$$

Once transformed to the natural coordinate system, the interaction integrals are computed numerically:

$$\begin{aligned} I^{(1,2)} &= \int_{-1}^{1}\int_{-1}^{1} f_J(\boldsymbol{\xi}) \cdot 1 \cdot |\mathbf{J}_{NG}(\boldsymbol{\xi})| \, d\xi \, d\eta \\ &= \sum_{k \in M_{GP}} f_J\left(\boldsymbol{\xi}_k\right) \left|\mathbf{J}_{NG}\left(\boldsymbol{\xi}_k\right)\right| w_k \end{aligned} \tag{3.67}$$

where M_{GP} is the set of integration points defined by their natural coordinates $\boldsymbol{\xi}_k$ and their weight w_k. For details on selecting the integration points in the context of XFEM refer to Chapter 4.

The interaction integrals for the whole domain can now be written as

$$I^{(1,2)} = \sum_{e \in M_{el}} \left\{ \sum_{k \in M_{GP}} \left[\left(\tilde{\sigma}_{ij}^{(1)} \frac{\vartheta \tilde{u}_i^{(2)}}{\vartheta \tilde{x}} + \tilde{\sigma}_{ij}^{(2)} \frac{\vartheta \tilde{u}_i^{(1)}}{\vartheta \tilde{x}} - w^{(1,2)} \delta_{1j} \right) \frac{\vartheta q}{\vartheta \tilde{x}_j} |\mathbf{J}_{NG}| \, w_k \right] \right\} \tag{3.68}$$

Remarks:

- Before using (3.68), 3 fields must be computed for the real state (1), the auxiliary states (2) = (I) and the auxilliary state (2) = (II): displacement gradient, strain tensor and stress tensor. Furthermore, the gradient of weighting function $q(\mathbf{x})$, which is a scalar field, must be calculated too.

- These fields are represented in different coordinate systems and they must be converted to the crack tip's local cartesian system, as required by (3.68). The global cartesian system could also be used, but it would require different equations than the ones presented in this and the following sections.

- First of all, the natural coordinates of each integration point $\boldsymbol{\xi}_k$ must be transformed to the other coordinate systems, since many of the involved analytic formulas are expressed there:

  1. Natural to global cartesian, using (A.1)
  2. Global cartesian to local cartesian, using (A.21)
  3. Local cartesian to local polar, using (A.27)

- The integration domain is converted to the natural system, while the fields of interest in (3.68) are converted to the local cartesian system.
- (3.68) must be applied for both auxiliary states thus computing two interaction integrals $I^{(1,\mathrm{I})}$ and $I^{(1,\mathrm{II})}$.

3.5.2 Determining the J-integral domain

Recall from Section 3.3.3 that the interaction integrals used for calculating the SIFs are defined in the area between an inner and outer contour, which are illustrated in Figure 3.6. In practice the inner contour Γ_3 is chosen to degenerate into the crack tip. The outer contour Γ_1 is selected to be the outer boundary of the elements intersected by a circle with radius r_J:

$$r_J = k \cdot h_{local} \tag{3.69}$$

where

- h_{local} is the mesh size in the vicinity of the crack tip. It can be estimated as the characteristic length of the element containing the crack tip

$$h_{local} = \sqrt{A_{\text{tip element}}} \tag{3.70}$$

- k is a magnification factor. Moes et al. [20] proposes $k = 2$.

Nevertheless, it is important that the J-integral domain radius r_J does not exceed the crack propagation length Δa, since the J-integral is path independent only for straight cracks [23]. The crack geometry is usually modeled by a series of line segments. If $r_J > \Delta L$, the integration domain will include more segments than just the one containing the crack tip and they will not be collinear. This will introduce errors into the computation the stress intensity factors. Hence, it is recommended in [28] to bound the J-integral domain radius in the following interval, so that the integrals are path and mesh independent:

$$1.5 h_{local} < r_J < \Delta L \tag{3.71}$$

According to what is stated above, the interaction integrals of (3.47) need to be calculated over all elements that are inside the circle with center the crack tip and radius r_J or intersected by it. However by appropriately selecting the weighting function $q(\mathbf{x})$, a more efficient domain can be selected. As will be described in Section 3.5.3, $\dfrac{\vartheta q}{\vartheta \tilde{x}_j} = 0 \Rightarrow I^{(1,2)} = 0$ in the subdomain Ω_e of each element that lies completely inside the circle. Therefore these elements can be excluded and the J-integral domain radius is composed of all elements that are intersected by the circle C(tip, r_J).

3.5.3 Weight function

As stated in Section 3.3.3 the weighting function $q(\mathbf{x})$ is a smooth function that has unity value at the inner contour and zero value at the outer contour. A convenient function is depicted in Figure 3.10 and defined as

- $q(\mathbf{x}) = 1$ for nodes inside the circle C(crack tip, r_J)
- $q(\mathbf{x}) = 0$ for nodes outside the circle C(crack tip, r_J)
- Inside the elements $q(\mathbf{x})$ is obtained using the isoparametric interpolation, which satisfies the smoothness requirement

$$q(\mathbf{x}(\xi)) = \sum_{i=1}^{n_{\text{nodes}}} N_i(\xi) q_i \tag{3.72}$$

where

- n_{nodes} are the element's nodes used for the isoparametric mapping
- q_i are the nodal weight values described in the previous two bullets: $q_i = 0$ or 1

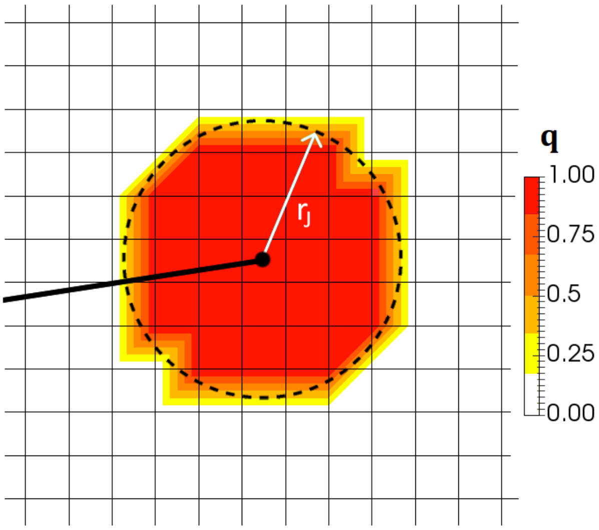


Fig. 3.10 The weighting function q

In order to obtain the necessary derivative $\frac{\vartheta q}{\vartheta \tilde{x}_j}$ at an integration point $\boldsymbol{\xi}_k$ the following calculations are required:

1. Compute the derivatives of the weight function with respect to the natural coordinates of the element

$$_{\xi}\nabla q = \begin{bmatrix} q,_{\xi} & q,_{\eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_{\text{nodes}}} \mathrm{N}_{i},_{\xi}\, q_i & \sum_{i=1}^{n_{\text{nodes}}} \mathrm{N}_{i},_{\eta}\, q_i \end{bmatrix} \tag{3.73}$$

2. Transform them to derivatives with respect to the global cartesian coordinates using the chain rule as described in (A.10)

$$\begin{aligned} {}_{\mathbf{x}}\nabla q &= {}_{\xi}\nabla q \cdot \mathbf{J}_{NG}^{-1} \\ \begin{bmatrix} q,_x & q,_y \end{bmatrix} &= \begin{bmatrix} q,_{\xi} & q,_{\eta} \end{bmatrix} \cdot \begin{bmatrix} \xi,_x & \xi,_y \\ \eta,_x & \eta,_y \end{bmatrix} \end{aligned} \tag{3.74}$$

3. Finally, transform the latter to the local cartesian coordinate system defined at the crack tip using (A.35)

$$\begin{aligned} {}_{\tilde{\mathbf{x}}}\nabla q &= {}_{\mathbf{x}}\nabla q \cdot \mathbf{J}_{GL}^{-1} \\ \begin{bmatrix} q,_{\tilde{x}} & q,_{\tilde{y}} \end{bmatrix} &= \begin{bmatrix} q,_x & q,_y \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \end{aligned} \tag{3.75}$$

   where α is the angle from the global x to the local $\tilde{x}$ axis (see Figure A.3).

Note that for all elements of the integration domain that are completely inside the circle C(crack tip, r_J), (3.72) becomes

$$q = \sum_{i=1}^{n_{\text{nodes}}} N_i \cdot 1 = 1 \tag{3.76}$$

which leads to

$$_{\xi}\nabla q = {}_{\mathbf{x}}\nabla q = {}_{\tilde{\mathbf{x}}}\nabla q = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{3.77}$$

As a result (3.47) reduces to

$$I^{(1,2)} = 0 \tag{3.78}$$

which means that the interaction integrals only need to be calculated for elements intersected by the circle C(crack tip, r_J). Moreover, only the weight values of nodes that are inside the circle need to be stored, or rather references to these nodes themselves since their weights are unity. This can significantly reduce the memory requirements of corresponding algorithms.

3.5.4 Fields of real state

To compute the displacement gradient, strain tensor and stress tensor of the real state, we go through the well known procedure of interpolating the nodal displacements, differentiating the shape functions and using the constitutive matrix. All computations are done using the natural coordinates of each integration point $\boldsymbol{\xi}_k$ directly.

Note that although the aim here is to use these fields in the interaction integral calculations, the same procedure can be used when the fields themselves are of interest, e.g. as output to be examnied by the engineer. Since these quantities are met in more than just state (1) calculations and to reduce verbosity, the (1) superscripts that appear in (3.68) will not be written explicitly here.

Nodal displacements of element

First of all the nodal displacements of each element must be retrieved. The J-integral computations typically occur right after solving the linear system of the current real state. Thus the nodal values at the global standard and artificial degrees of freedom are available as a unified vector. In this section we will follow the same dof numbering as in Section 2.3.1, meaning all artificial dofs are placed after all standard dofs. We will also use the boolean matrices $\mathbf{T}_{std}$ and $\mathbf{T}_{enr}$ defined there, in order to map from the global dofs to the element's standard and artificial dofs respectively. Equations (2.32) and (2.33) are repeated here for convenience:

$$\bar{\mathbf{U}} = \begin{bmatrix} \cdots & u_{Ix} & u_{Iy} & \cdots & a_{Jex} & a_{Jey} & \cdots \end{bmatrix}^T \tag{3.79a}$$

$$\bar{\mathbf{u}} = \mathbf{T}_{std} \cdot \bar{\mathbf{U}} \tag{3.79b}$$

$$\bar{\mathbf{a}} = \mathbf{T}_{enr} \cdot \bar{\mathbf{U}} \tag{3.79c}$$

where

- $\bar{\mathbf{U}}$ are the nodal displacements at the global standard and artificial dofs, obtained by solving the linear system at the current iteration of the analysis.

- $\bar{\mathbf{u}}$ are the nodal displacements at the element's standard dofs.
- $\bar{\mathbf{a}}$ are the nodal displacements at the element's artificial dofs.

Displacement gradient of real state

Before proceeding to the actual computations, some previous equations from Sections 2.3.1 and 2.5 are repeated here for convenience. In XFEM the displacement field inside an isoparametric element is given by (2.27):

$$\begin{aligned}
\mathbf{u}(\xi) &= \sum_{i\in M_{std}} N_i(\xi)\bar{\mathbf{u}}_i + \sum_{e=0}^{4} \sum_{j\in M_e} N_j(\xi)\left[\psi_e(\mathbf{x}(\xi)) - \psi_e(\mathbf{x}_j)\right]\bar{\mathbf{a}}_{je} \\
&= \sum_{i\in M_{std}} N_i(\xi)\bar{\mathbf{u}}_i + \sum_{e=0}^{4} \sum_{j\in M_e} N_{je}^{enr}(\xi)\bar{\mathbf{a}}_{je}
\end{aligned}$$

where the enriched basis function $N_{je}^{enr}(\xi)$ associated with node j and enrichment function e and its derivatives with respect to global cartesian coordinates have been defined in (2.56) and (2.69) respectively as

$$\begin{aligned}
&N_{je}^{enr}(\xi) = N_j(\xi)\left[\psi_e(\mathbf{x}(\xi)) - \psi_e(\mathbf{x}_j)\right] \\
&N_{je}^{enr},_x(\xi) = N_j,_x(\xi)\left[\psi_e(\mathbf{x}(\xi)) - \psi_e(\mathbf{x}_j)\right] + N_j(\xi)\,\psi_e,_x(\mathbf{x}(\xi)) \\
&N_{je}^{enr},_y(\xi) = N_j,_y(\xi)\left[\psi_e(\mathbf{x}(\xi)) - \psi_e(\mathbf{x}_j)\right] + N_j(\xi)\,\psi_e,_y(\mathbf{x}(\xi))
\end{aligned}$$

and

- $e = 0$ corresponds to the Heaviside enrichment and $e = 1...4$ to the asymptotic tip enrichments.
- M_{std} is the set of all nodes of the finite element and $N_i(\mathbf{x})$ are the well known shape functions.
- M_e is the set of nodes enriched with the function $\psi_e(\mathbf{x}(\xi))$ and $N_j(\xi)$ are the corresponding shape functions, which are also Lagrange polynomials but different from the standard shape functions $N_i(\xi)$ in the general case.
- $\bar{\mathbf{u}}_i = \begin{bmatrix} u_{ix} & u_{iy} \end{bmatrix}^T$ are the standard nodal displacements associated with the node j.

- $\bar{\mathbf{a}}_{je} = \begin{bmatrix} a_{jex} & a_{jey} \end{bmatrix}^T$ are the artificial nodal displacements associated with the node j and the enrichment function ψ_e

The displacement field's gradient with respect to global cartesian coordinates can now be computed as

$$u_{x},_{x}(\xi) = \sum_{i \in M_{std}} N_{i},_{x}(\xi)\, \bar{u}_{ix} + \sum_{e=0}^{4} \sum_{j \in M_e} N^{enr}_{je},_{x}(\xi)\, \bar{a}_{jex} \tag{3.81a}$$

$$u_{x},_{y}(\xi) = \sum_{i \in M_{std}} N_{i},_{y}(\xi)\, \bar{u}_{ix} + \sum_{e=0}^{4} \sum_{j \in M_e} N^{enr}_{je},_{y}(\xi)\, \bar{a}_{jex} \tag{3.81b}$$

$$u_{y},_{x}(\xi) = \sum_{i \in M_{std}} N_{i},_{x}(\xi)\, \bar{u}_{iy} + \sum_{e=0}^{4} \sum_{j \in M_e} N^{enr}_{je},_{x}(\xi)\, \bar{a}_{jey} \tag{3.81c}$$

$$u_{y},_{y}(\xi) = \sum_{i \in M_{std}} N_{i},_{y}(\xi)\, \bar{u}_{iy} + \sum_{e=0}^{4} \sum_{j \in M_e} N^{enr}_{je},_{y}(\xi)\, \bar{a}_{jey} \tag{3.81d}$$

where the derivatives of the shape functions $N_i(\xi)$, $N_j(\xi)$ are calculated according to (2.57) and the derivatives of the enrichment functions $\psi_e(\mathbf{x})$ are calculated according to (2.59) and (2.64). Finally the displacement gradient must be converted from the global cartesian to the local cartesian coordinate system defined at the crack tip. This is performed by applying (A.47):

$$\begin{gathered} {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{u}} = \mathbf{Q} \cdot {}_{\mathbf{x}}\nabla\mathbf{u} \cdot \mathbf{J}_{GL}^{-1} \\ \Leftrightarrow \begin{bmatrix} \tilde{u}_{x},_{\tilde{x}} & \tilde{u}_{x},_{\tilde{y}} \\ \tilde{u}_{y},_{\tilde{x}} & \tilde{u}_{y},_{\tilde{y}} \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \cdot \begin{bmatrix} u_{x},_{x} & u_{x},_{y} \\ u_{y},_{x} & u_{y},_{y} \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \end{gathered} \tag{3.82}$$

where α is the counter-clockwise angle from the global cartesian x axis to the local cartesian $\tilde{x}$ axis.

Strain tensor of real state

The global cartesian components of the strain tensor can be computed by simply applying the definition (2.1):

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} u_{x},_{x} \\ u_{y}, y \\ \frac{1}{2}\left(u_{x},_{y} + u_{y},_{x}\right) \end{bmatrix} \tag{3.83}$$

Then they must be converted to the local cartesian system using (A.53)

$$\begin{aligned} \tilde{\varepsilon}_{xx} &= \frac{\varepsilon_{xx} + \varepsilon_{yy}}{2} + \frac{\varepsilon_{xx} - \varepsilon_{yy}}{2} cos2\alpha + \varepsilon_{xy} sin2\alpha \\ \tilde{\varepsilon}_{yy} &= \frac{\varepsilon_{xx} + \varepsilon_{yy}}{2} + \frac{\varepsilon_{xx}(\mathbf{x}) - \varepsilon_{yy}}{2} cos2\alpha - \varepsilon_{xy} sin2\alpha \\ \tilde{\varepsilon}_{xy} &= \varepsilon_{xy} cos2\alpha - \frac{\varepsilon_{xx} - \varepsilon_{yy}}{2} sin2\alpha \end{aligned} \tag{3.84}$$

Alternatively, the local cartesian components of the strain tensor could be directly (no conversions from other systems) obtained by applying (2.1) on the local cartesian displacement gradient, which is more efficient since it would have to be computed anyway

$$\begin{bmatrix} \tilde{\varepsilon}_{xx} \\ \tilde{\varepsilon}_{yy} \\ \tilde{\varepsilon}_{xy} \end{bmatrix} = \begin{bmatrix} \tilde{u}_{x},_{\tilde{x}} \\ \tilde{u}_{y},_{\tilde{y}} \\ \frac{1}{2}\left(\tilde{u}_{x},_{\tilde{y}} + \tilde{u}_{y},_{\tilde{x}}\right) \end{bmatrix} \tag{3.85}$$

Stress tensor of real state

The global cartesian components of the stress tensor can be computed by using the constitutive law (2.2):

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E^*}{1-(\nu^*)^2} \begin{bmatrix} 1 & \nu^* & 0 \\ \nu^* & 1 & 0 \\ 0 & 0 & 1-\nu^* \end{bmatrix} \cdot \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} \tag{3.86}$$

where

- E^* is the effective Young's modulus

$$E^* = \begin{cases} \dfrac{E}{1-\nu^2} & \text{for plane strain problems} \\ E & \text{for plane stress problems} \end{cases} \tag{3.87}$$

- ν^* is the effective Poisson's ratio

$$\nu^* = \begin{cases} \dfrac{\nu}{1-\nu} & \text{for plane strain problems} \\ \nu & \text{for plane stress problems} \end{cases} \tag{3.88}$$

Then they must be converted to the local cartesian system using (A.53)

$$\begin{aligned} \tilde{\sigma}_{xx} &= \frac{\sigma_{xx}+\sigma_{yy}}{2} + \frac{\sigma_{xx}-\sigma_{yy}}{2} cos2\alpha + \sigma_{xy} sin2\alpha \\ \tilde{\sigma}_{yy} &= \frac{\sigma_{xx}+\sigma_{yy}}{2} + \frac{\sigma_{xx}(\mathbf{x})-\sigma_{yy}}{2} cos2\alpha - \sigma_{xy} sin2\alpha \\ \tilde{\sigma}_{xy} &= \sigma_{xy} cos2\alpha - \frac{\varepsilon_{xx}-\sigma_{yy}}{2} sin2\alpha \end{aligned} \tag{3.89}$$

Alternatively, the local cartesian components of the stress tensor could be directly (no conversions from other systems) obtained by applying the constitutive law for the local cartesian strains, which is more efficient since they would have to be computed anyway

$$\begin{bmatrix} \tilde{\sigma}_{xx} \\ \tilde{\sigma}_{yy} \\ \tilde{\sigma}_{xy} \end{bmatrix} = \frac{E^*}{1-(\nu^*)^2} \begin{bmatrix} 1 & \nu^* & 0 \\ \nu^* & 1 & 0 \\ 0 & 0 & 1-\nu^* \end{bmatrix} \cdot \begin{bmatrix} \tilde{\varepsilon}_{xx} \\ \tilde{\varepsilon}_{yy} \\ \tilde{\varepsilon}_{xy} \end{bmatrix} \tag{3.90}$$

3.5.5 Fields of Mode I auxiliary state

The displacement and stress fields are expressed as function of the coordinates of the local polar system defined at the crack tip. Therefore we must first transform the natural coordinates of each integration point $\boldsymbol{\xi}_k$ to the global cartesian (A.1), then to the local cartesian (A.21) and finally to the local polar (A.27) system. Also note that the (I) superscript of the displacement, strain and stress fields presented below is dropped, since the notation would become too heavy.

Displacement gradient of Mode I

By setting $K_{\mathrm{I}}^{(2)} = 1$ (see (3.42)) and after some trigonometric manipulations, the local cartesian components of the displacement field from (3.18) can be written as

$$\tilde{u}_x = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}} \cos\frac{\theta}{2}(\kappa - \cos\theta) = s_o \cdot s_r \cdot s_1^{(\mathrm{I})} \tag{3.91a}$$

$$\tilde{u}_y = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}} \sin\frac{\theta}{2}(\kappa - \cos\theta) = s_o \cdot s_r \cdot s_2^{(\mathrm{I})} \tag{3.91b}$$

where for convenience the following notation has been introduced:

$$s_o = \frac{1}{2\mu\sqrt{2\pi}} = const \tag{3.92a}$$

$$s_r = \sqrt{r} = s_r(r) \tag{3.92b}$$

$$s_1^{(\mathrm{I})} = \cos\frac{\theta}{2}(\kappa - \cos\theta) = s_1^{(\mathrm{I})}(\theta) \tag{3.92c}$$

$$s_2^{(\mathrm{I})} = \sin\frac{\theta}{2}(\kappa - \cos\theta) = s_2^{(\mathrm{I})}(\theta) \tag{3.92d}$$

The relevant derivatives of the above 1D temporary functions are

$$s_r,_r = \frac{1}{2\sqrt{r}} \tag{3.93a}$$

$$s_1^{(\mathrm{I})},_\theta = -\frac{1}{2}\sin\frac{\theta}{2}(\kappa - \cos\theta) + \cos\frac{\theta}{2}\sin\theta \tag{3.93b}$$

$$s_2^{(\mathrm{I})},_\theta = \frac{1}{2}\cos\frac{\theta}{2}(\kappa - \cos\theta) + \sin\frac{\theta}{2}\sin\theta \tag{3.93c}$$

The displacement field's gradient with respect to local polar coordinates can now be calculated as

$${}_{\mathbf{r}}\nabla\tilde{\mathbf{u}} = \begin{bmatrix} \tilde{u}_x,_r & \tilde{u}_x,_\theta \\ \tilde{u}_y,_r & \tilde{u}_y,_\theta \end{bmatrix} = \begin{bmatrix} s_o \cdot s_r,_r \cdot s_1^{(\mathrm{I})} & s_o \cdot s_r \cdot s_1^{(\mathrm{I})},_\theta \\ s_o \cdot s_r,_r \cdot s_2^{(\mathrm{I})} & s_o \cdot s_r \cdot s_2^{(\mathrm{I})},_\theta \end{bmatrix} \tag{3.94}$$

Lastly the displacement field's gradient with respect to local cartesian coordinates can be obtained using (A.48):

$$\begin{gathered} {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{u}} = {}_{\mathbf{r}}\nabla\tilde{\mathbf{u}} \cdot \mathbf{J}_{PL}^{-1} \\ \Leftrightarrow \begin{bmatrix} \tilde{u}_x,_{\tilde{x}} & \tilde{u}_x,_{\tilde{y}} \\ \tilde{u}_y,_{\tilde{x}} & \tilde{u}_y,_{\tilde{y}} \end{bmatrix} = \begin{bmatrix} \tilde{u}_x,_r & \tilde{u}_x,_\theta \\ \tilde{u}_y,_r & \tilde{u}_y,_\theta \end{bmatrix} \cdot \begin{bmatrix} cos\theta & sin\theta \\ \frac{-sin\theta}{r} & \frac{cos\theta}{r} \end{bmatrix} \end{gathered} \tag{3.95}$$

Strain tensor of Mode I

The local cartesian components of the strain tensor can be directly (no conversions from other systems) obtained by simply applying (2.1)

$$\tilde{\varepsilon}_{xx} = \tilde{u}_x,_{\tilde{x}} \tag{3.96a}$$

$$\tilde{\varepsilon}_{yy} = \tilde{u}_y,_{\tilde{y}} \tag{3.96b}$$

$$\tilde{\varepsilon}_{xy} = \frac{1}{2}\left(\tilde{u}_x,_{\tilde{y}} + \tilde{u}_y,_{\tilde{x}}\right) \tag{3.96c}$$

Stress tensor of Mode I

The local cartesian components of the stress tensor can be directly (no conversions from other systems) obtained by using (3.19), after substituting $K_{\mathrm{I}} = 1$

$$\tilde{\sigma}_{xx} = \frac{1}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.97a}$$

$$\tilde{\sigma}_{yy} = \frac{1}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.97b}$$

$$\tilde{\sigma}_{xy} = \frac{1}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{3.97c}$$

3.5.6 Fields of Mode II auxiliary state

The displacement and stress fields are expressed as function of the coordinates of the local polar system defined at the crack tip. Therefore we must first transform the natural coordinates of each integration point $\boldsymbol{\xi}_k$ to the global cartesian (A.1), then to the local cartesian (A.21) and finally to the local polar (A.27) system. Also note that the (II) superscript of the displacement, strain and stress fields presented below is dropped, since the notation would become too heavy.

Displacement gradient of Mode II

By setting $K_{\mathrm{II}}^{(2)} = 1$ (see (3.44)) and after some trigonometric manipulations, the local cartesian components of the displacement field from (3.20) can be written as

$$\tilde{u}_x = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}} \sin\frac{\theta}{2}(2 + \kappa + \cos\theta) = s_o \cdot s_r \cdot s_1^{(\mathrm{II})} \tag{3.98a}$$

$$\tilde{u}_y = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}} \cos\frac{\theta}{2}(2 - \kappa - \cos\theta) = s_o \cdot s_r \cdot s_2^{(\mathrm{II})} \tag{3.98b}$$

where for convenience the following notation has been introduced:

$$s_o = \frac{1}{2\mu\sqrt{2\pi}} = const \tag{3.99a}$$

$$s_r = \sqrt{r} = s_r(r) \tag{3.99b}$$

$$s_1^{(\mathrm{II})} = \sin\frac{\theta}{2}(2 + \kappa + \cos\theta) = s_1^{(\mathrm{II})}(\theta) \tag{3.99c}$$

$$s_2^{(\mathrm{II})} = \cos\frac{\theta}{2}(2 - \kappa - \cos\theta) = s_2^{(\mathrm{II})}(\theta) \tag{3.99d}$$

The relevant derivatives of the above 1D temporary functions are

$$s_{r},_{r} = \frac{1}{2\sqrt{r}} \tag{3.100a}$$

$$s_1^{(\mathrm{II})},_{\theta} = \frac{1}{2}\cos\frac{\theta}{2}(2 + \kappa + \cos\theta) - \sin\frac{\theta}{2}\sin\theta \tag{3.100b}$$

$$s_2^{(\mathrm{II})},_{\theta} = -\frac{1}{2}\sin\frac{\theta}{2}(2 - \kappa - \cos\theta) + \cos\frac{\theta}{2}\sin\theta \tag{3.100c}$$

The displacement field's gradient with respect to local polar coordinates can now be calculated as

$$_{\mathbf{r}}\nabla\tilde{\mathbf{u}} = \begin{bmatrix} \tilde{u}_{x},_{r} & \tilde{u}_{x},_{\theta} \\ \tilde{u}_{y},_{r} & \tilde{u}_{y},_{\theta} \end{bmatrix} = \begin{bmatrix} s_o \cdot s_{r},_{r} \cdot s_1^{(\mathrm{II})} & s_o \cdot s_r \cdot s_1^{(\mathrm{II})},_{\theta} \\ s_o \cdot s_{r},_{r} \cdot s_2^{(\mathrm{II})} & s_o \cdot s_r \cdot s_2^{(\mathrm{II})},_{\theta} \end{bmatrix} \tag{3.101}$$

Lastly the displacement field's gradient with respect to local cartesian coordinates can be obtained using (A.48):

$$\begin{aligned} _{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{u}} &= {}_{\mathbf{r}}\nabla\tilde{\mathbf{u}} \cdot \mathbf{J}_{PL}^{-1} \\ \Leftrightarrow \begin{bmatrix} \tilde{u}_{x},_{\tilde{x}} & \tilde{u}_{x},_{\tilde{y}} \\ \tilde{u}_{y},_{\tilde{x}} & \tilde{u}_{y},_{\tilde{y}} \end{bmatrix} &= \begin{bmatrix} \tilde{u}_{x},_{r} & \tilde{u}_{x},_{\theta} \\ \tilde{u}_{y},_{r} & \tilde{u}_{y},_{\theta} \end{bmatrix} \cdot \begin{bmatrix} cos\theta & sin\theta \\ \frac{-sin\theta}{r} & \frac{cos\theta}{r} \end{bmatrix} \end{aligned} \tag{3.102}$$

Strain tensor of Mode II

The local cartesian components of the strain tensor can be directly (no conversions from other systems) obtained by simply applying (2.1)

$$\tilde{\varepsilon}_{xx} = \tilde{u}_{x},_{\tilde{x}} \tag{3.103a}$$

$$\tilde{\varepsilon}_{yy} = \tilde{u}_{y},_{\tilde{y}} \tag{3.103b}$$

$$\tilde{\varepsilon}_{xy} = \frac{1}{2}\left(\tilde{u}_{x},_{\tilde{y}} + \tilde{u}_{y},_{\tilde{x}}\right) \tag{3.103c}$$

Stress tensor of Mode II

The local cartesian components of the stress tensor can be directly (no conversions from other systems) obtained by using (3.21), after substituting $K_{\mathrm{II}} = 1$

$$\tilde{\sigma}_{xx} = \frac{-1}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\left(2 + \cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right) \tag{3.104a}$$

$$\tilde{\sigma}_{yy} = \frac{1}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{3.104b}$$

$$\tilde{\sigma}_{xy} = \frac{1}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{3.104c}$$

3.5.7 Overview

Algorithm 3.1 summarizes the procedure of finding the crack propagation angle and length, after the global displacements are obtained by solving the linear system at a given iteration

Algorithm 3.1 Crack propagation

```
Input: $\bar{\mathbf{U}}$ are the displacements at the standard and artificial dofs, obtained by solving the linear system.
Determine the radius $r_J$ of the J-integral's outer contour, as described in Section 3.5.2.
Find the set $M_{el}$ of elements that are intersected by the outer contour, as described in Section 3.5.2.
Find all nodes that are inside the outer contour and store them in a set $M_{internal}$. Whenever the nodal weights $q_i$ are required: $q_i = \begin{cases} 1, & i \in M_{internal} \\ 0, & i \notin M_{internal} \end{cases}$
Initialize the interaction integrals: $I^{(1,\mathrm{I})} \leftarrow 0$ and $I^{(1,\mathrm{II})} \leftarrow 0$.
for each element $\in M_{el}$ do
    Find the element's nodal displacements using (3.79).
    Use one of the integration rules described in Chapter 4 to obtain its set of integration points $M_{GP}$.
    for each integration point $\left(\xi_k, \eta_k, w_k\right) \in M_{GP}$ do
        Transform its natural coordinates $\boldsymbol{\xi}_k$ to global cartesian $\mathbf{x}_k$, then local cartesian $\tilde{\mathbf{x}}_k$ and finally local polar $\mathbf{r}_k$ coordinates.
        Calculate the gradient of the weight function $q$ with respect to local cartesian coordinates, as described in Section 3.5.3.
        Calculate the local cartesian displacement gradient, strain tensor and stress tensor of the real state, according to Section 3.5.4
        Calculate the local cartesian displacement gradient, strain tensor and stress tensor of the mode I auxiliary state, according to Section 3.5.5.
        Calculate the local cartesian displacement gradient, strain tensor and stress tensor of the mode II auxiliary state, according to Section 3.5.6.
        Calculate the contributions of this integration point to the interaction integrals using (3.68) and add them to $I^{(1,\mathrm{I})}$ and $I^{(1,\mathrm{II})}$.
Calculate the stress intensity factors $K_{\mathrm{I}}$ and $K_{\mathrm{II}}$ using (3.43) and (3.45).
Let $failure \leftarrow false$ be a boolean value that signals whether failure has occurred or not.
If failure has occurred according to Section 3.4.1, set $failure \leftarrow true$.
Determine the crack propagation angle $\theta_c$ using one of the criteria presented in Section 3.4.2.
Determine the crack propagation length $\Delta L$ using one of the criteria presented in Section 3.4.3: constant propagation length for static crack growth problems or Paris' law for fatigue crack growth problems.
Find the global cartesian position of the new crack tip. If it lies outside the domain's boundaries, set $failure \leftarrow true$.
Output: Return the crack propagation angle $\theta_c$, the crack propagation length $\Delta L$ and the failure flag $failure$.
```

The full crack propagation analysis, from the moment a cracked body is loaded until failure occurs, is presented in Algorithm 3.2. This quasi-static analysis is suitable for bodies under static loading and without any material or geometric non-linearity. It can also be used for fatigue-crack propagation problems. In this case, the amplitude of the actual dynamic loads ΔP must applied to the cracked body and the crack propagation length is calculated using Paris' law.

Algorithm 3.2 Quasi-static analysis

```
    Input: Mesh, material properties, boundary conditions (including loads), initial crack
    geometry, max iterations $n_{iter}$
 1: Initialize the crack geometry
 2: for $iter$ from 1 to $n_{iter}$ do
 3:     Enrich the appropriate nodes with Heaviside and/or asymptotic tip functions.
 4:     Evaluate the enrichment functions at the nodes and store those values in a dictionary
        $nodal\Psi\left\langle node_j, \left[\cdots \quad \psi_e\left(\mathbf{x}_j\right) \quad \cdots\right]\right\rangle$.
 5:     Enumerate standard and artificial dofs.
 6:     Assemble the global stiffness matrix $\mathbf{K}$ and the global force vector $\mathbf{F}$ from the corre-
        sponding stiffness matrices $\mathbf{k}^e$, traction and body force vectors $\mathbf{f}_t^e + \mathbf{f}_b^e$ of each element
        $e$.
 7:     Solve the linear system $\mathbf{K} \cdot \bar{\mathbf{U}} = \mathbf{F}$ to obtain the nodal displacements $\bar{\mathbf{U}}$ at the standards
        and artificial dofs.
 8:     Plot the current crack geometry as well as the displacement, strain and stress fields,
        if they are requested by the user.
 9:     Input $\bar{\mathbf{U}}$ into Algorithm 3.1, in order to obtain the crack propagation angle $\theta_c$, the
        crack propagation length $\Delta L$ and the failure flag $failure$.
10:     if $failure == true$ then
11:         Notify user that failure has occured.
12:         break
13:     else
14:         Update the crack geometry using $\theta_c$ and $\Delta L$
15: Plot the final crack geometry as well as the displacement, strain and stress fields.
```

Remarks:

- The operations described in lines 1 and 14 are performed according to the method chosen for modeling the crack geometry. For more details see Chapter 5.

- In line 5, the order of dofs should be considered alongside the linear system solver,as described in Section 2.5.5.

Chapter 4

Numerical integration in XFEM

4.1 Introduction

In traditional FEM, the integrands that appear in the weak form are created by multiplying and dividing the polynomial shape function derivatives and thus they are polynomials themselves. The Gaussian quadrature technique guarantees the exact integration of polynomials with the minimum number of required integration points, which depends on the polynomial's degree. Since it is also naturally coupled with the isoparametric formulation, it has been established as the de facto integration rule in FEM.

However XFEM uses non-polynomial enriched basis functions, which means that the integrands are not polynomials too. Using Gaussian quadrature introduces a substantial loss in accuracy, even if the number of integration points is increased. As a result, alternative integration rules must be developed, that are consistent with the enrichment functions and the geometry of discontinuities. In crack propagation problems, the following cases must be considered:

- Elements that are intersected by the crack interface. These elements feature discontinuous basis functions due to the Heaviside enrichment. This translates into integrands that are discontinuous over the integration domain Ω_e, but still polynomial in the subdomains above and below the crack interface.
- Elements with nodes that are enriched by the asymptotic tip functions. The derivatives of these functions introduce a $\frac{1}{\sqrt{r}}$ singularity in the integration over the elements' domains. Such elements are:
  - The element containing the crack tip.

  - Elements around the crack tip with all their nodes enriched with the asymptotic tip functions, according to the fixed enrichment area scheme (see Section 2.4.2)
  - Blending elements where some nodes are enriched with the asymptotic tip functions, while others are not.

- Elements of the previous case may also feature a jump (discontinuity) across the crack interface. This discontinuity is modeled by the first asymptotic function $\sqrt{r}\sin\frac{\theta}{2}$. It is present in the crack tip element and in all other elements that have at least one node enriched by the asymptotic tip functions and are intersected by the tangent line at the crack tip (though not after the crack ends).

Note that the above do not concern:

- Elements without any enriched nodes (standard elements).

- Blending elements that have Heaviside enriched nodes, but are not intersected by the crack themselves, in which case the integrands are polynomials. Note that blending elements with at least one node that is enriched with the asymptotic tip functions do not belong in this case.

These elements can use the regular Gaussian quadrature without any accuracy loss. Indeed, applying the more complicated integration rules presented in the next sections over the whole domain, would dramatically increase the computational cost for no reason.

Numerical integration is also used for the computation of the J-integral, or rather the interaction integrals (see Section 3.3.2). The Equivalent Domain Integral is typically used (see Section 3.3.3), which requires the evaluation of the integral in (3.47) over the J-integral domain described in Section 3.5.2. The integrand of (3.47) contains terms that are both discontinuous across the crack interface and singular (1/r singularity again). The former is a concern only for the elements that are actually intersected by the crack, but the latter applies to all elements in the J-integral domain. Even standard elements feature this singularity since it originates from the asymptotic displacement and stress fields assumed in the auxiliary states, as described in Section 3.3.2. Therefore appropriate integration rules must be employed for all elements of the J-integral domain.

4.2 Integration with sub-quadrilaterals

This method was originally proposed by Dolbow in [19]. The element's domain is divided into quadrilateral subdomains and then a conventional Gaussian quadrature is applied for integrating over each of these sub-quads. A different number of sub-quads and a different number of integration points per sub-quad can be used for enriched elements intersected by the crack, tip enriched elements or tip blending elements. Figure 4.1 illustrates this integration rule.

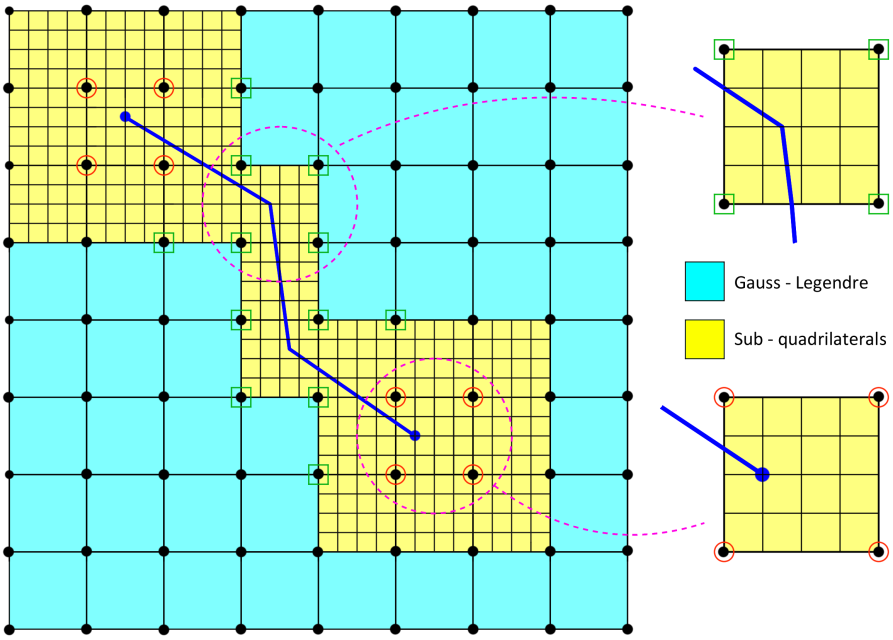


Fig. 4.1 Numerical integration of enriched elements using sub-quadrilaterals

Contrary to Section 4.3, it is neither necessary nor possible to conform the sub-quads to the geometry of the crack. This inescapably results in an inexact integration, but the accuracy can be increased by using more sub-quads and more integration points per sub-quad. In practice, it is observed that by doing so adequate accuracy can be achieved. Consider an element in Figure 4.1 that is intersected by the crack interface, but without nodes enriched with asymptotic tip functions:

- When integrating over a sub-quad that *is not* intersected by the crack, the integrand is polynomial, since the discontinuities of the basis functions lie outside this sub-quad's domain. As a result the integration with conventional Gauss quadratures is exact. There is not even a need to increase the order of the Gauss quadrature, therefore the same integration points as in FEM can be used.

- When integrating over a sub-quad that *is* intersected by the crack, the conventional Gauss quadrature used is unsuitable for the discontinuous integrand. Error is introduced in this sub-quad, as would happen if the Gauss quadrature was applied over the whole element. However the error is now much smaller, since only a few sub-quads are affected, while the rest are integrated exactly. Increasing the number of sub-quads, further increases the accuracy of the integration over the whole element.

For elements with tip enriched nodes, the integrands contain singular terms. Applying the conventional Gauss quadrature for the sub-quads is inexact, even if they are not intersected by the crack interface. Still, the accuracy can be increased by partitioning the element into more sub-quads and also by using a higher order Gauss quadrature, i.e. more integration points per sub-quad.

4.2.1 Implementation

First of all, the domain of a quadrilateral isoparametric element needs to be divided into sub-quadrilaterals, as shown in Figure 4.2. It is much more convenient to apply this partition at the natural coordinate system directly, resulting in a partition into sub-rectangles. Thus any integral that has been expressed with respect to natural coordinates can be written as:

$$I = \int_{-1}^{1}\int_{-1}^{1} g(\boldsymbol{\xi}) d\xi d\eta = \sum_{q=1}^{n_{quads}} \int_{\xi_1}^{\xi_2}\int_{\eta_1}^{\eta_2} g(\boldsymbol{\xi}) d\xi d\eta \tag{4.1}$$

where

- The integrand includes the determinant of the isoparametric mapping's Jacobian matrix:

  $$g(\xi, \eta) = f(\boldsymbol{\xi}) \cdot \left|\mathbf{J}_{NG}\right| \tag{4.2}$$

  For example in (2.70) $f(\boldsymbol{\xi})$ represents the original integrand, which is multiplied with $\left|\mathbf{J}_{NG}\right|$ during the change of variables technique.

- ξ_1, ξ_2, η_1, η_2 are the limits of each sub-rectangle q in the natural system, as depicted in Figure 4.2

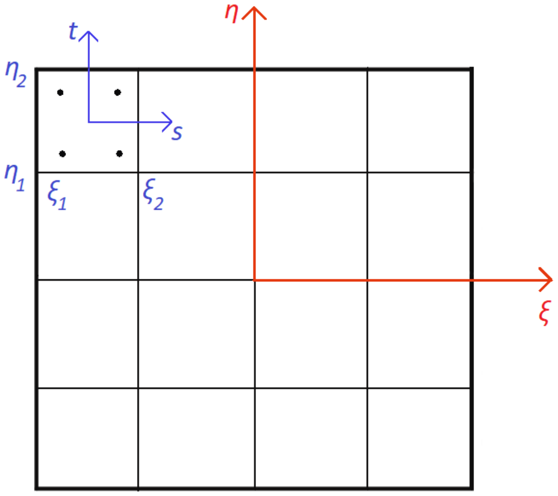


Fig. 4.2 Numerical integration of enriched elements using sub-quadrilaterals

An auxiliary coordinate system (s,t) can be defined for each sub-rectangle. The mapping from this auxiliary system to the natural system is a linear transformation (without any rotation):

$$\xi = \frac{\xi_2 - \xi_1}{2} s + \frac{\xi_1 + \xi_2}{2} \tag{4.3a}$$

$$\eta = \frac{\eta_2 - \eta_1}{2} t + \frac{\eta_1 + \eta_2}{2} \tag{4.3b}$$

The Jacobian matrix of the direct mapping and its determinant are

$$\mathbf{J}_{AN} = \begin{bmatrix} \xi,_s & \xi,_t \\ \eta,_s & \eta,_t \end{bmatrix} = \begin{bmatrix} \frac{\xi_2-\xi_1}{2} & 0 \\ 0 & \frac{\eta_2-\eta_1}{2} \end{bmatrix} \tag{4.4a}$$

$$\left|\mathbf{J}_{AN}\right| = \frac{\xi_2 - \xi_1}{2} \frac{\eta_2 - \eta_1}{2} \tag{4.4b}$$

Applying the change of variables technique between the natural and auxiliary system

$$I_q = \int_{\xi_1}^{\xi_2} \int_{\eta_1}^{\eta_2} g(\boldsymbol{\xi}) d\xi d\eta = \int_{-1}^{1} \int_{-1}^{1} g(\mathbf{s}) \left|\mathbf{J}_{AN}(\mathbf{s})\right| dsdt \tag{4.5}$$

Now the integral is again evaluated over $[-1,1] \times [-1,1]$ and the conventional Gauss-Legendre quadrature can be applied

$$I_q = \int_{-1}^{1} \int_{-1}^{1} g(\mathbf{s}) \left| \mathbf{J}_{AN} \right| dsdt = \sum_{k \in M_{GP}(q)} g\left(\mathbf{s}_k\right) \cdot \left| \mathbf{J}_{AN}\left(\mathbf{s}_k\right) \right| \cdot \hat{w}_k \tag{4.6}$$

where

- $M_{GP}(q)$ is the set of integration points of the quadrilateral q.
- $\mathbf{s}_k$ and $\hat{w}_k$ are their coordinates in the auxiliary system and their weights. Both these can be obtained by the well known Gauss-Legendre quadrature formulas or tables.

Since $g\left(\mathbf{s}_k\right)$ will be evaluated in the natural coordinate system and most finite element routines do not have knowledge of this auxiliary system anyway, it is more convenient to express the integration points in terms of natural coordinates and incorporate $\left|\mathbf{J}_{AN}\right|$ into the weights:

$$\xi_k = \frac{\xi_2 - \xi_1}{2} s_k + \frac{\xi_1 + \xi_2}{2} \tag{4.7a}$$

$$\eta_k = \frac{\eta_2 - \eta_1}{2} t_k + \frac{\eta_1 + \eta_2}{2} \tag{4.7b}$$

$$w_k = \frac{\xi_2 - \xi_1}{2} \frac{\eta_2 - \eta_1}{2} \hat{w}_k \tag{4.7c}$$

The triplets $\left(\xi_k, \eta_k, w_k\right)$ can now be used the same way conventional Gauss points would be, thus abstracting the whole sub-quads procedure. Therefore the integral over the whole element can be written as

$$I = \int_{-1}^{1} \int_{-1}^{1} g(\boldsymbol{\xi}) d\xi d\eta = \sum_{q=1}^{n_{quads}} \sum_{k \in M_{GP}(q)} g\left(\boldsymbol{\xi}_k\right) \cdot w_k \tag{4.8}$$

All steps described above are summarized in Algorithm 4.1:

Algorithm 4.1 Integration over an element partitioned into sub-quadrilaterals

```
Input: $n_\xi$, $n_\eta$ are the number of rectangles along the $\xi$, $\eta$ axis of the element respectively. Usually $n_\xi = n_\eta$.
Input: $\left(s_k, t_k, \hat{w}_k\right)$ are the Gauss points obtained by using Gauss-Legendre quadrature for each sub-quad and $n_{GP}$ is their number
$L_\xi \leftarrow \frac{2}{n_\xi}$
$L_\eta \leftarrow \frac{2}{n_\eta}$
for $i$ from $1$ to $n_\xi$ do
    for $j$ from $1$ to $n_\eta$ do
        $\xi_1 \leftarrow -1 + L_\xi * (i - 1)$
        $\xi_2 \leftarrow -1 + L_\xi * i$
        $\eta_1 \leftarrow -1 + L_\eta * (j - 1)$
        $\eta_2 \leftarrow -1 + L_\eta * j$
        for $k$ from $1$ to $n_{GP}$ do
            $\xi_k \leftarrow \frac{\xi_2 - \xi_1}{2} * s_k + \frac{\xi_1 + \xi_2}{2}$
            $h_k \leftarrow \frac{\eta_2 - \eta_1}{2} * t_k + \frac{\eta_1 + \eta_2}{2}$
            $w_k \leftarrow \frac{\xi_2 - \xi_1}{2} * \frac{\eta_2 - \eta_1}{2} * \hat{w}_k$
Output: The $n_\xi \cdot n_\eta \cdot n_{GP}$ integration points in the natural system, as $\left(\xi_k, \eta_k, w_k\right)$ triplets.
```

4.2.2 Advantages and drawbacks

In general the sub-quads of each isoparametric quadrilateral element form a curvilinear grid in the global cartesian system, which can be obtained by mapping a rectilinear grid in the natural coordinate system with the standard shape functions. This can be observed in Figure 4.2, where the the element's domain is partitioned into sub-rectangles in the natural coordinate system. Moreover the integration takes place in the natural coordinate system, where the sub-divisions are square or at least rectangular. If the partition was performed in the global coordinate system, as in Section 4.3, the global coordinates of the integration points would have to be converted to natural coordinates, which would require using the cumbersome inverse isoparametric mapping.

Consequently, the subdivision of this method is very easy to implement and computationally efficient. It can also be extended to higher order elements and 3D or non linear problems without any additional complications. Its main drawback is the reduced accuracy, but that can be mitigated significantly, by using more sub-quads and more integration points per sub-quad. Note that doing so will increase the computational cost, so a satisfactory trade-off solution should be sought.

On the other hand, this method cannot be applied for triangular finite elements, which dominate unstructured meshes. Nevertheless, one could develop similar partitioning methods that are computationally efficient, but like the quadrilateral version do not take the crack geometry onto consideration, thus exhibiting decreased accuracy.

4.3 Integration with sub-triangles

A second approach that was also proposed in [19] is to divide an element that is intersected by the crack interface into sub-triangles, such that their edges conform to the crack. None of these triangles is intersected by the crack, and thus conventional Gauss quadrature can be applied for integrating over each one.

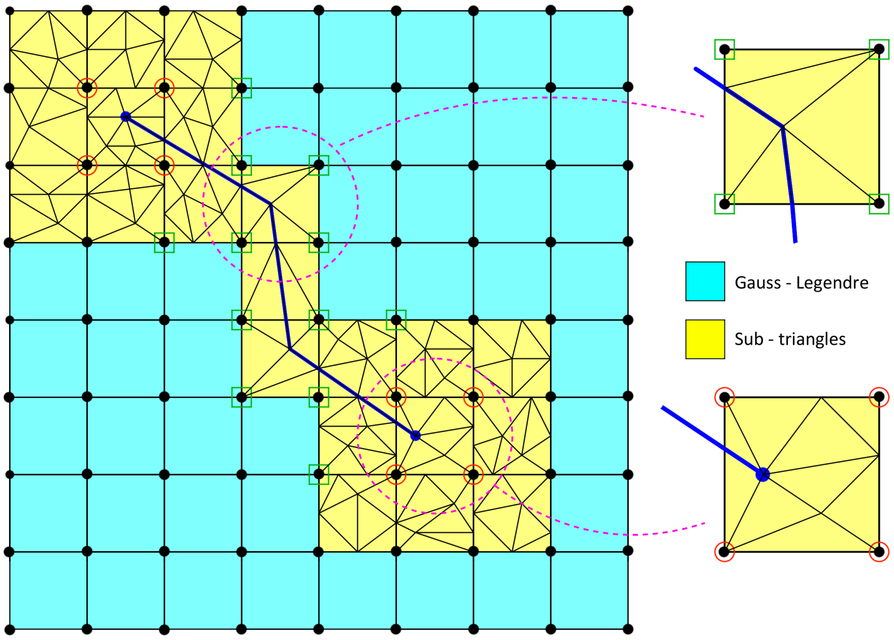


Fig. 4.3 Numerical integration of enriched elements using sub-triangles

Contrary to Section 4.2, this integration rule is exact for elements that do not have nodes enriched with asymptotic tip functions, since the integrands are continuous in *all* sub-triangles. As before, there is no need to use more integration points for these elements than classical FEM uses. On the other hand, integrating over elements with tip enriched nodes involves singular terms ($\frac{1}{\sqrt{r}}$) in the integrand. Such integrations are not exact but accuracy can be increased by using more sub-triangles and more integration points per sub-triangle, similar to Section 4.2. To increase the number of sub-triangles, once completing the initial triangulation that conforms to the crack geometry, further divide the produced sub-triangles into smaller ones. Typically these improvements are employed at least for the element containing the crack tip.

It is not necessary to use only triangles for the element partition. Quadrilaterals and even polygons with known integration rules could be used, as long as no such sub-division is intersected by the crack interface. For elements without singularities, such sub-divisions can be more efficient than triangles, since less integration points are needed in total.

Nevertheless, triangles are easy to work with and will be preferred in this thesis. More importantly, triangular meshes are very popular and there are a lot of algorithms, the most famous of which is the Delauny triangulation. In this case however, a constrained Dalauny triangulation algorithm is needed in order to guarantee that sub-triangles conform to the crack geometry.

4.3.1 Implementation

Partition of element into sub-triangles

First of all, the element's domain needs to be divided into sub-triangles. Assuming that the crack geometry is piece-wise linear (rather than curved):

1. Identify the following points of interest inside the element:

    - crack tip
    - kink points: points where two consecutive crack segments meet
    - intersection points: points where the crack segments intersect the element's boundaries.

    The determination of these points depends on the method chosen to model that crack geometry and will be elaborated in Sections 5.2.5 and 5.3.5.

2. Convert the global cartesian coordinates of these points to natural coordinates using the inverse isoparametric mapping described in Appendix A.1.4. In this thesis, only first order elements (tri3, quad4) are considered, for which there exist analytic expressions. For higher order elements, iterative numerical methods would have to be employed, hence increasing the computational cost.

   The element's nodes also need to be converted to the natural system. However this is trivial as the natural coordinates of nodal points are standard. E.g. the bottom-left node of a quad4 element is always placed at $(\xi, \eta) = (-1, -1)$. The nodes, crack tip, kink points and intersection points are inputted into the constrained Delauny triangulation and will be called ”triangulation points” from now on.

3. In addition to the triangulation points, the crack segments are also needed as input for the constrained Delauny triangulation. Mapping these line segments to the natural coordinate system is complicated in general. However, for first order elements the direct and inverse isoparametric mappings are linear transformations, thus lines are mapped into lines. This means that the crack segments in the natural coordinate system can be defined simply by their end points (crack tip, intersection points, kink points).

4. Using the triangulation points and crack segments as input, apply the constrained Delauny triangulation in the natural system to obtain 3 vertices (ξ_i, η_i) for each subtriangle. Since triangulation algorithms are a subject of Computational Geometry and out of this thesis' scope, the reader is referred to [16] for more details.

Determination of integration points

Any integral that has been expressed with respect to natural coordinates can be written as:

$$I = \int_{-1}^{1} \int_{-1}^{1} g(\boldsymbol{\xi}) d\xi d\eta = \sum_{t=1}^{n_{triangles}} \int_{\Omega_t} g(\boldsymbol{\xi}) d\xi d\eta \tag{4.9}$$

where

- The integrand includes the determinant of the isoparametric mapping's Jacobian matrix:

  $$g(\xi, \eta) = f(\boldsymbol{\xi}) \cdot \left| \mathbf{J}_{NG} \right| \tag{4.10}$$

  For example in (2.70) $f(\boldsymbol{\xi})$ represents the original integrand, which is multiplied with $\left| \mathbf{J}_{NG} \right|$ during the change of variables technique.

- Ω_t is the domain of the sub-triangle t in the natural coordinate system, as depicted in Figure 4.4

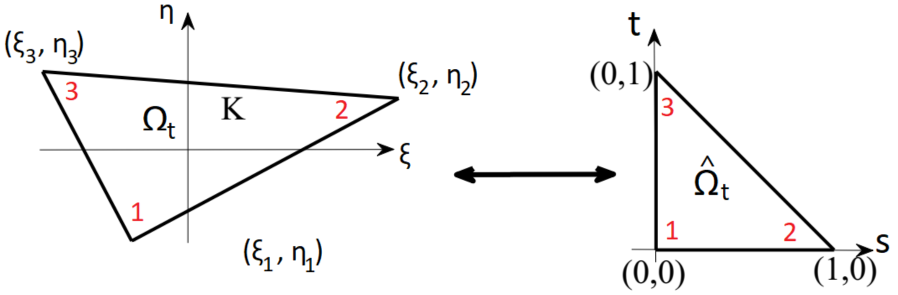


Fig. 4.4 Numerical integration of enriched elements using sub-quadrilaterals

As with the partition into the sub-quads, we define an auxiliary coordinate system (s,t) for each sub-triangle, as shown in Figure 4.4. The mapping from this auxiliary system to the natural system is a linear transformation, identical to the one used in the isoparametric formulation:

$$\boldsymbol{\xi} = (1-s-t)\boldsymbol{\xi}_1 + s\boldsymbol{\xi}_2 + t\boldsymbol{\xi}_3 \tag{4.11}$$

where $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, $\boldsymbol{\xi}_3$ are the natural coordinates of the triangle's vertices, obtained by the triangulation algorithm. The Jacobian matrix of the direct mapping is

$$\mathbf{J}_{AN} = \begin{bmatrix} \xi,_s & \xi,_t \\ \eta,_s & \eta,_t \end{bmatrix} = \begin{bmatrix} \xi_2 - \xi_1 & \xi_3 - \xi_1 \\ \eta_2 - \eta_1 & \eta_3 - \eta_1 \end{bmatrix} \tag{4.12}$$

and its determinant is

$$\begin{aligned} \left|\mathbf{J}_{AN}\right| &= \left(\xi_2 - \xi_1\right)\left(\eta_3 - \eta_1\right) - \left(\xi_3 - \xi_1\right)\left(\eta_2 - \eta_1\right) \\ &= \xi_1\left(\eta_2 - \eta_3\right) + \xi_2\left(\eta_3 - \eta_1\right) + \xi_3\left(\eta_1 - \eta_2\right) \end{aligned} \tag{4.13}$$

Note that $\left|\mathbf{J}_{AN}\right| = 2A_t$, with A_t being the area of the triangle in the natural system. Applying the change of variables technique between the natural and auxiliary system leads to

$$I_t = \int\limits_{\Omega_t} g(\boldsymbol{\xi}) d\xi d\eta = \int\limits_{\hat{\Omega}_t} g(\mathbf{s}) \left|\mathbf{J}_{AN}(\mathbf{s})\right| ds\, dt \tag{4.14}$$

Now the integral is evaluated over the reference triangle in the auxiliary system. The traditional symmetrical Gauss quadrature can be applied on this reference triangle, resulting in

$$I_t = \int\limits_{\hat{\Omega}_t} g(\mathbf{s}) \left|\mathbf{J}_{AN}(\mathbf{s})\right| ds\, dt = \sum_{k \in M_{GP}(t)} g\left(\mathbf{s}_k\right) \cdot \left|\mathbf{J}_{AN}\left(\mathbf{s}_k\right)\right| \cdot \hat{w}_k \tag{4.15}$$

where

- $M_{GP}(t)$ is the set of integration points of the sub-triangle t.
- $\mathbf{s}_k$ and $\hat{w}_k$ are their coordinates in the auxiliary system and their weights. Both these can be obtained by the well known symmetrical Gauss quadrature formulas or tables.

> Note that sometimes symmetrical Gauss quadrature is written as
>
> $$\int\limits_{\Omega_t} f(x, y) dx dy = \frac{1}{2} \sum_{k \in M_{GP}} f\left(x_k, y_k\right) w_k$$
>
> In this thesis the $\frac{1}{2}$ is assumed to be incorporated into the weights of symmetrical Gauss quadrature.

Since $g\left(\mathbf{s}_k\right)$ will be evaluated in the natural coordinate system and most finite element routines do not have knowledge of this auxiliary system anyway, it is more convenient to express the integration points in terms of natural coordinates and incorporate $\left|\mathbf{J}_{AN}\right|$ into the weights:

$$\xi_k = \left(1 - s_k - t_k\right) \xi_1 + s_k \xi_2 + t_k \xi_3 \tag{4.16a}$$

$$\eta_k = \left(1 - s_k - t_k\right) \eta_1 + s_k \eta_2 + t_k \eta_3 \tag{4.16b}$$

$$w_k = \left[\xi_1 \left(\eta_2 - \eta_3\right) + \xi_2 \left(\eta_3 - \eta_1\right) + \xi_3 \left(\eta_1 - \eta_2\right)\right] \hat{w}_k \tag{4.16c}$$

The triplets $\left(\xi_k, \eta_k, w_k\right)$ can now be used the same way conventional Gauss points would be, thus abstracting the whole sub-triangles procedure. Therefore the integral over the whole element can be written as

$$I = \int\limits_{-1}^{1} \int\limits_{-1}^{1} g(\boldsymbol{\xi}) d\xi d\eta = \sum_{t=1}^{n_{triangles}} \sum_{k \in M_{GP}(t)} g\left(\boldsymbol{\xi}_k\right) \cdot w_k \tag{4.17}$$

Note that the partition of the element into sub-triangles that conform to the crack geometry was performed in the natural coordinate system. An alternative approach would be to triangulate directly in the global cartesian system. Indeed that would eliminate the need to convert the triangulation points and crack segments to the natural system. The inverse isoparametric mapping of line segments is exact for first order elements, but approximations might be necessary for higher order.

Furthermore, all integration points would be determined in the auxiliary system and then mapped to the global cartesian system. However, their coordinates would have to be converted to the natural system as well, where many quantities (e.g. shape functions) are defined in X-FEM. This would require even more computations, as there are more integration points than triangulation points. Moreover, the integral of (4.17) would be defined in the global cartesian system and instead of $\left(\xi_k, \eta_k, w_k\right)$ triplets, the integration points would be expressed as $\left(x_k, y_k, w_k\right)$ triplets.

While this is not a problem by itself, it deviates from the formulation presented so far. All the integrals introduced in Chapters 2 and 3 are transformed from the global cartesian to the natural system and the Jacobian's determinant $\left|\mathbf{J}_{NG}\right|$ is present in all equations. This is also how most coding implementations would handle integration. Therefore, it is preferable to be consistent and perform the triangulation in the natural coordinate system.

4.3.2 Advantages and drawbacks

The main advantage of this method is its ability to perform exact integration, independently of the mesh size, over elements where the displacement field exhibits discontinuities but not singularities. As the crack propagates, most enriched or blending elements fall under this category and thus this method exhibits higher accuracy than the one in Section 4.2. However this requires an element partition that conforms to the crack geometry, which is much more cumbersome than the simple partition of Section 4.2. It also necessitates the use of computational geometry algorithms and depends on the method used for modeling the crack geometry.

Moreover, the triangulation points must be converted from the global cartesian to each element's natural system. This is adds yet another layer of computations, which are cumbersome for higher order elements. In various problems data must be stored at each integration point. For example if the material is non linear, applying the non linear constitutive law requires storing the strain history at each integration point. As the crack propagates, its geometry changes, resulting in different partitions of the same element at different iterations, until the crack tip moves away from said element. As the sub-triangles change over time, so do their integration points, creating the need to transfer data between the old and new ones.

Another concern is that this technique requires the crack geometry to be linear or piece-wise linear inside the elements. Curved cracks must first be linearized, resulting in a loss of accuracy. Nevertheless, non linear problems, higher order elements and curved cracks are out of the scope of this thesis and thus partitioning into sub-triangles is a valid alternative for the simple cases examined here.

Chapter 5

Geometric aspects

5.1 Introduction

XFEM removes the need of conforming the finite element mesh to the crack interface by appropriately enriching certain nodes. Nevertheless, the crack geometry is still an important aspect of many procedures when using XFEM for crack propagation problems. The accurate description and update of the crack geometry and its interaction with the mesh are quite tricky even for 2D problems.

The most common operations that involve the crack geometry are:

- Determining the local cartesian and local polar system at each crack tip.
- Calculating the signed distance from any point to the crack interface.
- Finding which elements are intersected by the crack and which contain crack tips. The nodes of the former elements will be enriched with Heaviside function, while the nodes of the latter, will be enriched with asymptotic tip functions.
- Decomposing an element into sub-polygons that conform to the the crack geometry. This is necessary for certain integration rules (see Section 4.3) and for calculating the subareas of the local support of a node above and below the crack (see Sections 2.4.1 and 2.4.4). In practice, the element is decomposed into triangles, since triangulation algorithms are readily available, such as the constrained Delauny triangulation.
- Updating the crack representation during propagation. In a 2D setting the crack grows from the tips, according to the direction and length described in Sections 3.4.2 and 3.4.3. The rest of the existing crack body remains the same.

In general, two categories of crack representation methods can be observed:

- Explicit crack descriptions. The crack is represented as a collection of geometric primitives: straight line segments in 2D and flat triangles in 3D. Updating the crack geometry is easily performed by adding a new building block to the existing crack. Crack-mesh interactions also reduce to interactions of the mesh with these primitives. However, such operations tend to require cumbersome computational geometry algorithms that cannot be easily extended to 3D.

- Implicit crack descriptions. These are usually adaptations of the Level Set Method (LSM) for describing curves. They store the signed distance (level set function) from the mesh nodes to the crack. The crack curve is identified as the locus of all points with a zero level set. Implicit methods tend to be more efficient in crack-mesh interaction operations and can be intuitively extended to 3D problems. However, their accuracy depends on the mesh size and the order of the finite elements. In addition to not being exact, there are complications in the geometry update procedures as well.

5.2 Explicit crack geometry description

In this section a basic explicit crack description for 2D problems will be presented. The crack geometry is modeled as a polyline, which is a piece-wise linear curve, that is, it consists of a series of straight line segments. Such a crack can be seen in Figure 5.1. In this approach, the line segments or the vertices of the polyline must be explicitly stored in the correct order.

For interior cracks, the crack tips coincide with the first and last vertex of the polyline. For exterior cracks, there is only one crack tip and its corresponding end of the polyline, while the other end is called the crack ”mouth”. All other vertices are called kink points, since they belong to exactly two line segments with generally different orientation.

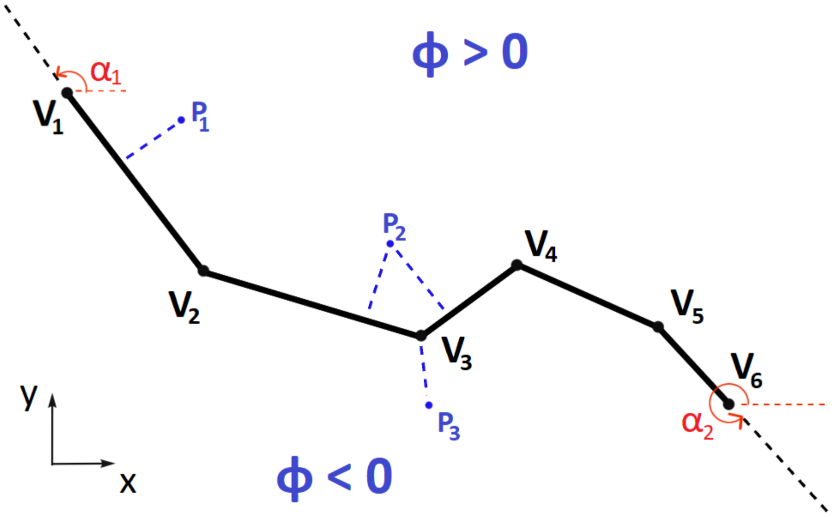


Fig. 5.1 Explicit crack description. The crack is represented as a series of straight line segments.

The local cartesian coordinate systems at the start and end crack tip are defined by the orientation of the first and last line segment. These orientations are defined as counter-clockwise angles α_1 and α_2 from the global x axis to the line segments, or more accurately to their extensions outwards, as shown in Figure 5.1.

5.2.1 Signed distances

Figure 5.2 illustrates the contours of the signed distance function $\varphi(\mathbf{x})$ at all points of the plane, with the crack interface corresponding to the contour $\varphi = 0$. $\varphi(\mathbf{x})$ is often called the level set function. To calculate the signed distance of an arbitrary point, the closest line segment or vertex of the polyline must first be identified. For example, consider Figure 5.1:

- The signed distance of P_1 is the distance of P_1 to the segment V_1V_2 with a positive sign.
- P_2 is close to two line segments. Its signed distance is the minimum distance of P_1 to the segments V_2V_3 and V_3V_4 with a positive sign.
- P_3 associated with vertex V_3 instead of any line segment. Its signed distance is $-\left|P_3V_3\right|$.

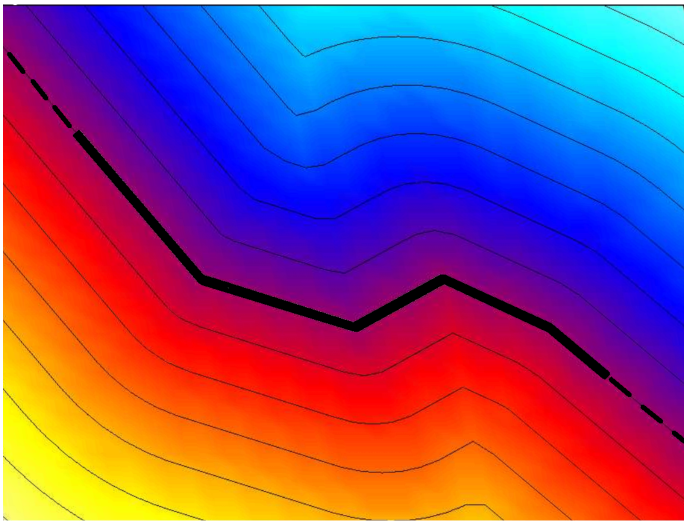


Fig. 5.2 Contours of the signed distance function $\varphi(\mathbf{x})$.

There are various ways to calculate the signed distance of a point P to a line segment. When dealing with a polyline, an efficient approach is to use a local cartesian coordinate system $(\hat{x}, \hat{y})$ for each line segment, as depicted in Figure 5.4. This local system has the following properties:

- Its origin is placed at the start of each line segment V_iV_{i+1}, that is at the vertex with the lowest index V_i.
- The local $\hat{x}$ axis is collinear with the segment itself.
- The positive local $V_i\hat{y}$ half-axis is oriented towards the region where signed distances are positive.

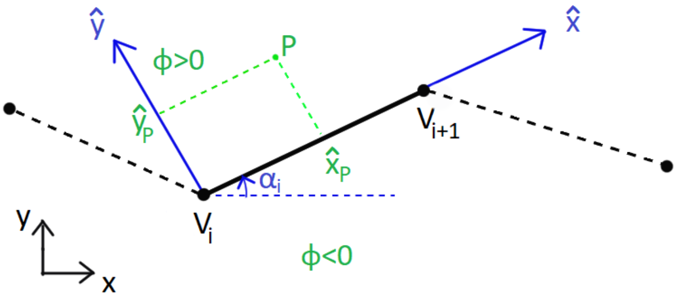


Fig. 5.3 Local coordinate system of a crack segment.

To use the coordinate system of segment S_i with vertices V_i, V_{i+1}, first convert the global coordinates (x, y) of P to local coordinates $(\hat{x}, \hat{y})$. This is similar to the local cartesian system defined at the crack tip, which is described in Appendix A.2.1. In this case:

$$\hat{\mathbf{x}} = \mathbf{Q}_i \cdot \mathbf{x} + \mathbf{b}_i \tag{5.1}$$

where

- $\mathbf{Q}_i$ is an orthonormal rotation matrix

$$\mathbf{Q}_i = \begin{bmatrix} cos\alpha_i & sin\alpha_i \\ -sin\alpha_i & cos\alpha_i \end{bmatrix} \tag{5.2}$$

- $\mathbf{b}$ is an offset vector

$$\mathbf{b}_i = -\mathbf{Q}_i \cdot \mathbf{x}_{V_i} \tag{5.3}$$

- $\mathbf{x}_{V_i}$ are the global coordinates of the start vertex of the line segment.

- α_i is the counter clockwise angle from global x axis to local $\hat{x}$ axis.

The signed distance is now easily obtained as $\varphi(\mathbf{x}) = \hat{y}$. The $\hat{x}$ coordinate of the point is also quite useful, since it determines if the point P should be associated with the segment S_i. Consider Figure 5.4a, where the lines that are perpendicular to the segment S_i and pass through its vertices are drawn. Three regions can then be identified:

- The region before the segment $\Omega_1^{(i)}$: $\mathbf{x} \in \Omega_1^{(i)} \Leftrightarrow \hat{x} < 0$

- The region where points are associated with the segment $\Omega_2^{(i)}$: $\mathbf{x} \in \Omega_2^{(i)} \Leftrightarrow 0 \leq \hat{x} \leq L_i$, where L_i is the length of the segment.

- The region after the segment $\Omega_3^{(i)}$: $\mathbf{x} \in \Omega_3^{(i)} \Leftrightarrow \hat{x} > L_i$

Generally only points in $\Omega_2^{(i)}$ are associated with the segment S_i. However, points are associated with the first segment when $\mathbf{x} \in \Omega_1^{(\text{first})} \cup \Omega_2^{(\text{first})}$. Similarly points are associated with the last segment when $\mathbf{x} \in \Omega_2^{(\text{last})} \cup \Omega_3^{(\text{last})}$.

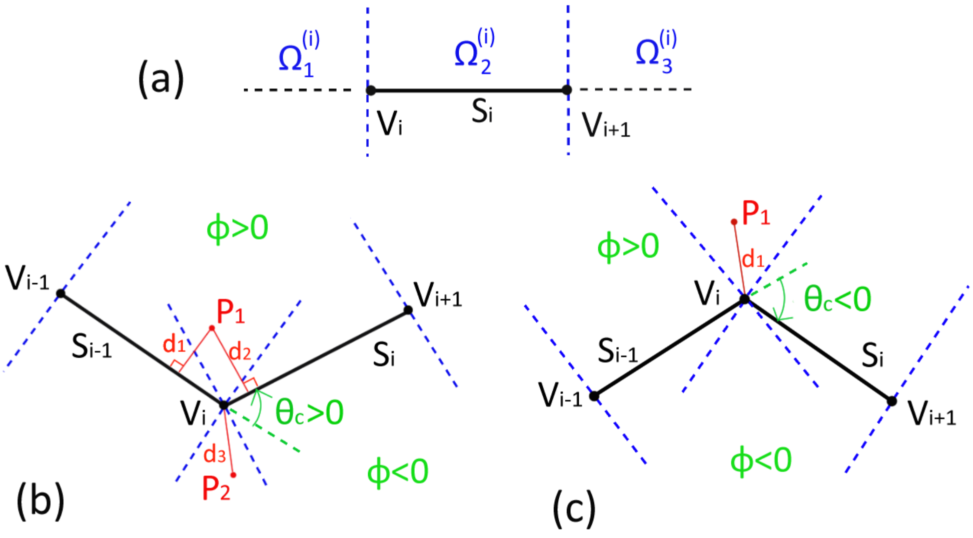


Fig. 5.4 Determining the signed distance of a point.

For a given point P, only the signed distances to the segments it is associated with must be considered.

- If there is only one segment associated with P, then that signed distance is used as $\varphi(P)$.
- If two or more segments are associated with P, then $\varphi(P)$ is the signed distance with the minimum absolute value, of all signed distances of P to the relevant segments:

$$I = minarg\left(\left|d_i\right|\right) \tag{5.4a}$$

$$\varphi(P) = d_I \tag{5.4b}$$

For example, in Figure 5.4b P_1 is associated with segments S_{i-1} and S_i, since $P_1 \in \Omega_2^{(i-1)} \cup \Omega_2^{(i)}$. The corresponding signed distances are d_1 and d_2, with $\left|d_1\right| < \left|d_2\right|$. Therefore $\varphi\left(P_1\right) = d_1$.

- It is also possible that P is not associated with any segment, which means that $P \notin \Omega_2^{(j)}$, $\forall j$. Also note that $P \in \Omega_3^{(i-1)} \cup \Omega_1^{(i)}$ for some i. In this case:
  - The absolute distance is defined as the distance between P and the vertex V_j

$$\left|\varphi(P)\right| = \left|PV_i\right| \tag{5.5}$$

– To determine the sign of $\varphi(P)$, consider the counter-clockwise angle from segment S_i to S_{i-1}. If that angle is convex, then $\varphi(P) < 0$. Conversely if the that angle is non convex, than $\varphi(P) > 0$. It is easier to work with the propagation angle θ_c instead of the angle between the segments. In this approach

$$\varphi(P) = \begin{cases} -\left|PV_i\right| & ,\theta_c > 0 \\ +\left|PV_i\right| & ,\theta_c < 0 \end{cases} \tag{5.6}$$

For example consider P_2 in Figure 5.4b. It is not associated with any segments, since $P_2 \in \Omega_3^{(i-1)} \cup \Omega_1^{(i)}$. In addition $\theta_c > 0$. Thus $\varphi\left(P_2\right) = -\left|P_2V_i\right|$. On the other hand, $\theta_c < 0$ in Figure 5.4c. Therefore $\varphi\left(P_1\right) = +\left|P_1V_i\right|$.

For an efficient implementation, L_i,$\mathbf{Q}_i$ and $\mathbf{b}_i$ and θ_c should be stored for each line segment when it is first added to the polyline. The following listing describes an algorithm to calculate the signed distance of an arbitrary point, according to the rules described above.

During an XFEM analysis, Algorithm 5.1 will be called for each integration point. Note that for each integration point, all line segments are iterated resulting in a complexity of $O\left(n_S\right)$. Even though, more efficient versions are possible, e.g. by terminating the iteration over the segments once all the relevant ones have been found, by caching which segments are relevant for the integration points of the same element, etc, the complexity remains $O\left(n_S\right)$. This inefficiency is an inherent disadvantage of explicit descriptions, since they are composed of multiple elementary shapes: line segments in 2D and flat triangles in 3D.

Algorithm 5.1 Calculation of the signed distance from an arbitrary point to the polyline.

```
Input: $V$, $S$, $\Theta$ are *lists* containing the $n_S + 1$ vertices, $n_S$ segments and $n_S - 1$ propagation angles of the crack polyline. Segment $S[i]$ has $V[i]$, $V[i+1]$ as vertices. Its propagation angle from the previous segment is $\Theta[i-1]$, since there is no such angle for the first segment.
Input: $L$, $Q$, $b$ are *lists* containing the lengths, rotation matrices and offset vectors of the $n_S$ segments.
Input: $\mathbf{x}$ are the global coordinates of the point $P$ in question.
Let $D$ be a *list* of the signed distances to various segments or vertices encountered.
$is\Omega_3 \leftarrow$ false    ▷ a flag signaling if $P$ is in the $\Omega_3$ region of the last processed segment
$\hat{\mathbf{x}} \leftarrow Q[1] \cdot \mathbf{x} + b[1]$ according to (5.1)    ▷ Process first line segment
if $\hat{x} \leq L[1]$ then
    Add $\hat{y}$ to $D$.
else
    $is\Omega_3 \leftarrow$ true
for $i$ from 2 to $n_S - 1$ do    ▷ Process subsequent line segments
    $\hat{\mathbf{x}} \leftarrow Q[i] \cdot \mathbf{x} + b[i]$ according to (5.1)
    if $\hat{x} < 0$ then
        if $is\Omega_3 =$ true then
            $d \leftarrow -sign(\Theta[i-1]) \left\| \hat{\mathbf{x}} - V[i] \right\|$ according to (5.6)
            Add $d$ to $D$.
        $is\Omega_3 \leftarrow$ false
    else if $\hat{x} \leq L[i]$ then
        Add $\hat{y}$ to $D$.
        $is\Omega_3 \leftarrow$ false
    else
        $is\Omega_3 \leftarrow$ true
$\hat{\mathbf{x}} \leftarrow Q[n_S] \cdot \mathbf{x} + b[n_S]$ according to (5.1)    ▷ Process last line segment
if $\hat{x} < 0$ then
    if $is\Omega_3 =$ true then
        $d \leftarrow -sign\left(\Theta[n_S - 1]\right) \left\| \hat{\mathbf{x}} - V[n_S] \right\|$ according to (5.6)
        Add $d$ to $D$.
else
    Add $\hat{y}$ to $D$.
Find the index $j$ into $D$ for which $\left| D[j] \right|$ is minimum.
Output: The signed distance of point: $\varphi(P) = D[j]$
```

5.2.2 Detecting tip elements

Determining if an element contains the crack tip is equivalent to figuring out if the crack tip is inside or on the boundary of the polygonal outline of the finite element. This is vary common computational geometry procedure called point-in-polygon. In this case, the outline of finite elements are convex polygons. A simple test to determine if a point P is inside a convex polygon is illustrated in Figure 5.5.

1. For each edge $e_k = V_i V_j$ of the polygon, form the triangle with vertices $\left(P, V_i, V_j\right)$ and calculate its area A_k according to (2.55).
2. Calculate the the sum of all triangles' areas.
3. Calculate the area of the convex polygon according to

$$A_{pol} = \frac{1}{2}\left|\left(x_1 y_2 - y_1 x_2\right) + \left(x_2 y_3 - y_2 x_3\right) + \cdots + \left(x_n y_1 - y_n x_1\right)\right| \tag{5.7}$$

   where the vertices of the polygon $1, 2, \cdots n$ are numbered in a counter-clockwise order.
4. If this sum is equal to the area of the polygon, then the point lies inside the polygon or on its boundary. Otherwise it lies outside.

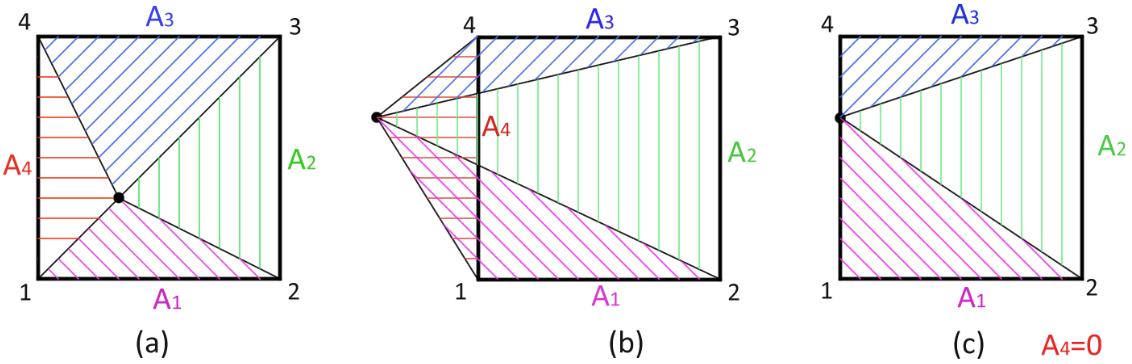


Fig. 5.5 Point in polygon test: a) Point inside polygon, b) Point outside polygon, c) Point on polygon boundary

The previous test includes points inside the polygon and on its boundary in the same group, which is desired when identifying tip elements. Some care must be taken regarding the tolerance of the equivalence between the area of the polygon and the sum of triangles' areas. Of course, there are other various other procedures, e.g. ray casting algorithms, that may be more efficient and robust.

Note that more than one elements will be identified as tip elements, if the crack tip lies on an element edge or a node. While this is very improbable to happen during crack propagation, it is still quite possible in the first iteration, if the mesh is aligned with the initial crack geometry. All elements, whose outline contains or passes through a crack tip, must have their nodes enriched with the asymptotic tip functions. In any case, such mesh dependencies can be avoided by using a fixed area enrichment scheme (see Section 2.4.2).

5.2.3 Intersection with line segments

Before delving into the interaction of the crack polyline with the mesh's elements, it is beneficial to go over the relative positions of two line segments. Figure 5.6 depicts the possible configurations we will be interested in. More distinctions could be made, but they would not be useful for the purposes of this chapter.

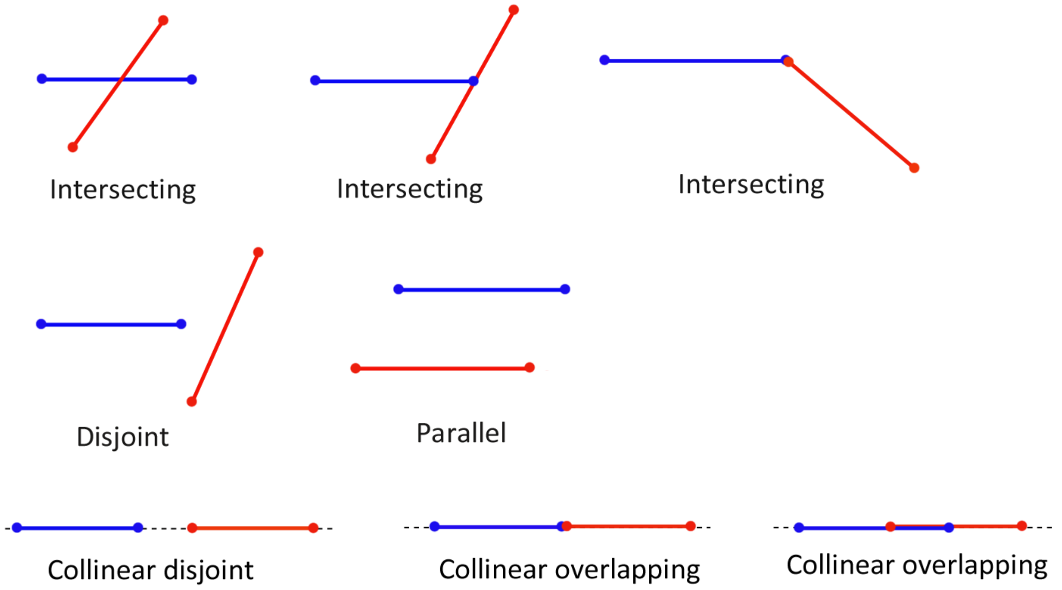


Fig. 5.6 Relative positions of two line segments.

A computational geometry method for determining the relative position and possible intersection point of two line segments will be presented in the following:

- Some important properties of the cross product of two vectors $\mathbf{v}, \mathbf{w}$ in 2D are

$$\mathbf{v} \times \mathbf{w} = \begin{vmatrix} v_x & v_y & 0 \\ w_x & w_y & 0 \\ 0 & 0 & 1 \end{vmatrix} = v_x\, w_y - v_y\, w_x \tag{5.8a}$$

$$(-\mathbf{v}) \times \mathbf{w} = -(\mathbf{v} \times \mathbf{w}) \tag{5.8b}$$

$$\mathbf{w} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{w}) \tag{5.8c}$$

$$\mathbf{v} \times \mathbf{w} = 0 \Leftrightarrow \mathbf{v}, \mathbf{w} \text{ are parallel} \tag{5.8d}$$

- Assume two line segments defined by their end points $S_1 = \left(\mathbf{x}_1, \mathbf{x}_2\right)$ and $S_2 = \left(\mathbf{x}_1, \mathbf{x}_2\right)$, as depicted in Figure 5.7. Their parametric forms are

$$S_1 = \left\{\mathbf{x}_1 + a\left(\mathbf{x}_2 - \mathbf{x}_1\right) \ : \ a \in [0, 1]\right\} \tag{5.9a}$$

$$S_2 = \left\{\mathbf{x}_3 + b\left(\mathbf{x}_4 - \mathbf{x}_3\right) \ : \ b \in [0, 1]\right\} \tag{5.9b}$$

- If the vectors $\mathbf{x}_2 - \mathbf{x}_1$ and $\mathbf{x}_4 - \mathbf{x}_3$ are parallel, that is if

$$\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) = 0 \tag{5.10}$$

  - If the vectors $\mathbf{x}_2 - \mathbf{x}_1$ and $\mathbf{x}_3 - \mathbf{x}_1$ are parallel, that is if

$$\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_3 - \mathbf{x}_1\right) = 0 \tag{5.11}$$

    then project $\mathbf{x}_3 - \mathbf{x}_1$ and $\mathbf{x}_4 - \mathbf{x}_1$ onto $\mathbf{x}_2 - \mathbf{x}_1$

$$a_3 = \frac{\left(\mathbf{x}_3 - \mathbf{x}_1\right) \cdot \left(\mathbf{x}_2 - \mathbf{x}_1\right)}{\left(\mathbf{x}_2 - \mathbf{x}_1\right) \cdot \left(\mathbf{x}_2 - \mathbf{x}_1\right)} \tag{5.12a}$$

$$a_4 = \frac{\left(\mathbf{x}_4 - \mathbf{x}_1\right) \cdot \left(\mathbf{x}_2 - \mathbf{x}_1\right)}{\left(\mathbf{x}_2 - \mathbf{x}_1\right) \cdot \left(\mathbf{x}_2 - \mathbf{x}_1\right)} \tag{5.12b}$$

    * If $\left(a_3 < 0 \textbf{ AND } a_4 < 0\right)$ **OR** $\left(a_3 > 1 \textbf{ AND } a_4 > 1\right)$ then the two segments are *collinear disjoint* (see Figure 5.7b).
    * Else they are *collinear* overlapping.

  - Else if $\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_3 - \mathbf{x}_1\right) \neq 0$ the two line segments are *parallel* (see Figure 5.7c).

- Else if $\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) \neq 0$ then the lines passing through the vectors $\mathbf{x}_2 - \mathbf{x}_1$ and $\mathbf{x}_4 - \mathbf{x}_3$ intersect at the point

$$\mathbf{x}_1 + a_0 \left(\mathbf{x}_2 - \mathbf{x}_1\right) = \mathbf{x}_3 + b_0 \left(\mathbf{x}_4 - \mathbf{x}_3\right) \tag{5.13}$$

Multiplying (5.13) with $\times \left(\mathbf{x}_4 - \mathbf{x}_3\right)$

$$\begin{aligned} \mathbf{x}_1 \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) + a_0 \left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) &= \mathbf{x}_3 \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) + b_0 \left(\mathbf{x}_4 - \mathbf{x}_3\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) \\ a_0 \left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) &= \left(\mathbf{x}_3 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right) \\ a_0 &= \frac{\left(\mathbf{x}_3 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right)}{\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right)} \end{aligned} \tag{5.14}$$

Similarly, by multiplying with $\times \left(\mathbf{x}_2 - \mathbf{x}_1\right)$

$$b_0 = \frac{\left(\mathbf{x}_1 - \mathbf{x}_3\right) \times \left(\mathbf{x}_2 - \mathbf{x}_1\right)}{\left(\mathbf{x}_4 - \mathbf{x}_3\right) \times \left(\mathbf{x}_2 - \mathbf{x}_1\right)} = \frac{\left(\mathbf{x}_3 - \mathbf{x}_1\right) \times \left(\mathbf{x}_2 - \mathbf{x}_1\right)}{\left(\mathbf{x}_2 - \mathbf{x}_1\right) \times \left(\mathbf{x}_4 - \mathbf{x}_3\right)} \tag{5.15}$$

  - If $a_0 \in [0, 1]$ **AND** $b_0 \in [0, 1]$, then the intersection of the two lines lies on the segments S_1 and S_2. The two segments *intersect at the point* $\mathbf{x}_1 + a_0 \left(\mathbf{x}_2 - \mathbf{x}_1\right)$.
  - Else they are *disjoint* (see Figure 5.7a).

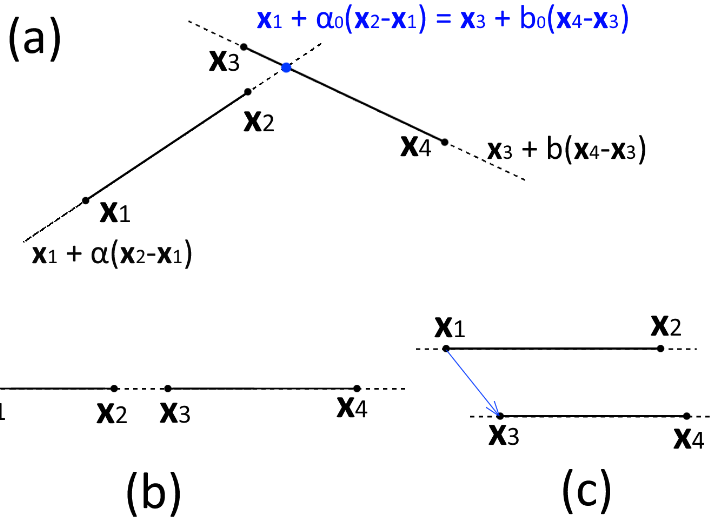


Fig. 5.7 Computing the intersection of two line segments.

5.2.4 Detecting elements intersected by the crack

To determine if an element is intersected by the crack polyline, one has to check if any line segment of the crack intersects or overlaps with any edge of the element, as demonstrated in Section 5.2.3. This requires iterating over all segments or at least until a segment that intersects the element boundary is found. This is an inefficient operation with $O\left(n_S\right)$ complexity. As in Section 5.2.1, this inefficiency is inherent for explicit crack descriptions.

Some examples are given in Figure 5.8. Note that if two successive vertices lie inside the element, the line segment between them will not intersect any element edge. However, the segments before and after it will indeed intersect element edges, so the element will be correctly flagged as intersected. Nevertheless, such configurations introduce a lot of problems (e.g. for J-integrals) and should generally be avoided by selecting the element size to be sufficiently smaller than the crack propagation length.

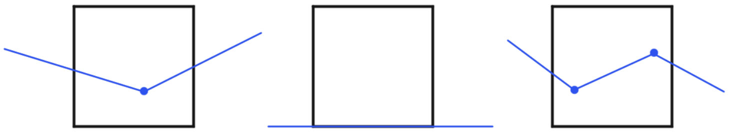


Fig. 5.8 Elements intersected by the polyline.

5.2.5 Triangulation of elements

The aim here is to identify the points and line segments that will be used as input for a constrained triangulation algorithm. The points will become vertices and the line segments edges of the triangular mesh. The points of interest are:

- The crack tip(s)
- The rest of the polyline's vertices, also called kink points
- Intersection points of crack segments with element edges
- The element's nodes

The line segments of interest are defined by the points of interest, except for the nodes. Handling crack tips is the same as handling all other vertices of the polyline, when identifying the points and segments of interest. In contrast, when using the integration described in Section 4.3, one needs to take spacial care of the tip elements, e.g. by further refining the triangular mesh.

Figures 5.9 to 5.11 depict some of the possible relative positions of an element and a line segment, along with the corresponding triangular meshes. To reduce the number of possibilites, no distinction is made whether the vertices of the line segment are tips or kink points.

An intersection point between a line segment and an element edge is identified only if their relative position is intersecting (see Figure 5.6). Depending on how many intersection points can be identified for a given crack segment:

- If there are 2 intersection points, then the segment in between them will be used as input to the triangulation algorithm.

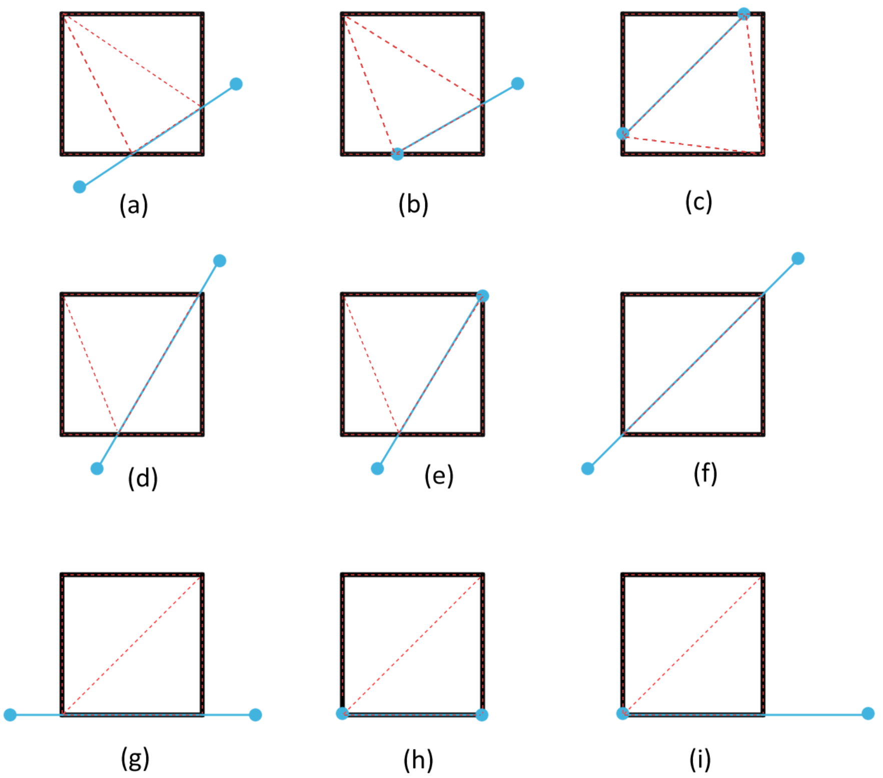


Fig. 5.9 Possible configurations of an element and a line segment: 2 intersection points.

- If there is 1 intersection point and a vertex of the crack segment is inside the element, then the segment between that vertex and the intersection point will be used as input to the triangulation algorithm.

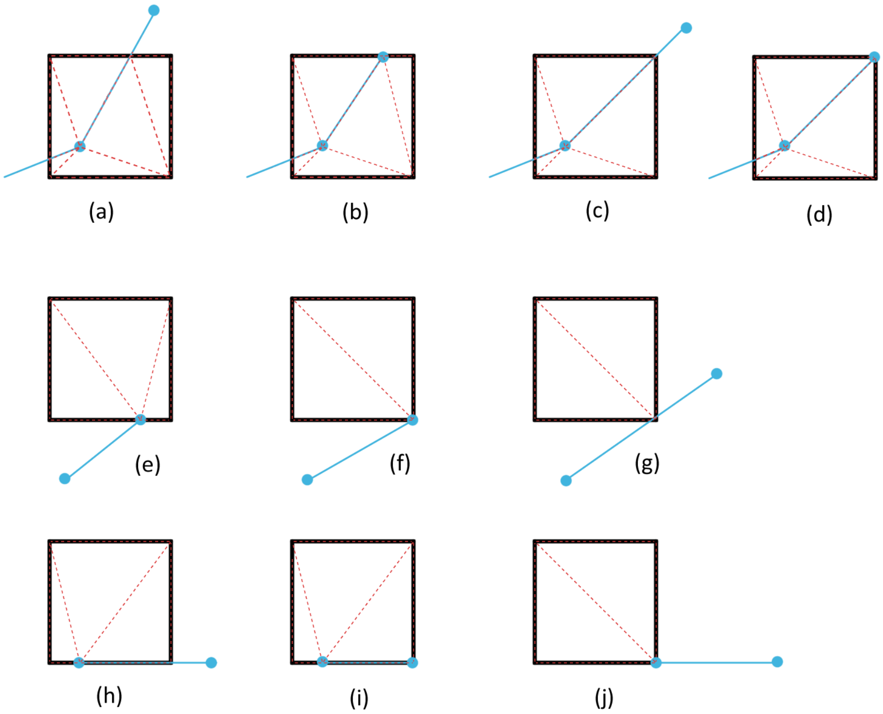


Fig. 5.10 Possible configurations of an element and a line segment: 1 intersection point.

- If there are 0 intersection points and both vertices of the crack segment lie inside the element, then the whole crack segment will be used as input to the triangulation algorithm.

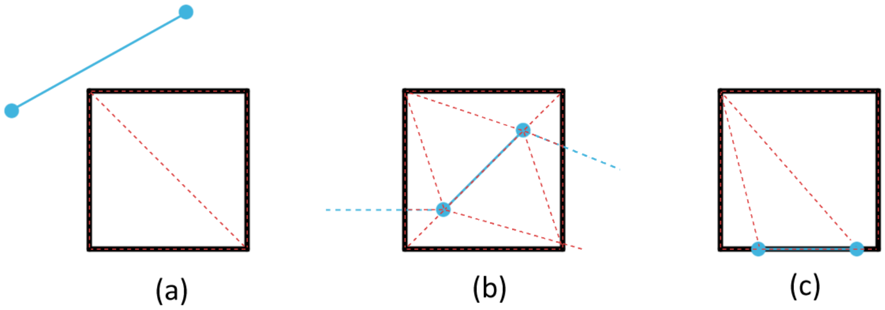


Fig. 5.11 Possible configurations of an element and a line segment: 0 intersection points.

Algorithm 5.2 describes the whole procedure for triangulating an element. As before, it is an inefficient $O(n_S)$ operation, since all segments of the polyline are iterated. A more efficient version could terminate the iterations, once a batch of consecutive intersecting elements is completed. That only works when the crack does not intersect the same element after that first batch, which is reasonable for most problems with an sufficiently fine mesh. However even then, the complexity would remain $O\left(n_S\right)$. Especially for elements not intersected by the crack, all segments are iterated.

Algorithm 5.2 Create triangular mesh for an element.

```
Input: $V$, $S$ are lists containing the $n_S+1$ vertices and $n_S$ segments of the crack polyline. Segment $S[i]$ has $V[i]$, $V[i+1]$ as vertices.
Input: The element, its nodes and edges.
Let $M_P$ and $M_S$ be sets containing the points and segments that will be used as input to the constrained triangulation algorithm. They do not accept duplicate entries.
for each node of the element do
    Add the node to $M_P$.
for each edge of the element do
    Add the edge to $M_S$.
for i from 1 to $n_S + 1$ do
    if $V[i]$ is inside or on the boundary of the element (see Section 5.2.2) then
        Add the vertex $V[i]$ to $M_P$.
for i from 1 to $n_S$ do
    Let $M_I$ be the set of $n_{M_I}$ unique intersection points of segment $S[i]$ with the element edges.
    for each edge of the element do
        Find the intersection point of the segment $S[i]$ and the edge, if there is one.
        Add this intersection point to $M_I$ and $M_P$.
    if $n_{M_I} = 2$ then
        Add a new segment $\left(M_I[1], M_I[2]\right)$ to $M_S$.
    else if $n_{M_I} = 1$ then
        if $V[i] \in M_P$ AND $V[i] \neq M_I[1]$ then
            Add a new segment $\left(V[i], M_I[1]\right)$ to $M_S$.
        if $V[i+1] \in M_P$ AND $V[i+1] \neq M_I[1]$ then
            Add a new segment $\left(V[i+1], M_I[1]\right)$ to $M_S$.
    else                                        ▷ No intersection points found
        if $V[i] \in M_P$ AND $V[i+1] \in M_P$ then
            Add a new segment $(V[i], V[i+1])$ to $M_S$.
Launch the constrained triangulation algorithm with $M_P$ and $M_S$ as input.
Output: The vertices $\left(P_1, P_2, P_3\right)$ of all sub-triangles.
```

5.2.6 Geometry initialization

Initializing the crack description given a starting line segment is trivial:

1. Initialize the collections that store the vertices, segments, crack propagation angles and any other data that will be stored throughout the propagation analysis.

2. Add the initial line segment V_1V_2 and its vertices to the respective collections.

3. Calculate the orientation of the initial segment

$$a = atan2\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{5.16}$$

   and store the local coordinate systems at the crack tips, defined by

   - The tip coordinates: V_1 and V_2 for the start and end tip respectively.
   - The counter-clockwise angle from global x to local $\tilde{x}$ axis: $a_1 = a+\pi$ and $a_2 = a$ for the start and end tip respectively.

5.2.7 Geometry update

Updating the crack geometry is also very straightforward and efficient for explicit crack descriptions. Algorithm 5.3 describes the procedure for propagating the polyline from a given crack tip.

Algorithm 5.3 Update the explicit crack geometry.

```
**Input**: $V$, $S$, $\Theta$ are *lists* containing the $n_S + 1$ vertices, $n_S$ segments and $n_S - 1$ propagation angles of the crack polyline. Segment $S[i]$ has $V[i]$, $V[i+1]$ as vertices. Its propagation angle from the previous segment is $\Theta[i-1]$, since there is no such angle for the first segment.
**Input**: $a_1$, $a_2$ are the counter-clockwise angles from global $x$ to local $\tilde{x}$ axes of the coordinate systems defined at the start and end tip respectively.
**Input**: $\theta c$ is the crack propagation direction, defined as the counter-clockwise angle from the extension of the segment containing the crack tip to the new segment.
**Input**:$\Delta L$ is the crack propagation length.
if propagating from the start tip then
    $a_1 \leftarrow a_1 + \theta_c$
    $\mathbf{\Delta x} \leftarrow \begin{bmatrix} \Delta L \cos a_1 & \Delta L \sin a_1 \end{bmatrix}^T$
    $\mathbf{x_{new}} \leftarrow V[1] + \mathbf{\Delta x}$
    Add the new tip $\mathbf{x_{new}}$ to the start of the vertices *list* $V$.
    Add a new segment $\left(\mathbf{x_{new}}, V[1]\right)$ to the start of the segments *list* $S$.
    Add the opposite propagation angle $-\theta_c$ to the start of the propagation angles $\Theta$.
    Overwrite the local system at the start tip with the new one: $\left(\mathbf{x_{new}}, a_1\right)$.
else
    $a_2 \leftarrow a_2 + \theta_c$
    $\mathbf{\Delta x} \leftarrow \begin{vmatrix} \Delta L \cos a_2 & \Delta L \sin a_2 \end{vmatrix}^T$
    $\mathbf{x_{new}} \leftarrow V[n_S + 1] + \mathbf{\Delta x}$
    Add the new tip $\mathbf{x_{new}}$ to the end of the vertices *list* $V$.
    Add a new segment $\left(V[n_S + 1], \mathbf{x_{new}}\right)$ to the end of the segments *list* $S$.
    Add the the propagation angle $\theta_c$ to the end of the propagation angles $\Theta$.
    Overwrite the local system at the end tip with the new one: $\left(\mathbf{x_{new}}, a_2\right)$.
```

5.3 Implicit crack geometry description

In this section a crack geometry description based on the Level Set Method (LSM) will be presented. LSM was originally proposed in [15] to track moving interfaces. In this technique, a time dependent function $\varphi(\mathbf{x}, t)$ is defined over the domain and the interface of interest $\gamma(t)$ coincides with the contour $\varphi(\mathbf{x}, t) = 0$, as illustrated in Figure 5.12:

$$\gamma(t) = \left\{ \mathbf{x} \in \mathbb{R}^2 \, : \, \varphi(\mathbf{x}, t) = 0 \right\} \tag{5.17}$$

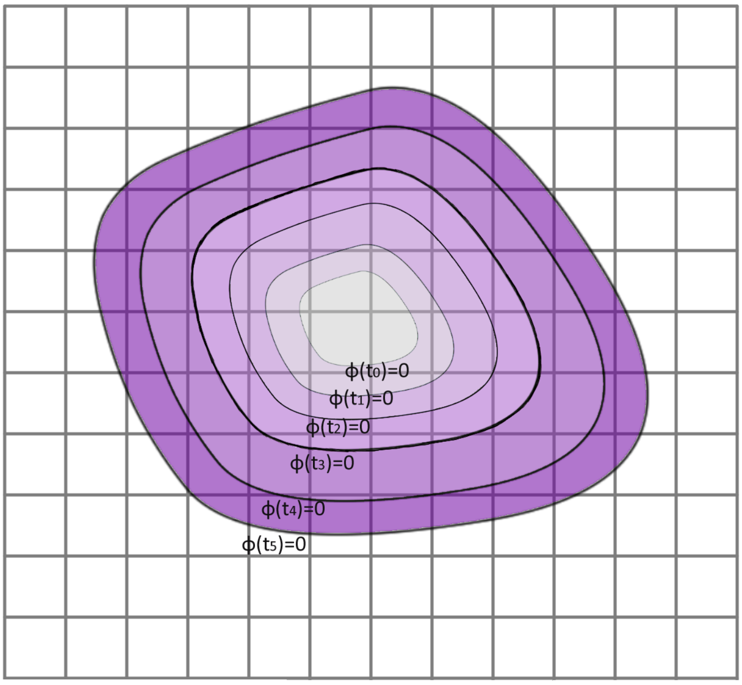


Fig. 5.12 The Level Set Method for tracking a moving interface: the interface of interest corresponds to $\varphi(\mathbf{x}, t) = 0$ at any time.

The level set function $\varphi(\mathbf{x}, t)$ is evaluated at the nodes of a fixed mesh and interpolated inside each element, if necessary. The unit normal vector to the interface can be evaluated as

$$\mathbf{n} = \frac{\nabla \varphi}{\|\nabla \varphi\|} \tag{5.18}$$

The nodal level sets are initialized by calculating the signed distance of each node to an initial curve $\gamma(t_0)$

$$\varphi(\mathbf{x}, t_0) = \|\mathbf{x} - \mathbf{x}^*\| \; \mathrm{sign}(\mathbf{n} \cdot (\mathbf{x} - \mathbf{x}^*)) \tag{5.19}$$

where $\mathbf{x}^*$ is the point on $\gamma(t_0)$ that is closest to $\mathbf{x}$. To update the interface when it moves with a velocity $\mathbf{v}$ normal to the interface, the nodal level sets are updated by solving the Hamilton-Jacobi equation

$$\frac{\vartheta \varphi}{\vartheta t} + \mathbf{v} \, \|\nabla \varphi\| = 0 \tag{5.20}$$

5.3.1 LSM for crack geometries

The Level Set Method provides an efficient and elegant technique to describe the crack geometry. It couples naturally with XFEM, since the level set function $\varphi(\mathbf{x})$ is evaluated over a fixed mesh and is also used in XFEM as the signed distance. In addition, it can be extended to 3D problems more easily than explicit crack descriptions. Using LSM for modelling the geometry of cracks was first proposed by Stolarska et al. [24] and has been established as the most common approach ever since. To represent the crack interface, the following level set functions are utilized, as illustrated in Figure 5.13:

- A level set function $\varphi(\mathbf{x})$ to describe the crack body. It extends outwards in a direction tangential to the crack at each crack tip. It is evaluated as the signed distance of a point to the crack interface Γ_c or its extension.
- A level set function $\psi(\mathbf{x})$ to describe a line perpendicular to Γ_c at the crack tip. It is evaluated as the signed distance of a point to that orthogonal line. The sign of $\psi(\mathbf{x})$ is negative in the half plane that contains the crack interface and positive in the half plane that contains its extension.

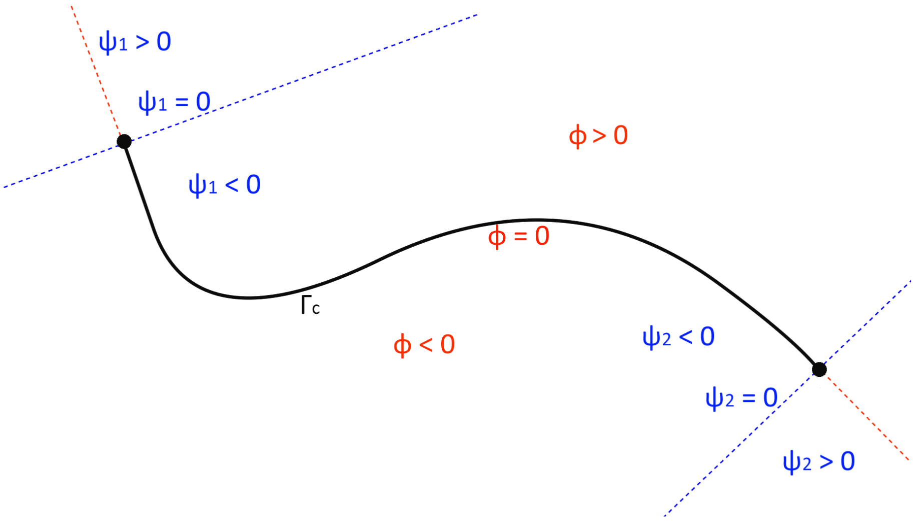


Fig. 5.13 LSM representation of a crack interface.

The crack interface is defined as the locus of points with the property

$$\Gamma_c = \left\{\mathbf{x} \in \mathbb{R}^2 \,:\, \varphi(\mathbf{x}, t) = 0 \textbf{ AND } \psi(\mathbf{x}) \leq 0\right\} \tag{5.21}$$

The crack tip is implied as the intersection of the two zero level sets

$$\varphi\left(\mathbf{x}_T\right) = 0 \textbf{ AND } \psi\left(\mathbf{x}_T\right) = 0 \tag{5.22}$$

For interior cracks two level set functions $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$ are needed, one for each crack tip. To use the same equations as for the edge crack case above, a unified level set function can be defined:

$$\psi(\mathbf{x}) = \max\left(\psi_1(\mathbf{x}), \psi_2(\mathbf{x})\right) \tag{5.23}$$

Regarding the implementation of the method, the following quantities need to be stored and updated at each iteration of the crack propagation analysis:

- The ”body” level set $\varphi(\mathbf{x})$ at each node.
- The ”tip” level sets $\psi_j(\mathbf{x})$ at each node. For interior cracks it is more convenient to store both $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$, instead of $\psi(\mathbf{x})$, in order to differentiate between the two crack tips, even though it requires more memory.
- The current global coordinates of the crack tips $\mathbf{x}_{T_1}$, $\mathbf{x}_{T_2}$. While this technically deviates from a pure implicit crack description, crack tip coordinates simplify many of the procedures that will be presented afterwards, are easy to calculate and require negligible memory.
- Likewise, the local coordinate systems defined at each crack tip are also convenient to handle explicitly. Each one is fully described by its origin, that is the already stored crack tip, and a counter-clockwise angle from global x to local $\tilde{x}$ axis.

5.3.2 Level set and signed distance functions

When solving crack propagation problems with XFEM, the signed distance from a point P to the crack interface is needed for the Heaviside enrichment function. By describing the crack geometry with LSM, this signed distance is identical to the ”body” level set function $\varphi(\mathbf{x})$, which is calculated at each node of the mesh and stored. Two cases can be identified:

- P is a node. In this case, $\varphi(\mathbf{x})$ is usually already stored for this node and only needs to be retrieved from the appropriate data structure. The calculation of $\varphi(\mathbf{x})$ for the node is performed when initializing or updating the crack geometry description. For more details see Sections 5.3.6 and 5.3.7.

- P is an integration point inside a given element. In the elements' interior $\varphi(\mathbf{x})$ can be interpolated using the nodal level set values and the shape functions of that element. For an isoparametric finite element and an integration point $\boldsymbol{\xi}$:

$$\varphi(\mathbf{x}(\boldsymbol{\xi})) = \sum_{i \in M_{nodes}} N_i(\boldsymbol{\xi}) \cdot \varphi_i \tag{5.24}$$

where M_{nodes} is the set of the element's nodes and $N_i(\boldsymbol{\xi})$ are the standard shape functions of the element. Compared to explicit crack descriptions (see Section 5.2.1), LSM provides a way to calculate the signed distances of integration points, that is very simple, efficient and easy to extend to 3D problems.

The ”tip” level set functions $\psi_j(\mathbf{x})$ can also be interpolated inside each element

$$\psi_j(\mathbf{x}(\boldsymbol{\xi})) = \sum_{i \in M_{nodes}} N_i(\boldsymbol{\xi}) \cdot \psi_{ji} \tag{5.25}$$

It is important to note that the crack geometry depends on the order of the finite elements. For first order elements, the shape functions and hence $\varphi(\mathbf{x})$ are linear inside each element. Therefore only piecewise linear cracks can be described. For curved cracks higher order elements must be used. Nevertheless, the crack geometries covered thus far are indeed piecewise linear, since a propagation angle and length define the update at each iteration of the analysis.

5.3.3 Detecting tip and intersected elements

The LSM crack description also provides a straightforward way to determine if an element is intersected by the crack or it contains the crack tip, in order to enrich its nodes with the appropriate function. An example is given in Figure 5.14.

- Find the minimum and maximum nodal level sets of the element:

$$\varphi^{min}, \varphi^{max}, \psi_j^{min}, \psi_j^{max}, \psi^{min}, \psi^{max}$$

- The element contains the crack tip j if

$$\varphi^{min} \cdot \varphi^{max} \leq 0 \;\; \textbf{AND} \;\; \psi_j^{min} \cdot \psi_j^{max} \leq 0 \tag{5.26}$$

- The element is intersected by the the crack interface, *but does not contain the crack tip*, if

$$\varphi^{min} \cdot \varphi^{max} \leq 0 \;\; \textbf{AND} \;\; \psi^{max} < 0 \tag{5.27}$$

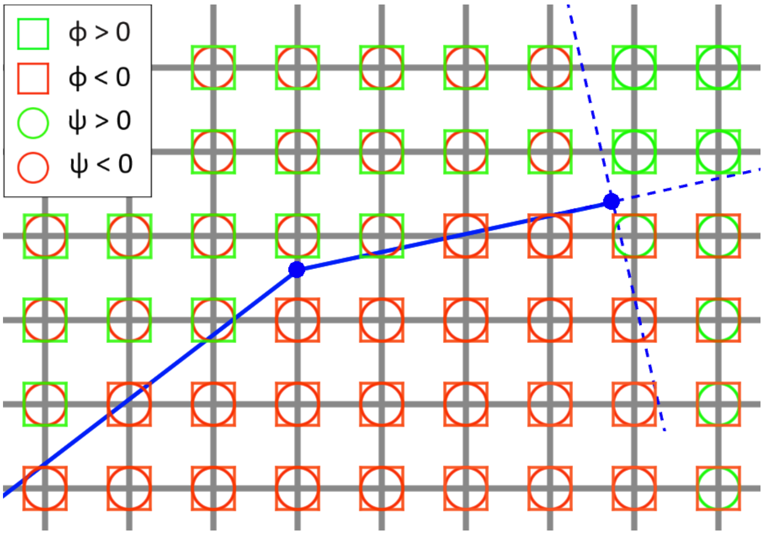


Fig. 5.14 Detecting tip and intersected elements using the nodal level sets.

5.3.4 Intersection with line segments

Before delving into the generation of a triangular mesh that conforms to the crack interface when the latter intersects an element, the intersection of the implicit crack with an edge of the element needs to be explained. Assuming first order finite elements, both the crack and the element edges are linear. In [25] a simple test was proposed: a line segment with vertices $\mathbf{x}_1$ and $\mathbf{x}_2$ is intersected by the crack if $\varphi_1 \cdot \varphi_2 < 0$, as illustrated in Figure 5.15. The intersection point can be located in global coordinates as

$$\mathbf{x}_I = \mathbf{x}_1 + \lambda \left(\mathbf{x}_2 - \mathbf{x}_1\right) \tag{5.28a}$$

$$\lambda = \frac{\left\|\mathbf{x}_I - \mathbf{x}_1\right\|}{\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|} = \frac{\varphi_I - \varphi_1}{\varphi_2 - \varphi_1} = \frac{-\varphi_1}{\varphi_2 - \varphi_1} \in (0,1) \tag{5.28b}$$

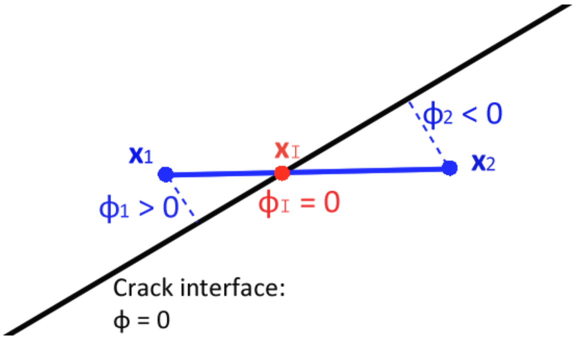


Fig. 5.15 Intersection of a line segment with the implicit crack interface.

Considering that the crack interface is not an infinite line, the above criterion must be adapted to account for the crack tip j. In this thesis, the following analysis is proposed to accurately determine the relative positions of a line segment and the crack interface:

- If $\varphi_1 \cdot \varphi_2 > 0$ the crack and segment are *disjoint* or *parallel* as seen in Figure 5.16.

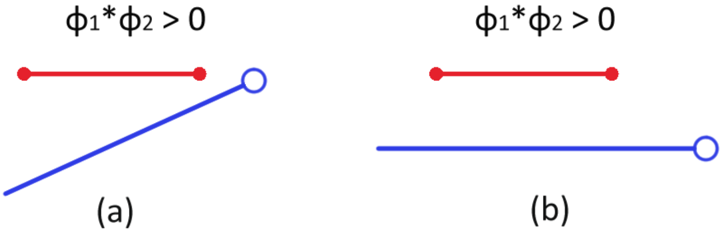


Fig. 5.16 Relative positions of a line segment and the crack interface: a) Disjoint, b) Parallel

- If $\varphi_1 = 0$ **AND** $\varphi_2 = 0$ the crack and segment are *collinear*.
  - If $\psi_{j1} > 0$ **AND** $\psi_{j2} > 0$ they are *collinear disjoint* (see Figure 5.17a).
  - If $\psi_{j1} < 0$ **AND** $\psi_{j2} < 0$ they *overlap* (see Figure 5.17b).
  - If $\psi_{j1} \cdot \psi_{j2} \leq 0$ they *overlap* and the tip j lies on the segment (see Figure 5.17c,d).

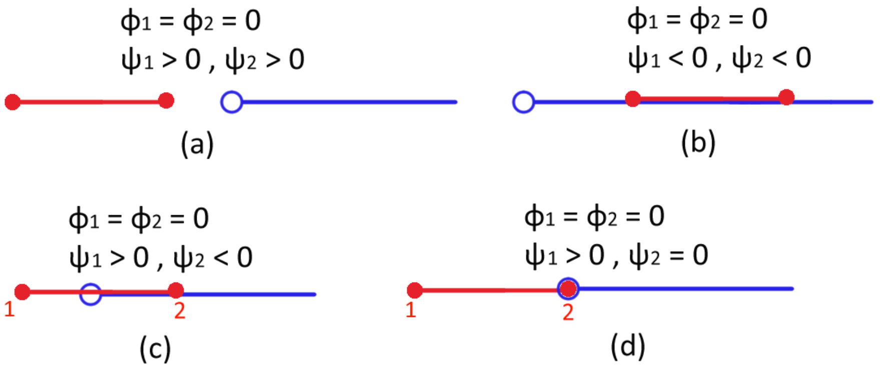


Fig. 5.17 Relative positions of a line segment and the crack interface: collinear

- The crack or its extension passes through only one node. Without loss of generality the case $\varphi_1 = 0$ **AND** $\varphi_2 \neq 0$ will be presented:
  - If $\psi_{j1} > 0$ the crack and segment are *disjoint* (see Figure 5.18a).
  - If $\psi_{j1} < 0$ the crack *intersects* the segment at node 1 (see Figure 5.18b).
  - If $\psi_{j1} = 0$ the crack *intersects* the segment at node 1, which coincides with the crack tip j (see Figure 5.18c).

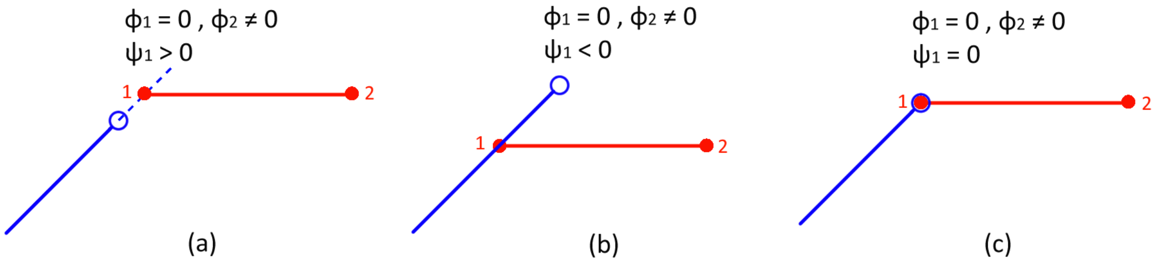


Fig. 5.18 Relative positions of a line segment and the crack interface: the crack passes through one node only

- If $\varphi_1 \cdot \varphi_2 < 0$ the crack or its extension intersects the segment at a point $\mathbf{x}_I$ that lies between the two vertices and can be obtained by (5.28). To differentiate between the various cases, ψ_{jI} must be calculated. Since ψ_j is always linear

$$\psi_{jI} = \psi_j\left(\mathbf{x}_I\right) = \psi_{j1} + \lambda\left(\psi_{j2} - \psi_{j1}\right) \tag{5.29}$$

where λ is the same as in (5.28). The following cases can be identified:

- If $\psi_{jI} > 0$ the crack and segment are *disjoint* (see Figure 5.19a, b).
- If $\psi_{jI} < 0$ the crack and segment *intersect* at $\mathbf{x}_I$ (see Figure 5.19c).
- If $\psi_{jI} = 0$ the crack and segment *intersect* at $\mathbf{x}_I$, which coincides with the crack tip j (see Figure 5.19d).

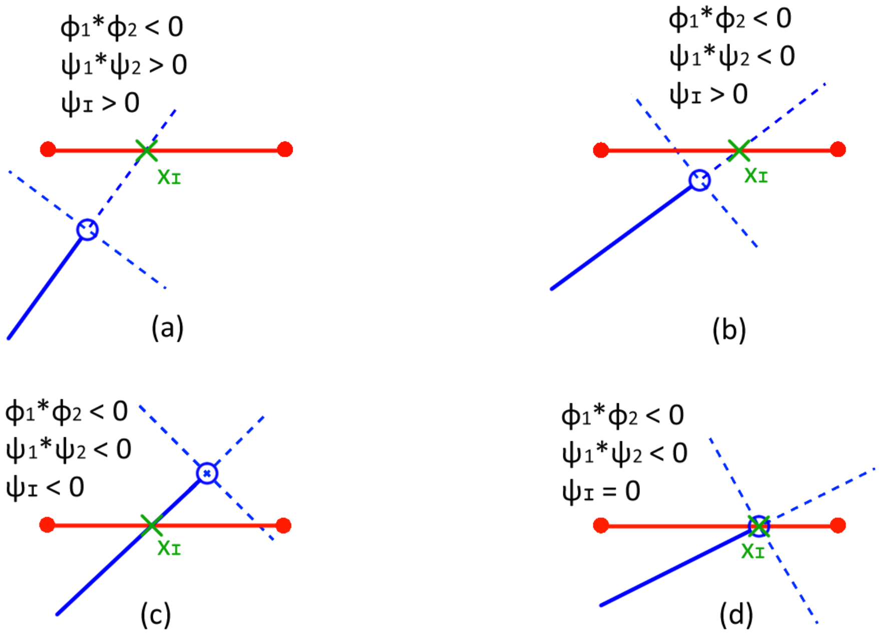


Fig. 5.19 Relative positions of a line segment and the crack interface: The crack intersects the segment

The following algorithm finds the intersection point of the crack interface with a line segment between vertices $\mathbf{x}_1$ and $\mathbf{x}_2$, according to the previous rules. No intersection point is returned, if they are collinear, parallel or disjoint.

Algorithm 5.4 Find the intersection point of a line segment between two nodes with the implicit crack interface, unless they are collinear, parallel or disjoint.

```
Input: $\Phi$, $\Psi_1$, $\Psi_2$ are *lists* containing the values of the "body", "start tip" and "end tip" level set functions at each node of the mesh respectively.
Input: $X$ are the global coordinates of each node of the mesh.
Input: $p_1$ and $p_2$ are the indices of the two nodes.
if $\Phi[p_1] * \Phi[p_2] < 0$ then
    $\lambda \leftarrow \frac{-\Phi[p_1]}{\Phi[p_2]-\Phi[p_1]}$
    $\mathbf{x}_I \leftarrow X[p_1] + \lambda * (X[p_2] - X[p_1])$
    $\psi_{1I} \leftarrow \Psi_1[p_1] + \lambda * (\Psi_1[p_2] - \Psi_1[p_1])$
    $\psi_{2I} \leftarrow \Psi_2[p_1] + \lambda * (\Psi_2[p_2] - \Psi_2[p_1])$
    if $\psi_{1I} \leq 0$ AND $\psi_{2I} \leq 0$ then
        return $\mathbf{x}_I$
else if $\Phi[p_1] = 0$ AND $\Phi[p_2] \neq 0$ then
    if $\Psi_1[p_1] \leq 0$ AND $\Psi_2[p_1] \leq 0$ then
        return $X[p_1]$
else if $\Phi[p_2] = 0$ AND $\Phi[p_1] \neq 0$ then
    if $\Psi_1[p_2] \leq 0$ AND $\Psi_2[p_2] \leq 0$ then
        return $X[p_2]$
else
    return null                    ▷ or any construct that denotes an invalid return value
```

5.3.5 Triangulation of elements

For first order finite elements the crack interface is linear in each element's region, therefore there cannot be any kink points inside the elements. While this makes LSM unable to model the real crack geometry exactly, it greatly simplifies the generation of a conforming triangular mesh. A constrained triangulation algorithm can be used to create the conforming mesh. Its input is:

- The element nodes and edges (line segments).
- The intersection points of the crack with the element edges, the crack tips and the segments defined by these points.

Algorithm 5.5 iterates over all edges of a given element, similar to the explicit crack description. The intersection points and the corresponding segments are aggregated in two *sets* M_P and M_S, which do not accept duplicate entries. The following notes are useful to better understand and validate the algorithm:

- If an intersection point coincides with a node or a crack tip, it can be safely added to M_P, which should automatically handle duplicate entries. Likewise, line segments that overlap with edges or other segments can be safely added to M_S.
- If the element is intersected by the crack, but does not contain any crack tips:
  - It usually has two intersection points, either of which could coincide with a node (Figure 5.20a, b, c).
  - It is possible that the crack passes through only one node, which is the single intersection point (Figure 5.20d).
  - In case the crack is collinear with an element edge, no intersection points will be found for that edge when Algorithm 5.4 is used. The two nodes of that edge will be correctly identified as intersection points, when the other edges are processed (Figure 5.20e). Nevertheless, adding them and the edge to M_P and M_S respectively, does not affect the triangulation.

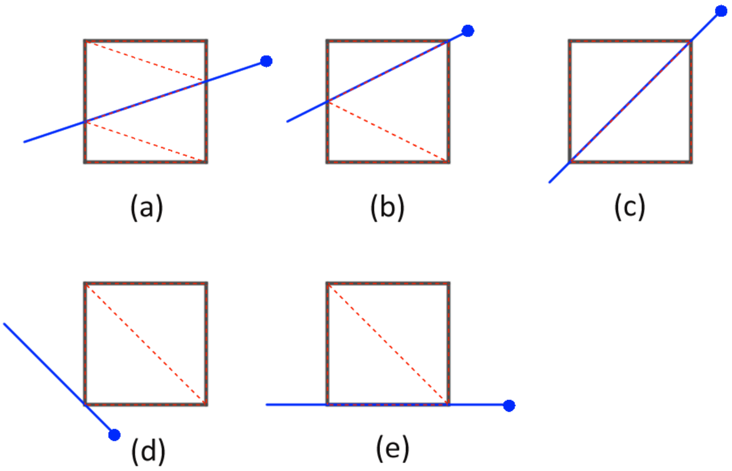


Fig. 5.20 Possible configurations of an elements and the crack interface: intersected element

- If the element contains a crack tip:
  - If the tip is an interior point of the element, only one intersection point will be identified, which could lie on a node or not (Figure 5.21a,b). In this case, the tip must be added to M_P and the segment between the tip and that intersection point must be added to M_S.
  - If the tip lies on the element's boundary (even on a node) and the crack does not intersect the element at any other point, then only the tip will be detected as an intersection point (Figure 5.21c,d). In this case, no line segment is defined.
  - If the tip lies on the element's boundary and the crack intersects the element at another point, then two intersection points will be detected. The tip, the other intersection point, both or none could coincide with a node (Figure 5.21e,f,g,h). Since the tip is one of the intersection points, adding it to M_P will have no adverse effect.
  - If the crack is collinear with an element edge, no intersection points will be found for that edge when Algorithm 5.4 is used. However, one or both nodes of that edge will be correctly identified as intersection points, when the other edges are processed (Figure 5.21i,j,k). In this case, it may be necessary to add the tip to M_P. The triangulation will not be affected by adding the line segment between the crack tip and the node that does not coincide with it to M_S.

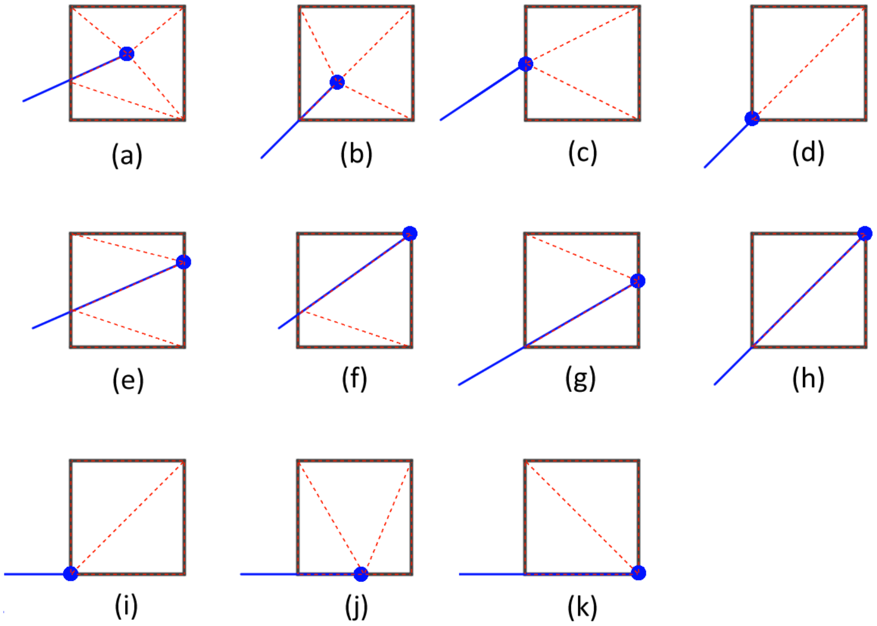


Fig. 5.21 Possible configurations of an elements and the crack interface: tip element

Algorithm 5.5 Create triangular mesh for an element.

```
Input: $\Phi$, $\Psi_1$, $\Psi_2$ are lists containing the values of the "body", "start tip" and "end tip" level set functions at each node of the mesh.
Input: $X$ is a list containing the global coordinates of the $n_{nodes}$ nodes of the element
Input: $\mathbf{x}_{T_1}, \mathbf{x}_{T_2}$ are the global coordinates of the crack tips.
Let $M_P$ and $M_S$ be sets containing the $n_{M_P}$ points and $n_{M_S}$ segments that will be used as input to the constrained triangulation algorithm. These data structures reject duplicate entries.
Initialize the flags: $isTip1 \leftarrow$ false, $isTip2 \leftarrow$ false, $isCut \leftarrow$ false
if the element contains the crack tip 1, according to (5.26) then
    $isTip1 \leftarrow$ true
else if the element contains the crack tip 2, according to (5.26) then
    $isTip2 \leftarrow$ true
else if the element is intersected by the crack, according to (5.27) then
    $isCut \leftarrow$ true
for i from 1 to $n_P$ do
    Add the node $X[i]$ to $M_P$.
    Specify this node $p_1 = i$ and the next one $p_2 = \left(i \text{ modulo } n_P\right) + 1$.
    Add the edge $\left(X\left[p_1\right], X\left[p_2\right]\right)$ to $M_S$.
    if $isTip1 =$ true OR $isTip2 =$ true OR $isCut =$ true then
        Find the intersection point of the edge between nodes $p_1$ and $p_2$, using Algorithm 5.4 and add it to $M_P$.
if $isTip1 =$ true then
    Add the first tip $T_1$ to $M_P$.
else if $isTip2 =$ true then
    Add the second tip $T_2$ to $M_P$.
Assert: $n_{M_P} \leq 2$
if $n_{M_P} = 2$ then Add the line segment between $M_P[1]$ and $M_P[2]$ to $M_S$.
Launch the constrained triangulation algorithm with $M_P$ and $M_S$ as input.
Output: The vertices $\left(P_1, P_2, P_3\right)$ of all sub-triangles.
```

Remarks:

- Algorithm 5.5 corresponds to interior cracks. For edge cracks, only one tip is involved and the algorithm needs to be adjusted slightly.

- If the element contains a crack tip, it is explicitly added to the intersection points. It is generally a good idea to store the coordinates of the current crack tips, as an explicit crack description would do.
- Regarding the assertion in line 19, the lack of kink points means that at most two intersection points are possible, including the crack tip. This is true even if an element edge is collinear with the crack interface, as was described in detail previously.

5.3.6 Geometry initialization

This section presents the initialization of the tracked quantities, given a starting line segment T_1T_2. In [24] the tip level sets are initialized as

$$\psi_j(\mathbf{x}) = \left(\mathbf{x} - \mathbf{x}_{T_j}\right) \cdot \hat{\mathbf{n}}_j \tag{5.30}$$

where $\hat{\mathbf{n}}_j$ is the tangent unit vector at the crack tip j, heading outwards. In other words $\psi_j(\mathbf{x})$ is the projection of the vector from the crack tip j to the point of interest onto $\hat{\mathbf{n}}_j$.

This thesis proposes the use of a local coordinate system for the line segment T_1T_2, which was introduced in Section 5.2.1 and is illustrated in Figure 5.22. The two approaches are equivalent, but the coordinate system of the segment facilitates the computation of other necessary quantities as well:

1. The rotation angle of the segment's local coordinate system with respect to global x axis is
$$\alpha = atan2\left(\frac{y_{T_2} - y_{T_1}}{x_{T_2} - x_{T_1}}\right) \tag{5.31}$$
2. The tip coordinate systems are defined as $\left(\mathbf{x}_{T_1}, \alpha + \pi\right)$ and $\left(\mathbf{x}_{T_2}, \alpha\right)$
3. Convert the coordinates of each node from the global system $\mathbf{x}_i$ to the segment's local system $\hat{\mathbf{x}}_i$ using (5.1).
4. The "body" level set is calculated as the signed distance of the node to the segment
$$\varphi\left(\mathbf{x}_i\right) = \hat{y}_i \tag{5.32}$$

5. The ”tip” level sets are calculated as the projections of $\mathbf{x}_i - \mathbf{x}_{T_j}$ onto $\hat{\mathbf{n}}_j$

$$\psi_1\left(\mathbf{x}_i\right) = -\hat{x}_i \tag{5.33a}$$

$$\psi_2\left(\mathbf{x}_i\right) = \hat{x}_i - L \tag{5.33b}$$

where L is the length of the line segment.

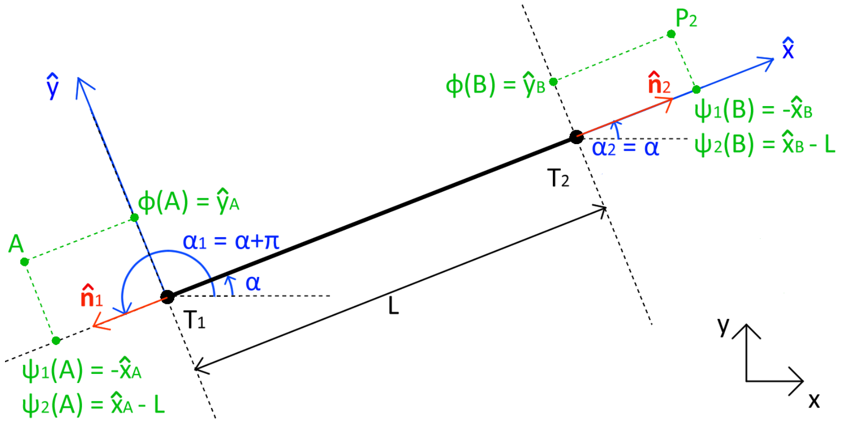


Fig. 5.22 Using the local coordinate system of a crack segment to calculate the level sets and other necessary quantities.

5.3.7 Geometry update

To update the level sets φ^t to φ^{t+1} and ψ_j^t to ψ_j^{t+1}, [24] uses the Hamilton-Jacobi equation (5.20). For crack propagation problems the displacement

$$\mathbf{\Delta x} = \Delta t\ \mathbf{v} = \mathbf{x}_{T_j}^{t+1} - \mathbf{x}_{T_j}^t \tag{5.34}$$

orthogonal to the front $\psi_j^{t+1} = 0$ is known instead of the velocity. The complete procedure is:

1. Rotate the current front $\psi_j^t = 0$ such that it is parallel to ψ_j^{t+1}, as depicted in Figure 5.23. To perform this, the rotated current level sets are calculated as

$$\psi_j^{rot} = \left(x - x_{T_j}^t\right) \frac{\Delta x}{\|\mathbf{\Delta x}\|} + \left(y - y_{T_j}^t\right) \frac{\Delta y}{\|\mathbf{\Delta x}\|} \tag{5.35}$$

2. Since the crack body does not move at all, most ”body” level sets should not be updated. Thus two regions can be identified

$$\Omega^{\text{update}} = \left\{ \mathbf{x} \in \mathbb{R}^2 \, : \, \psi_j^{rot} > 0 \right\} \tag{5.36a}$$

$$\Omega^{\text{no update}} = \left\{ \mathbf{x} \in \mathbb{R}^2 \, : \, \psi_j^{rot} \leq 0 \right\} \tag{5.36b}$$

The new level sets are then calculated as the signed distances of the nodes to the new segment in the direction of $\mathbf{\Delta x}$. In [24] the following is used

$$\begin{aligned} \mathbf{x} \in \Omega^{\text{no update}} : \quad & \varphi^{t+1} = \varphi^t \\ \mathbf{x} \in \Omega^{\text{update}} : \quad & \varphi^{t+1} = \pm \left| \left(\mathbf{x} - \mathbf{x}_{T_j}^t \right) \times \frac{\mathbf{\Delta x}}{\|\mathbf{\Delta x}\|} \right| \\ & \quad\;\;\, = \pm \left| \left(x - x_{T_j}^t \right) \frac{\Delta y}{\|\mathbf{\Delta x}\|} - \left(y - y_{T_j}^t \right) \frac{\Delta x}{\|\mathbf{\Delta x}\|} \right| \end{aligned} \tag{5.37}$$

3. Noting that by construction

$$\left\| \nabla \psi_j \right\| = 1 \tag{5.38}$$

the ”tip” level sets can now be updated using the adapted Hamilton-Jacobi equation

$$\psi_j^{t+1} = \psi_j^t - \|\mathbf{\Delta x}\| \tag{5.39}$$

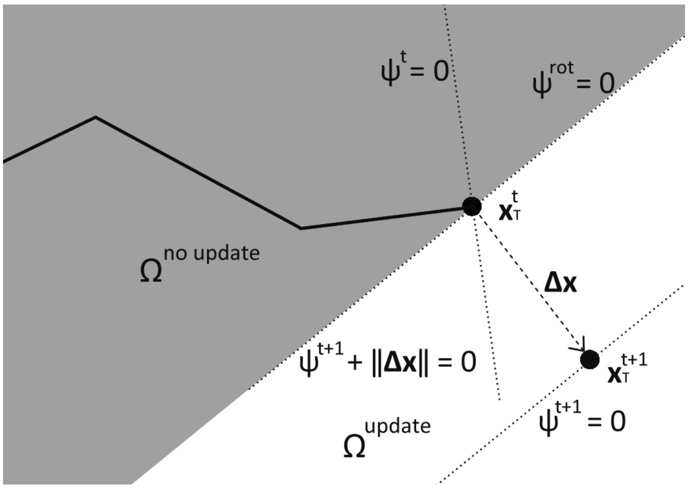


Fig. 5.23 Updating the level sets

For the purposes of this thesis, an equivalent procedure is proposed, using the local coordinate system of the new segment $\mathbf{\Delta x} = \mathbf{x}_{T_j}^{t+1} - \mathbf{x}_{T_j}^{t}$. Algorithm 5.6 describes the crack's propagation from tip j, taking into account that

- The end result of (5.35) and (5.39) is that new level set contour $\psi_j^{t+1} = 0$ is a line orthogonal to the crack interface at the new crack tip. The same could be achieved by using (5.33)b.
- The signed distance can be easily calculated using (5.32), which also provides the sign unlike (5.37).
- As shown in Figure 5.23, $\Omega^{\text{no update}}$ can also be defined as

$$\Omega^{\text{no update}} = \left\{ \mathbf{x} \in \mathbb{R}^2 \,:\, \psi_j^{t+1}(\mathbf{x}) + \|\mathbf{\Delta x}\| \leq 0 \right\} \tag{5.40}$$

Algorithm 5.6 Update the implicit crack geometry.

Input: X is a *list* containing the global coordinates of the n_{nodes} nodes of the element

Input: Φ, Ψ_j are *lists* containing the values of the ”body” and ”tip” j level set functions at each node of the mesh.

Input: $\mathbf{x}_j$ are the global coordinates of the crack tip j.

Input: a_j is the counter-clockwise angle from global x to local $\tilde{x}$ axis of the coordinate system defined at the crack tip j.

Input: θc is the crack propagation direction, defined as the counter-clockwise angle from the extension of the segment containing the crack tip to the new segment.

Input: ΔL is the crack propagation length.

```
$a_j \leftarrow a_j + \theta_c$
$\mathbf{\Delta x} \leftarrow \begin{bmatrix} \Delta L \cos a_j & \Delta L \sin a_j \end{bmatrix}^T$
$\mathbf{x}_j^{new} \leftarrow \mathbf{x}_j + \mathbf{\Delta x}$
Overwrite the local system at the tip $j$ with the new one: $\left(\mathbf{x}_j^{new}, a_j\right)$.
For the local coordinate system of the new segment $\left(\mathbf{x}_j, \mathbf{x}_j^{new}\right)$, find its rotation matrix
    $\mathbf{Q}$ and offset vector $\mathbf{b}$, according to (5.2) and (5.3).
for i from 1 to $n_{nodes}$ do
    $\hat{\mathbf{x}} \leftarrow \mathbf{Q} \cdot \mathbf{x} + \mathbf{b}$
    $\Psi_j[i] \leftarrow \hat{x} - \Delta L$
    if $\hat{x} > 0$ then
        $\Phi[i] \leftarrow \hat{y}$
```

5.3.8 Tip coordinate system

Instead of the explicit coordinate system at the crack tip, which is presented in detail in Appendix A.2, the level sets could be used for the operations necessary in XFEM, resulting in a purely implicit method. In [24] the following mapping between the global cartesian and local polar systems is proposed:

$$r = \sqrt{\varphi^2(\boldsymbol{\xi}) + \psi^2(\boldsymbol{\xi})} \tag{5.41a}$$

$$\theta = atan2\left(\frac{\varphi(\boldsymbol{\xi})}{\psi(\boldsymbol{\xi})}\right) \tag{5.41b}$$

and to determine the local cartesian axes, which are required in J-integral computations:

- Direction of local $\tilde{x}$ axis

$$\tilde{\mathbf{e}}_x = {}_{\mathbf{x}}\nabla\psi \tag{5.42}$$

- Direction of local $\tilde{y}$ axis

$$\tilde{\mathbf{e}}_y = \mathbf{e}_z \times {}_{\mathbf{x}}\nabla\psi \tag{5.43}$$

where $\mathbf{e}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

Elaborating on these, the Jacobian of the mapping in (5.41) is

$$\mathbf{J}_{NP} = \begin{bmatrix} r,_{\xi} & r,_{\eta} \\ \theta,_{\xi} & \theta,_{\eta} \end{bmatrix} = \begin{bmatrix} r,_{\varphi}\,\varphi,\xi + r,_{\psi}\,\psi,\xi & r,_{\varphi}\,\varphi,\eta + r,_{\psi}\,\psi,\eta \\ \theta,_{\varphi}\,\varphi,\xi + \theta,_{\psi}\,\psi,\xi & \theta,_{\varphi}\,\varphi,\eta + \theta,_{\psi}\,\psi,\eta \end{bmatrix} \tag{5.44}$$

where

$$r,_{\varphi} = \frac{\varphi}{r} \tag{5.45a}$$

$$r,_{\psi} = \frac{\psi}{r} \tag{5.45b}$$

$$\theta,_{\varphi} = \frac{\psi}{r^2} \tag{5.45c}$$

$$\theta,_{\psi} = \frac{-\varphi}{r^2} \tag{5.45d}$$

and the derivatives of the level sets functions with respect to the natural coordinates can be calculated using the shape functions

$$\varphi_{,\xi} = \sum_{i=1}^{n_{nodes}} N_{i,\xi}(\xi)\,\varphi_i \tag{5.46a}$$

$$\varphi_{,\eta} = \sum_{i=1}^{n_{nodes}} N_{i,\eta}(\xi)\,\varphi_i \tag{5.46b}$$

$$\psi_{,\xi} = \sum_{i=1}^{n_{nodes}} N_{i,\xi}(\xi)\,\psi_i \tag{5.46c}$$

$$\psi_{,\eta} = \sum_{i=1}^{n_{nodes}} N_{i,\eta}(\xi)\,\psi_i \tag{5.46d}$$

To compute the gradient of ψ needed in (5.42) and (5.43)

$$_{\mathbf{x}}\nabla\psi = \begin{bmatrix}\psi_{,x} & \psi_{,y}\end{bmatrix} = \begin{bmatrix}\psi_{,\xi} & \psi_{,\eta}\end{bmatrix} \cdot \mathbf{J}_{NG}^{-1} \tag{5.47}$$

where $\mathbf{J}_{NG}^{-1}$ is the Jacobian of the isoparametric mapping defined in Appendix A.1.2. Furthermore, the derivatives of the asymptotic tip functions B_e with respect to the global coordinates can be obtained according to (A.10)

$$\begin{bmatrix}B_{e,x} & B_{e,y}\end{bmatrix} = \begin{bmatrix}B_{e,\xi} & B_{e,\eta}\end{bmatrix} \cdot \mathbf{J}_{NG}^{-1} \tag{5.48}$$

The derivatives of Ψ_e with respect to the natural coordinates can in turn be calculated using the chain rule for the mapping from natural to local polar coordinates (5.41)

$$\begin{bmatrix}B_{e,\xi} & B_{e,\eta}\end{bmatrix} = \begin{bmatrix}B_{e,r} & B_{e,\theta}\end{bmatrix} \cdot \mathbf{J}_{PN}^{-1} = \begin{bmatrix}B_{e,r} & B_{e,\theta}\end{bmatrix} \cdot \mathbf{J}_{NP} \tag{5.49}$$

where

- The Jacobian $\mathbf{J}_{NP}$ was defined in (5.44).
- The derivatives of the asymptotic tip functions with respect to the local polar coordinates $B_{e,r}$ and $B_{e,\theta}$ where presented in Section 2.5.2.

5.3.9 LSM drawbacks

It is obvious that LSM provides a way to describe crack geometries that is elegant, efficient and couples intuitively with XFEM. However, it is not without complications, most of which stem from the fact that it cannot represent kinked cracks accurately. This section lists the most prominent sources of inaccuracy when using LSM. Most of these only apply for piecewise linear cracks and first order finite elements.

Incorrect nodal level sets

Using the level set update rule described in (5.37) may result in nodes having incorrect signed distances, which are equal to the level set values φ. When the crack propagates from $\mathbf{x}_T^t$ to $\mathbf{x}_T^{t+1}$, five regions can be identified between the lines $\psi^t = 0$, $\psi^{t+1} + \Delta L = 0$ and the bisector of the convex angle between the crack segments that meet on $\mathbf{x}_T^t$.

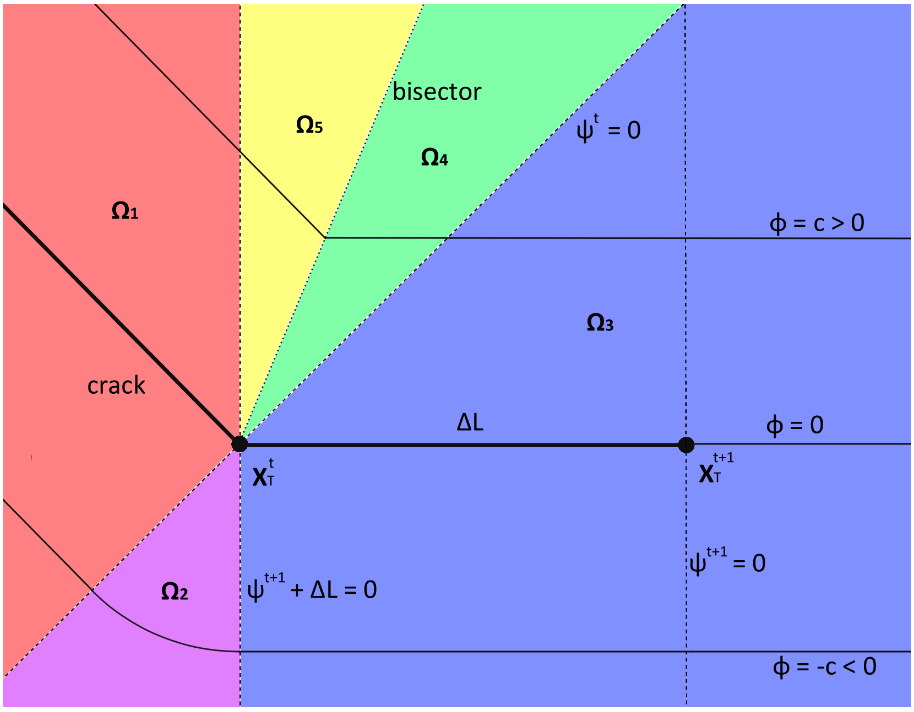


Fig. 5.24 Important regions during geometry update.

Figure 5.24 illustrates these regions. Observe that $\Omega^{\text{update}} = \Omega_3 \cup \Omega_4 \cup \Omega_5$ and $\Omega^{\text{no update}} = \Omega_1 \cup \Omega_2$. Table 5.1 presents the differences in the signed distance of the LSM crack description, when (5.37) is used, and the explicit crack description, which is considered exact.

Region	**LSM crack description:** **φ = signed distance to**	**Explicit crack description:** **φ = signed distance to**
Ω_1	segment $\mathbf{x}_T^t - \mathbf{x}_T^{t-1}$	segment $\mathbf{x}_T^t - \mathbf{x}_T^{t-1}$
Ω_2	segment $\mathbf{x}_T^t - \mathbf{x}_T^{t-1}$	vertex $\mathbf{x}_T^t$
Ω_3	segment $\mathbf{x}_T^{t+1} - \mathbf{x}_T^t$	segment $\mathbf{x}_T^{t+1} - \mathbf{x}_T^t$
Ω_4	segment $\mathbf{x}_T^{t+1} - \mathbf{x}_T^t$	segment $\mathbf{x}_T^{t+1} - \mathbf{x}_T^t$
Ω_5	segment $\mathbf{x}_T^{t+1} - \mathbf{x}_T^t$	segment $\mathbf{x}_T^t - \mathbf{x}_T^{t-1}$

Table 5.1 Signed distances using LSM vs the explicit crack description

Note that other implicit or hybrid crack geometry descriptions, such as the Vector Level Set method introduced in [26], use geometry update rules that produce the same nodal signed distances as the explicit crack description.

Inability to represent kink points exactly

As mentioned in Section 5.3.2, employing first order finite elements, means that the shape functions and consequently the zero level set $\varphi = 0$ are linear inside each element. Changes in direction are observed only on element edges. If the original crack interface is piecewise linear, then it cannot be represented exactly inside elements that should contain kink points, as shown in Figure 5.25.

This inaccuracy causes incorrect signed distances in elements containing kink points. If the signed distance of an integration point has the wrong sign, then the Heaviside enrichment also has the opposite sign and error is introduced in the XFEM approximation. This is made worse by the fact that nodal level sets are not always correct in elements near kink points, as described in the previous section.

Consider the case of calculating the level sets φ as the correct signed distances of the nodes to the crack interface, as depicted in Figure 5.24. Even then, using linear shape functions for elements in Ω_2 or intersected by the bisector would result in an inexact approximation of the signed distances of integration points.

On the other hand, refining the mesh does mitigate this issue. With a finer mesh, the zero level set curve $\varphi = 0$ matches the original crack geometry more closely (see Figure 5.25). Moreover, the fraction of elements with erroneous enrichments is smaller, which improves the quality of the discretized weak form.

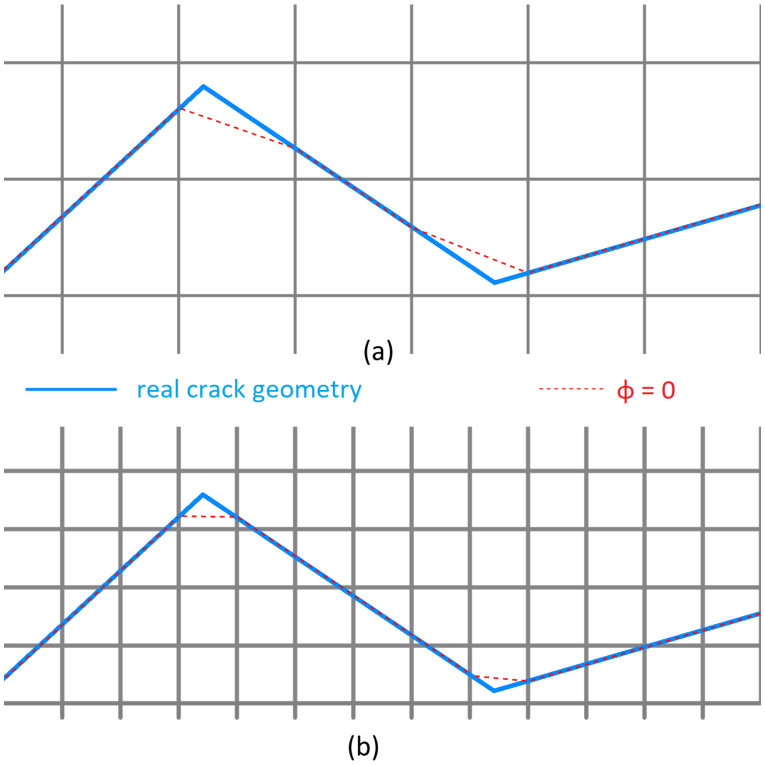


Fig. 5.25 Mesh dependency of the implicit crack description: a) Coarse mesh, b) Fine mesh

Incorrect intersections with element edges

Identifying the intersection point of the implicit crack geometry, that is the curve $\varphi = 0$, with the element edges is an important kernel, used extensively when generating a triangular mesh that conforms to the crack. To do so, (5.28) is applied, which assumes that φ is linear along the element edge. However, this is not the case for elements that would contain the kink points of the original crack. It is possible that the nodal level sets of an edge belonging to such an element are calculated as signed distances to different crack segments. In this case, φ is no longer linear along the element edge and (5.28) cannot locate the intersection point.

Examples:

- In Figure 5.26a, b the intersection point of $\varphi = 0$ with one edge of a quadrilateral or triangular element differ from the intersection point of the same edge with the actual crack.
- If the element containing the kink point is not intersected by the crack at consecutive edges, then the intersection points might be pinpointed accurately (see Figure 5.26c).

- An extreme case is portrayed in Figure 5.26d. The crack intersects the same edge twice, causing the level sets of its node to have the same sign. As a result $\varphi_1 \cdot \varphi_2 > 0$ and no intersection point can be determined at all by (5.28).

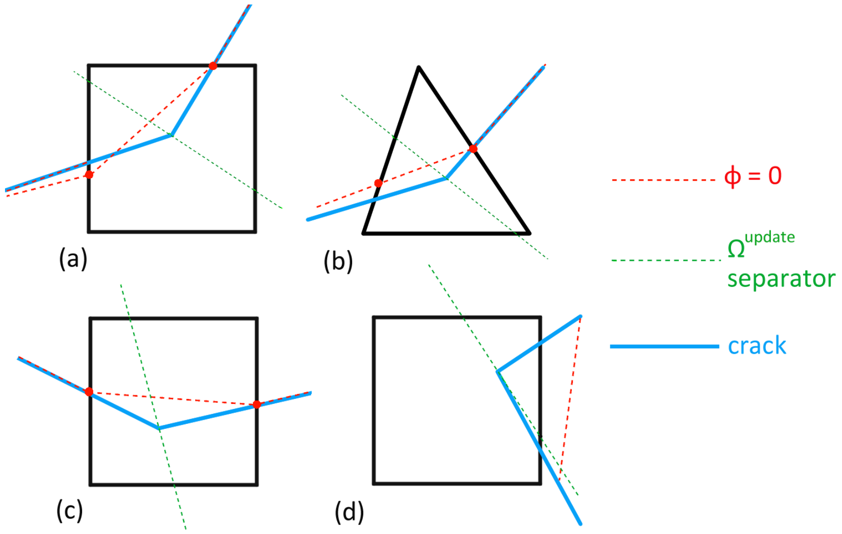


Fig. 5.26 Failure to correctly locate intersection points of crack and element, when nodal level sets are used.

Note that the intersection points cannot be located correctly for all elements where the nodal level sets are calculated as signed distances to different crack segments. Nevertheless, only the elements containing the kink points of the original crack are intersected by $\varphi = 0$. For the rest of these elements, no intersection points are identified anyway.

Incorrect detection of intersected and tip elements

To determine if an element is intersected by the crack interface or if it contains the crack tip, (5.27) and (5.26) respectively are proposed in [24]. However, these tests may fail in certain configurations of the mesh and the crack interface.

- In Figure 5.27a, element (5-6-9-8) is neither intersected by the crack nor does it contain the crack tip. Nevertheless

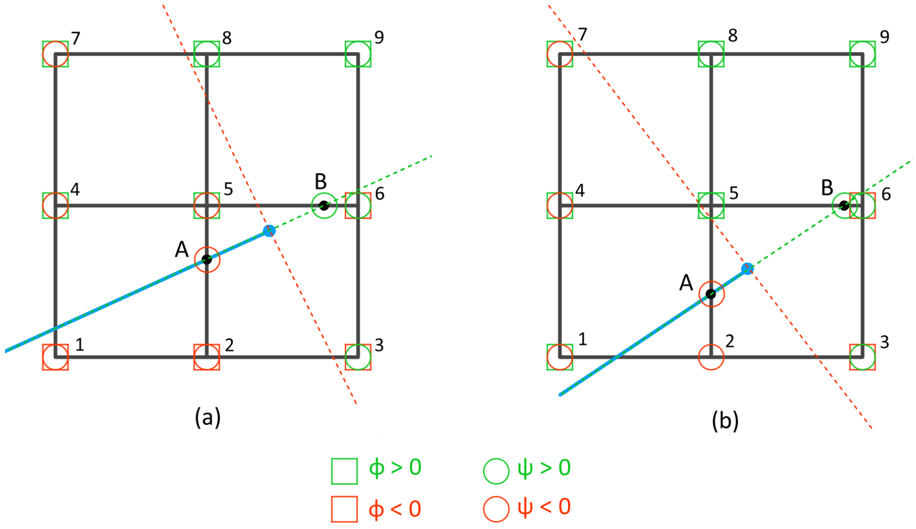
$$\begin{aligned}
\varphi^{min} &= \varphi\left(\mathbf{x}_6\right) < 0 \\
\varphi^{max} &= \varphi\left(\mathbf{x}_8\right) > 0 \\
\psi^{min} &= \psi\left(\mathbf{x}_5\right) < 0 \\
\psi^{max} &= \psi\left(\mathbf{x}_9\right) > 0
\end{aligned}$$

Thus $\varphi^{min} \cdot \varphi^{max} \leq 0$ **AND** $\psi_j^{min} \cdot \psi_j^{max} \leq 0$, so (5.26) will flag it as a tip element.

- In Figure 5.27a, element (1-2-5-4) is intersected by the crack, but does not contain the crack tip. Nevertheless

$$\begin{aligned} \varphi^{min} &= \varphi\left(\mathbf{x}_2\right) < 0 \\ \varphi^{max} &= \varphi\left(\mathbf{x}_4\right) > 0 \\ \psi^{min} &= \psi\left(\mathbf{x}_1\right) < 0 \\ \psi^{max} &= \psi\left(\mathbf{x}_5\right) > 0 \end{aligned}$$

Thus $\varphi^{min} \cdot \varphi^{max} \leq 0$ **AND** $\psi_j^{min} \cdot \psi_j^{max} \leq 0$, so (5.26) will flag it as a tip element. At the same time (5.27) will not flag it as an intersected element.



(a)

(b)

Fig. 5.27 Configurations where an element is incorrectly flagged as containing the crack tip.

To remedy this, Algorithm 5.7 is proposed in this thesis as a novel criterion for detecting tip and intersected elements. It utilizes the intersection points between the element edges and the curve $\varphi = 0$ (see (5.28)). By examining the level sets ψ evaluated at the intersection points (see (5.29)), the element can be characterized correctly.

Algorithm 5.7 Determine if an element is intersected by the crack interface or if it contains the crack tip.

```
Input: $X$ is a *list* containing the global coordinates of the $n_{nodes}$ nodes of the element
Input: $\Phi$, $\Psi$ are *lists* containing the values of the "body" and "tip" level set functions of the $n_{nodes}$ nodes of the element.
Go over all nodes of the element and find $\varphi^{min}, \varphi^{max}, \psi^{min}, \psi^{max}$
if $\varphi^{min} \cdot \varphi^{max} > 0$ then
    return "standard"
else
    if $\psi^{min} > 0$ then
        return "standard"
    else if $\psi^{max} < 0$ then
        return "intersected"
    else
        Go over each element edge and apply (5.28) and (5.29) to determine two intersection points $\left(\mathbf{x}_A, \mathbf{x}_B\right)$ and their level sets $\left(\psi_A, \psi_B\right)$.
        if $\mathbf{x}_A = \mathbf{x}_B$ then
            if $\psi_A > 0$ then
                return "standard"
            else if $\psi_A < 0$ then
                return "intersected"
            else                                              ▷ $\psi_A = 0$
                return "tip"
        else
            if $\psi_A > 0$ AND $\psi_B > 0$ then
                return "standard"
            else if $\psi_A < 0$ AND $\psi_B < 0$ then
                return "intersected"
            else                                              ▷ $\psi_A \cdot \psi_B \leq 0$
                return "tip"
Output: "tip" if the element contains the crack tip, "intersected" if the element is intersected by the crack interface (but does not contain the tip), "standard" in all other cases.
```

Remarks:

- It is evident that Algorithm 5.7 accurately characterizes the elements in Figure 5.27.

- The (extremely unlikely) case of $\varphi = 0$ passing through only one node of the element is also covered in line 11.
- Algorithm 5.7 depends on the exact identification of intersection points between the element edges and $\varphi = 0$. As elaborated in Section 5.3.9, the intersection points of elements containing kink points, are unlikely to be located properly. To prevent this, it is recommended that the tip elements do not also contain kink points. Nevertheless, such configurations pose complications in many other aspects of XFEM and crack propagation and should be avoided in general. This can be achieved by selecting the element size to be sufficiently smaller than the crack propagation length.

Issues with using the level sets for the tip coordinate system

In Section 5.3.8, the level set functions are used instead of an explicit local coordinate system. To achieve this, φ replaces the local cartesian $\tilde{y}$ coordinate and ψ replaces the local cartesian $\tilde{x}$ coordinate. While $\psi = \tilde{x}$ is always correct, $\varphi \neq \tilde{y}$ for curved or kinked cracks in general, as illustrated in Figure 5.28. However, if φ is evaluated for elements whose nodes are closest to the crack segment that contains the crack tip, then $\varphi = \tilde{y}$ indeed. Therefore, the J-integral (see Section 3.5.2) and fixed enrichment area (see Section 2.4.2) radii should be selected such that they are sufficiently smaller than the crack propagation length (see Section 3.4.3).

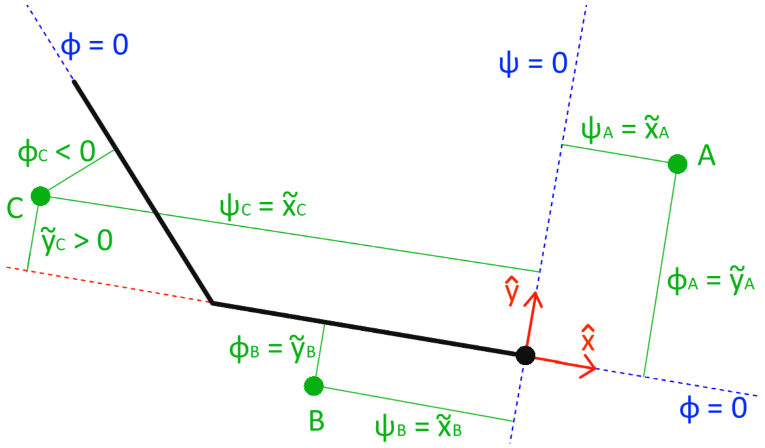


Fig. 5.28 Using the level sets as the local cartesian coordinates at a crack tip.

Another complication of using the level sets as local cartesian coordinates is rooted in the piecewise linear nature of the crack interface. As depicted in Figure 5.29, φ is discontinuous along the border between Ω^{update} and $\Omega^{no\ update}$, no matter how these are selected. In turn, this causes the polar coordinates (r, θ), calculated by (5.41), to also exhibit discontinuities.

As a result, spurious oscillations and decreased accuracy in the computation of the interaction integrals and the stress intensity factors may be observed. Errors can also be introduced in the XFEM approximation of the near tip displacement field. This is more probable when the fixed enrichment area scheme (see Section 2.4.2) is employed, since the local polar coordinates are more likely to be evaluated in the region where they are discontinuous.

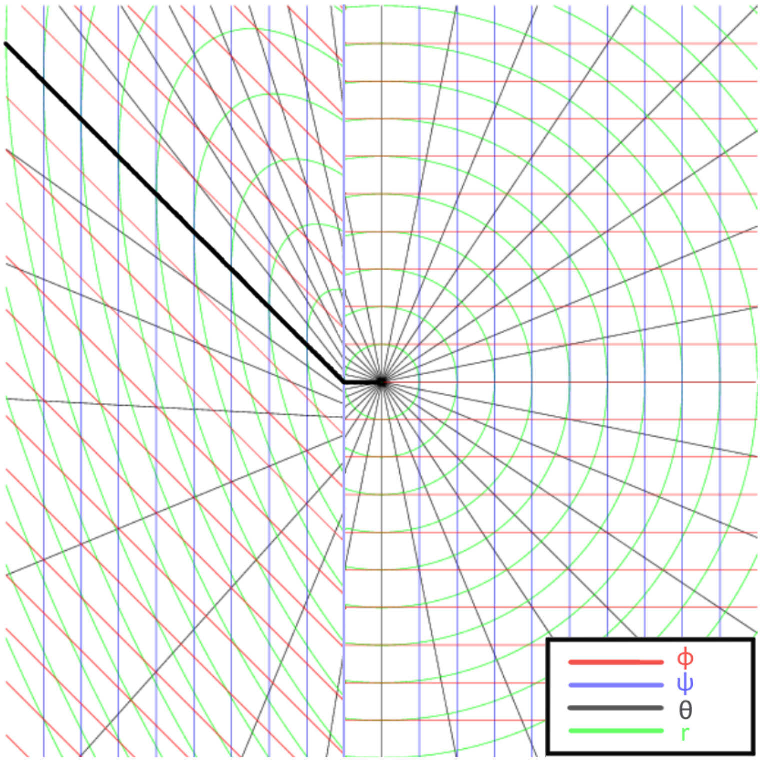


Fig. 5.29 Discontinuous level sets and local polar coordinates.

This issue was discovered by Duflot [30]. In the same paper, various update rules for the level sets are investigated and an acceptable one is proposed. In the numerical examples of this thesis, the issue is circumvented by simply using an explicit coordinate system at the crack tip.

Chapter 6

Numerical examples

6.1 Introduction

This chapter demonstrates the application of XFEM on a number of examples involving crack propagation.

- The first example is a somewhat detailed presentation of the various aspects of XFEM, when modeling a double cantilever beam with an edge crack. Intermediate results, like stiffness matrices and displacements, as well as the stress intensity factors (SIFs) are given.
- The second example examines the effect of various parameters, such as mesh size and J-integral radius, on the accuracy of the predicted SIFS for an infinite plate with a finite interior crack.
- The third example demonstrates a crack propagating from an I-beam with an edge crack above the fillet.
- The fourth example presents a crack growth simulation in a double cantilever beam. The performance of LSM is compared to the explicit crack description.

The generation of the finite element mesh is performed via the same *C#* code used for the rest of the analysis, for uniform and rectilinear meshes. On the other hand, unstructured meshes are generated using the open source software *Gmsh* ([35]).

6.2 XFEM modeling of a double cantilever beam

6.2.1 Problem formulation

The first example is a cantilever beam with an edge crack under pure Mode I loading. The model is depicted in Figure 6.1 and can be fully described by the following:

- The width of the beam $L = 20cm$ and the length is $3L$.
- The crack mouth is placed at (60, 10) and the crack tip at the center (30, 10).
- The left edge is fixed: the x and y degrees of freedom of the nodes along $x = 0$ are constrained.
- A displacement of $u_o = 0.5mm$ is prescribed on the top right and bottom right node: $u_{7y} = -u_{6y} = u_o$
- The beam is in plane strain condition.
- Material properties: Young's modulus $E = 2 \cdot 10^6 \frac{kg}{cm^2}$, Poisson ratio $\upsilon = 0.3$.

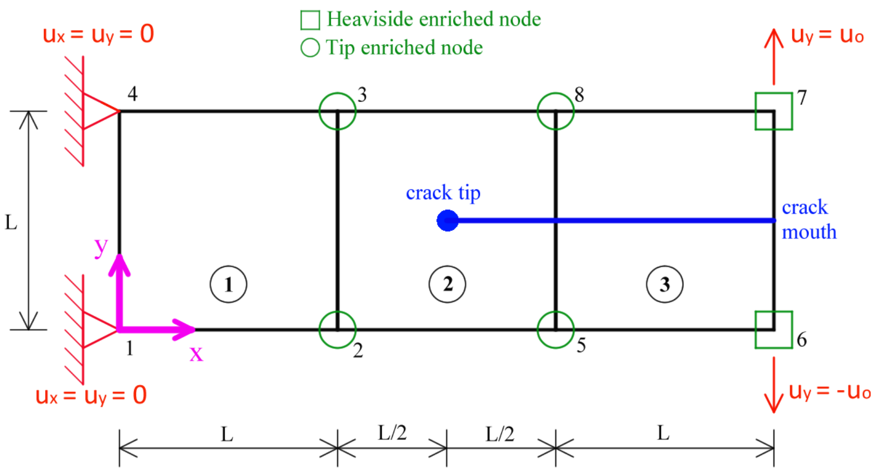


Fig. 6.1 A double cantilever beam.

At first the model is analyzed for a single step using XFEM under the following considerations:

- A uniform mesh of 3× Quad4 elements is used, exactly as in Figure 6.1.
- The nodes of the elements containing the crack tip are enriched with the asymptotic tip functions. The nodes of the elements intersected by the crack are enriched with the Heaviside function, unless they belong to the tip element too.

- The integration with sub-quads (see Section 4.2) is employed for the computation of the stiffness matrices. Enriched and tip blending elements are split into 8 × 8 sub-quads, and each sub-quad contains 2 × 2 integration points. For standard elements and blending elements with Heaviside enriched nodes only, the conventional Gauss-Legendre quadrature is used (2 × 2 integration points in this case).

The purpose of this example is to present in detail the steps involved in XFEM. The next two sections provide the stiffness matrices of each element and the displacements obtained after solving the linear system of the whole domain.

6.2.2 Stifness matrices

The stiffness matrix of element m has the form

$$\mathbf{K}^m = \begin{bmatrix} \mathbf{K}_{ss}^m & \left(\mathbf{K}_{es}^m\right)^T \\ \mathbf{K}_{es}^m & \mathbf{K}_{ee}^m \end{bmatrix} \tag{6.1}$$

The naming of freedom degrees is subject to the following rules

- u_{ij} are the actual displacements
- c_{ij} are artificial dofs introduced due to enrichment with the Heaviside function
- b_{ij}^e are artificial dofs introduced due to enrichment with the asymptotic tip functions
- i is the node index, $j = x$ or y
- e is the index of the tip enrichment function

For all three elements, the standard submatrix is the same

$$\mathbf{K}_{ss}^m = 10^6 \begin{array}{cc} \begin{array}{cccccccc} u_{1x} & u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{3y} & u_{4x} & u_{4y} \end{array} & \\ \begin{bmatrix} 1.1538 & 0.4808 & -0.7692 & 0.0962 & -0.5769 & -0.4808 & 0.1923 & -0.0962 \\ 0.4808 & 1.1538 & -0.0962 & 0.1923 & -0.4808 & -0.5769 & 0.0962 & -0.7692 \\ -0.7692 & -0.0962 & 1.1538 & -0.4808 & 0.1923 & 0.0962 & -0.5769 & 0.4808 \\ 0.0962 & 0.1923 & -0.4808 & 1.1538 & -0.0962 & -0.7692 & 0.4808 & -0.5769 \\ -0.5769 & -0.4808 & 0.1923 & -0.0962 & 1.1538 & 0.4808 & -0.7692 & 0.0962 \\ -0.4808 & -0.5769 & 0.0962 & -0.7692 & 0.4808 & 1.1538 & -0.0962 & 0.1923 \\ 0.1923 & 0.0962 & -0.5769 & 0.4808 & -0.7692 & -0.0962 & 1.1538 & -0.4808 \\ -0.0962 & -0.7692 & 0.4808 & -0.5769 & 0.0962 & 0.1923 & -0.4808 & 1.1538 \end{bmatrix} & \begin{array}{c} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ u_{4x} \\ u_{4y} \end{array} \end{array} \tag{6.2}$$

For element 1, the enriched-standard submatrix is

$$
\mathbf{K}_{es}^{1} = 10^{6}\left[\begin{array}{rrrrrrrrl}
u_{1x} & u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{3y} & u_{4x} & u_{4y} & \\
0.5706 & 0.4029 & -0.6656 & 0.1365 & -0.5930 & -0.0415 & 0.6880 & -0.4979 & b_{2x}^{1} \\
0.3585 & 0.0273 & 0.0011 & -0.3598 & 0.1414 & 0.0002 & -0.5010 & 0.3323 & b_{2y}^{1} \\
0.3116 & -0.0104 & -0.0183 & 0.1197 & -0.2368 & -0.4130 & -0.0565 & 0.3037 & b_{2x}^{2} \\
0.1352 & 0.5440 & -0.0623 & 0.4825 & -0.3776 & -0.5554 & 0.3047 & -0.4711 & b_{2y}^{2} \\
0.4763 & 0.2440 & -0.5966 & 0.0983 & -0.2021 & 0.0220 & 0.3223 & -0.3642 & b_{2x}^{3} \\
0.1857 & -0.0443 & 0.0425 & -0.3766 & 0.1379 & 0.1484 & -0.3660 & 0.2725 & b_{2y}^{3} \\
0.8779 & 0.7126 & -0.9909 & 0.1582 & -1.0410 & -0.0452 & 1.1540 & -0.8256 & b_{2x}^{4} \\
0.6708 & 0.0848 & -0.0903 & -0.4805 & 0.2599 & -0.1001 & -0.8404 & 0.4958 & b_{2y}^{4} \\
-0.6880 & -0.4979 & 0.5930 & -0.0415 & 0.6656 & 0.1365 & -0.5706 & 0.4029 & b_{3x}^{1} \\
-0.5010 & -0.3323 & 0.1414 & -0.0002 & 0.0011 & 0.3598 & 0.3585 & -0.0273 & b_{3y}^{1} \\
-0.0565 & -0.3037 & -0.2368 & 0.4130 & -0.0183 & -0.1197 & 0.3116 & 0.0104 & b_{3x}^{2} \\
-0.3047 & -0.4711 & 0.3776 & -0.5554 & 0.0623 & 0.4825 & -0.1352 & 0.5440 & b_{3y}^{2} \\
0.3223 & 0.3642 & -0.2021 & -0.0220 & -0.5966 & -0.0983 & 0.4763 & -0.2440 & b_{3x}^{3} \\
0.3660 & 0.2725 & -0.1379 & 0.1484 & -0.0425 & -0.3766 & -0.1857 & -0.0443 & b_{3y}^{3} \\
-1.1540 & -0.8256 & 1.0410 & -0.0452 & 0.9909 & 0.1582 & -0.8779 & 0.7126 & b_{3x}^{4} \\
-0.8404 & -0.4958 & 0.2599 & 0.1001 & -0.0903 & 0.4805 & 0.6708 & -0.0848 & b_{3y}^{4}
\end{array}\right] \tag{6.3}
$$

The enriched-enriched submatrix $\mathbf{K}_{ee}^{1}$ is split for viewing convenience into two submatrices by dividing its columns

$$
\mathbf{K}_{ee}^{1} = \begin{bmatrix} \mathbf{k}_{1}^{1} & \mathbf{k}_{2}^{1} \\ (16 \times 8) & (16 \times 8) \end{bmatrix} \tag{6.4}
$$

$$\mathbf{k}_1^1 = 10^6 \begin{array}{ccccccccc}
b_{2x}^1 & b_{2y}^1 & b_{2x}^2 & b_{2y}^2 & b_{2x}^3 & b_{2y}^3 & b_{2x}^4 & b_{2y}^4 & \\
1.2050 & -0.2530 & -0.1213 & 0.3409 & 0.7563 & -0.1004 & 2.0829 & -0.4456 & b_{2x}^1 \\
-0.2530 & 2.1231 & 0.3546 & 0.1312 & -0.0348 & 1.3000 & -0.4699 & 3.8834 & b_{2y}^1 \\
-0.1213 & 0.3546 & 1.9547 & 0.1789 & -0.2373 & 0.2316 & -0.1859 & 0.6445 & b_{2x}^2 \\
0.3409 & 0.1312 & 0.1789 & 0.9658 & 0.2325 & -0.0463 & 0.5926 & 0.3353 & b_{2y}^2 \\
0.7563 & -0.0348 & -0.2373 & 0.2325 & 0.5939 & -0.0007 & 1.2619 & -0.0390 & b_{2x}^3 \\
-0.1004 & 1.3000 & 0.2316 & -0.0463 & -0.0007 & 0.8505 & -0.1716 & 2.3459 & b_{2y}^3 \\
2.0829 & -0.4699 & -0.1859 & 0.5926 & 1.2619 & -0.1716 & 3.6368 & -0.8472 & b_{2x}^4 \\
-0.4456 & 3.8834 & 0.6445 & 0.3353 & -0.0390 & 2.3459 & -0.8472 & 7.1362 & b_{2y}^4 \\
-1.0852 & -0.0076 & -0.2473 & -0.4158 & -0.6900 & -0.0636 & -1.8860 & -0.0150 & b_{3x}^1 \\
0.0076 & -2.0107 & -0.4411 & -0.4721 & -0.1174 & -1.1835 & 0.0472 & -3.7348 & b_{3y}^1 \\
0.2473 & -0.4411 & 0.8163 & 0.0045 & -0.0229 & -0.2771 & 0.4167 & -0.7596 & b_{3x}^2 \\
-0.4158 & 0.4721 & -0.0045 & -0.0809 & -0.2477 & 0.3117 & -0.7086 & 0.7870 & b_{3y}^2 \\
0.6900 & -0.1174 & -0.0229 & 0.2477 & 0.3648 & -0.0336 & 1.2357 & -0.2262 & b_{3x}^3 \\
-0.0636 & 1.1835 & 0.2771 & 0.3117 & 0.0336 & 0.6626 & -0.1480 & 2.2229 & b_{3y}^3 \\
-1.8860 & -0.0472 & -0.4167 & -0.7086 & -1.2357 & -0.1480 & -3.2766 & -0.0863 & b_{3x}^4 \\
0.0150 & -3.7348 & -0.7596 & -0.7870 & -0.2262 & -2.2229 & 0.0863 & -6.9241 & b_{3y}^4
\end{array} \tag{6.5}$$

$$\mathbf{k}_2^1 = 10^6 \begin{array}{ccccccccc}
b_{3x}^1 & b_{3y}^1 & b_{3x}^2 & b_{3y}^2 & b_{3x}^3 & b_{3y}^3 & b_{3x}^4 & b_{3y}^4 & \\
-1.0852 & 0.0076 & 0.2473 & -0.4158 & 0.6900 & -0.0636 & -1.8860 & 0.0150 & b_{2x}^1 \\
-0.0076 & -2.0107 & -0.4411 & 0.4721 & -0.1174 & 1.1835 & -0.0472 & -3.7348 & b_{2y}^1 \\
-0.2473 & -0.4411 & 0.8163 & -0.0045 & -0.0229 & 0.2771 & -0.4167 & -0.7596 & b_{2x}^2 \\
-0.4158 & -0.4721 & 0.0045 & -0.0809 & 0.2477 & 0.3117 & -0.7086 & -0.7870 & b_{2y}^2 \\
-0.6900 & -0.1174 & -0.0229 & -0.2477 & 0.3648 & 0.0336 & -1.2357 & -0.2262 & b_{2x}^3 \\
-0.0636 & -1.1835 & -0.2771 & 0.3117 & -0.0336 & 0.6626 & -0.1480 & -2.2229 & b_{2y}^3 \\
-1.8860 & 0.0472 & 0.4167 & -0.7086 & 1.2357 & -0.1480 & -3.2766 & 0.0863 & b_{2x}^4 \\
-0.0150 & -3.7348 & -0.7596 & 0.7870 & -0.2262 & 2.2229 & -0.0863 & -6.9241 & b_{2y}^4 \\
1.2050 & 0.2530 & 0.1213 & 0.3409 & -0.7563 & -0.1004 & 2.0829 & 0.4456 & b_{3x}^1 \\
0.2530 & 2.1231 & 0.3546 & -0.1312 & -0.0348 & -1.3000 & 0.4699 & 3.8834 & b_{3y}^1 \\
0.1213 & 0.3546 & 1.9547 & -0.1789 & -0.2373 & -0.2316 & 0.1859 & 0.6445 & b_{3x}^2 \\
0.3409 & -0.1312 & -0.1789 & 0.9658 & -0.2325 & -0.0463 & 0.5926 & -0.3353 & b_{3y}^2 \\
-0.7563 & -0.0348 & -0.2373 & -0.2325 & 0.5939 & 0.0007 & -1.2619 & -0.0390 & b_{3x}^3 \\
-0.1004 & -1.3000 & -0.2316 & -0.0463 & 0.0007 & 0.8505 & -0.1716 & -2.3459 & b_{3y}^3 \\
2.0829 & 0.4699 & 0.1859 & 0.5926 & -1.2619 & -0.1716 & 3.6368 & 0.8472 & b_{3x}^4 \\
0.4456 & 3.8834 & 0.6445 & -0.3353 & -0.0390 & -2.3459 & 0.8472 & 7.1362 & b_{3y}^4
\end{array} \tag{6.6}$$

For element 2, the enriched-standard submatrix is

$$\mathbf{K}^2_{es} = 10^6 \begin{array}{rrrrrrrrl}
u_{1x} & u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{3y} & u_{4x} & u_{4y} & \\
-0.5761 & -0.0974 & 0.6166 & -0.4467 & 0.6529 & 0.4062 & -0.6934 & 0.1379 & b^1_{2x} \\
0.1447 & -0.0116 & -0.5074 & 0.1534 & 0.4466 & 0.2093 & -0.0839 & -0.3511 & b^1_{2y} \\
-0.3169 & 0.1912 & 0.0064 & -0.3055 & 0.2602 & 0.6160 & 0.0503 & -0.5017 & b^2_{2x} \\
0.0741 & -0.5989 & -0.1503 & -0.4879 & 0.6160 & 0.5640 & -0.5399 & 0.5227 & b^2_{2y} \\
-0.3465 & -0.0401 & 0.6299 & -0.3154 & 0.1996 & 0.0319 & -0.4830 & 0.3236 & b^3_{2x} \\
0.2205 & 0.3637 & -0.4575 & 0.6283 & 0.0323 & -0.3913 & 0.2047 & -0.6007 & b^3_{2y} \\
-1.0099 & -0.1968 & 0.9579 & -0.7050 & 1.1464 & 0.7570 & -1.0944 & 0.1448 & b^4_{2x} \\
0.0773 & -0.3142 & -0.6785 & 0.1323 & 0.7565 & 0.4690 & -0.1552 & -0.2870 & b^4_{2y} \\
0.6088 & 0.4692 & -1.5676 & 0.5747 & -0.8682 & 0.3841 & 1.8270 & -1.4280 & b^1_{5x} \\
0.1519 & -0.6876 & 0.5441 & -2.6683 & 0.8941 & 1.9723 & -1.5901 & 1.3835 & b^1_{5y} \\
0.6461 & 0.2835 & -0.4437 & 0.1139 & -0.4835 & -0.3163 & 0.2810 & -0.0811 & b^2_{5x} \\
0.3846 & 0.5336 & -0.1197 & 0.1748 & -0.1839 & -0.4397 & -0.0810 & -0.2687 & b^2_{5y} \\
0.7929 & 0.4607 & -0.8469 & 0.1655 & -0.6141 & -0.1115 & 0.6680 & -0.5146 & b^3_{5x} \\
0.4333 & 0.0818 & -0.0159 & -0.2705 & 0.0968 & -0.1469 & -0.5142 & 0.3356 & b^3_{5y} \\
0.6641 & 0.4752 & -0.3785 & -0.0551 & -0.6017 & -0.2305 & 0.3161 & -0.1896 & b^4_{5x} \\
0.6174 & 0.6414 & -0.3373 & 0.3583 & -0.0911 & -0.6384 & -0.1890 & -0.3613 & b^4_{5y} \\
-1.8270 & -1.4280 & 0.8682 & 0.3841 & 1.5676 & 0.5747 & -0.6088 & 0.4692 & b^1_{8x} \\
-1.5901 & -1.3835 & 0.8941 & -1.9723 & 0.5441 & 2.6683 & 0.1519 & 0.6876 & b^1_{8y} \\
0.2810 & 0.0811 & -0.4835 & 0.3163 & -0.4437 & -0.1139 & 0.6461 & -0.2835 & b^2_{8x} \\
0.0810 & -0.2687 & 0.1839 & -0.4397 & 0.1197 & 0.1748 & -0.3846 & 0.5336 & b^2_{8y} \\
0.6680 & 0.5146 & -0.6141 & 0.1115 & -0.8469 & -0.1655 & 0.7929 & -0.4607 & b^3_{8x} \\
0.5142 & 0.3356 & -0.0968 & -0.1469 & 0.0159 & -0.2705 & -0.4333 & 0.0818 & b^3_{8y} \\
-0.3161 & -0.1896 & 0.6017 & -0.2305 & 0.3785 & -0.0551 & -0.6641 & 0.4752 & b^4_{8x} \\
-0.1890 & 0.3613 & -0.0911 & 0.6384 & -0.3373 & -0.3583 & 0.6174 & -0.6414 & b^4_{8y} \\
0.6934 & 0.1379 & -0.6529 & 0.4062 & -0.6166 & -0.4467 & 0.5761 & -0.0974 & b^1_{3x} \\
-0.0839 & 0.3511 & 0.4466 & -0.2093 & -0.5074 & -0.1534 & 0.1447 & 0.0116 & b^1_{3y} \\
0.0503 & 0.5017 & 0.2602 & -0.6160 & 0.0064 & 0.3055 & -0.3169 & -0.1912 & b^2_{3x} \\
0.5399 & 0.5227 & -0.6160 & 0.5640 & 0.1503 & -0.4879 & -0.0741 & -0.5989 & b^2_{3y} \\
-0.4830 & -0.3236 & 0.1996 & -0.0319 & 0.6299 & 0.3154 & -0.3465 & 0.0401 & b^3_{3x} \\
-0.2047 & -0.6007 & -0.0323 & -0.3913 & 0.4575 & 0.6283 & -0.2205 & 0.3637 & b^3_{3y} \\
1.0944 & 0.1448 & -1.1464 & 0.7570 & -0.9579 & -0.7050 & 1.0099 & -0.1968 & b^4_{3x} \\
-0.1552 & 0.2870 & 0.7565 & -0.4690 & -0.6785 & -0.1323 & 0.0773 & 0.3142 & b^4_{3y}
\end{array} \tag{6.7}$$

$\mathbf{K}^2_{ee}$ is split for viewing convenience into four submatrices by dividing its columns

$$\mathbf{K}^2_{ee} = \begin{bmatrix} \mathbf{k}^2_1 & \mathbf{k}^2_2 & \mathbf{k}^2_3 & \mathbf{k}^2_4 \\ (32 \times 8) & (32 \times 8) & (32 \times 8) & (32 \times 8) \end{bmatrix} \tag{6.8}$$

$$
\mathbf{k}_1^2 = 10^6
\begin{array}{rrrrrrrrl}
b_{2x}^1 & b_{2y}^1 & b_{2x}^2 & b_{2y}^2 & b_{2x}^3 & b_{2y}^3 & b_{2x}^4 & b_{2y}^4 & \\
3.0196 & -0.5191 & -0.5159 & 0.4124 & 1.9575 & -0.7877 & 2.2076 & -0.3823 & b_{2x}^1 \\
-0.5191 & 4.4572 & 0.5011 & 0.9909 & -0.7970 & 2.9112 & -0.2305 & 5.2198 & b_{2y}^1 \\
-0.5159 & 0.5011 & 3.7414 & 0.0872 & -1.2502 & 0.1638 & 0.6077 & 1.0307 & b_{2x}^2 \\
0.4124 & 0.9909 & 0.0872 & 2.6385 & 0.1663 & 0.9452 & 1.0345 & 1.4335 & b_{2y}^2 \\
1.9575 & -0.7970 & -1.2502 & 0.1663 & 3.0158 & -0.6434 & 1.4736 & -0.5134 & b_{2x}^3 \\
-0.7877 & 2.9112 & 0.1638 & 0.9452 & -0.6434 & 4.6826 & -0.3256 & 2.8348 & b_{2y}^3 \\
2.2076 & -0.2305 & 0.6077 & 1.0345 & 1.4736 & -0.3256 & 4.8646 & -0.0839 & b_{2x}^4 \\
-0.3823 & 5.2198 & 1.0307 & 1.4335 & -0.5134 & 2.8348 & -0.0839 & 8.1103 & b_{2y}^4 \\
-1.9820 & 0.5447 & 2.4985 & -0.5732 & -1.8389 & -0.3746 & -1.7851 & 0.9694 & b_{5x}^1 \\
0.9324 & 2.2031 & -0.0012 & 3.1013 & -0.2271 & 0.2816 & 1.3215 & 2.2579 & b_{5y}^1 \\
0.2764 & 0.1857 & -1.2176 & -0.3677 & 0.9223 & 0.1610 & -0.4334 & 0.0251 & b_{5x}^2 \\
0.2259 & 1.4234 & -0.3536 & 0.5947 & 0.1449 & 2.3371 & 0.0597 & 0.7102 & b_{5y}^2 \\
-0.1235 & 0.5936 & 0.2680 & -0.4765 & 0.4327 & 0.3455 & -1.4820 & 0.6108 & b_{5x}^3 \\
0.7287 & 1.7169 & -0.5199 & 1.9425 & 0.3908 & 2.8466 & 0.5918 & 0.8339 & b_{5y}^3 \\
0.9031 & -0.2499 & -1.3446 & -0.3298 & 1.2432 & -0.1697 & 0.1809 & -0.5340 & b_{5x}^4 \\
-0.2251 & 2.7053 & -0.3477 & 0.0330 & -0.1557 & 2.0542 & -0.4898 & 2.7632 & b_{5y}^4 \\
0.0348 & -0.6363 & 0.2877 & 1.3010 & 0.2494 & -0.9987 & 1.5896 & 0.6412 & b_{8x}^1 \\
-0.5436 & -2.1444 & 1.8585 & -0.1977 & -0.8378 & -2.9134 & 1.0412 & -1.1022 & b_{8y}^1 \\
-0.1492 & 0.2335 & -0.4445 & -0.6037 & 0.1581 & 0.1274 & -0.5639 & 0.3323 & b_{8x}^2 \\
0.2973 & 1.2803 & -0.6033 & 1.0283 & 0.1280 & 1.0023 & 0.3344 & 1.1030 & b_{8y}^2 \\
-0.2621 & -0.0611 & 0.0066 & -0.5893 & 0.4067 & -0.3204 & -0.9755 & -0.2200 & b_{8x}^3 \\
0.0415 & 1.9645 & -0.5878 & 1.1667 & -0.3198 & 2.2895 & -0.2223 & 1.5055 & b_{8y}^3 \\
0.1608 & -0.4244 & -0.4132 & 0.4528 & -0.0727 & -0.2738 & 0.6353 & -0.8864 & b_{8x}^4 \\
-0.4664 & -1.8310 & 0.4480 & -1.7991 & -0.2713 & -1.3633 & -0.8866 & -1.6254 & b_{8y}^4 \\
-0.9639 & -0.0491 & -0.6394 & -0.4483 & -1.3047 & -0.0358 & -2.6648 & -0.5448 & b_{3x}^1 \\
0.0491 & -3.3390 & -0.2940 & -1.7389 & 0.0584 & -2.9983 & -0.5041 & -4.6157 & b_{3y}^1 \\
0.6394 & -0.2940 & 2.1119 & 0.0045 & -0.3063 & -0.4244 & 0.8268 & -0.3963 & b_{3x}^2 \\
-0.4483 & 1.7389 & -0.0045 & 0.7689 & -0.4526 & 1.4427 & -0.4192 & 2.0420 & b_{3y}^2 \\
1.3047 & 0.0584 & -0.3063 & 0.4526 & 1.4819 & 0.0467 & 2.0428 & 0.6081 & b_{3x}^3 \\
-0.0358 & 2.9983 & 0.4244 & 1.4427 & -0.0467 & 2.0738 & 0.5281 & 4.4724 & b_{3y}^3 \\
-2.6648 & 0.5041 & -0.8268 & -0.4192 & -2.0428 & 0.5281 & -3.0614 & 0.0740 & b_{3x}^4 \\
0.5448 & -4.6157 & -0.3963 & -2.0420 & 0.6081 & -4.4724 & -0.0740 & -5.6645 & b_{3y}^4
\end{array}
\qquad (6.9)
$$

$$\mathbf{k}_2^2 = 10^6 \begin{array}{cccccccc|c}
b^1_{5x} & b^1_{5y} & b^2_{5x} & b^2_{5y} & b^3_{5x} & b^3_{5y} & b^4_{5x} & b^4_{5y} & \\
\hline
-1.9820 & 0.9324 & 0.2764 & 0.2259 & -0.1235 & 0.7287 & 0.9031 & -0.2251 & b^1_{2x} \\
0.5447 & 2.2031 & 0.1857 & 1.4234 & 0.5936 & 1.7169 & -0.2499 & 2.7053 & b^1_{2y} \\
2.4985 & -0.0012 & -1.2176 & -0.3536 & 0.2680 & -0.5199 & -1.3446 & -0.3477 & b^2_{2x} \\
-0.5732 & 3.1013 & -0.3677 & 0.5947 & -0.4765 & 1.9425 & -0.3298 & 0.0330 & b^2_{2y} \\
-1.8389 & -0.2271 & 0.9223 & 0.1449 & 0.4327 & 0.3908 & 1.2432 & -0.1557 & b^3_{2x} \\
-0.3746 & 0.2816 & 0.1610 & 2.3371 & 0.3455 & 2.8466 & -0.1697 & 2.0542 & b^3_{2y} \\
-1.7851 & 1.3215 & -0.4334 & 0.0597 & -1.4820 & 0.5918 & 0.1809 & -0.4898 & b^4_{2x} \\
0.9694 & 2.2579 & 0.0251 & 0.7102 & 0.6108 & 0.8339 & -0.5340 & 2.7632 & b^4_{2y} \\
12.4321 & -4.8960 & 0.8240 & -1.6902 & 3.1138 & -3.1255 & 0.1337 & -0.5371 & b^1_{5x} \\
-4.8960 & 17.0180 & -1.3594 & 3.3215 & -2.4441 & 6.3657 & -0.4646 & 3.0697 & b^1_{5y} \\
0.8240 & -1.3594 & 2.6386 & 0.4593 & 1.6477 & 0.2535 & 1.8688 & 0.9391 & b^2_{5x} \\
-1.6902 & 3.3215 & 0.4593 & 3.9208 & 0.2142 & 4.6991 & 0.9085 & 2.7293 & b^2_{5y} \\
3.1138 & -2.4441 & 1.6477 & 0.2142 & 4.0634 & 0.0102 & 1.3863 & 0.6447 & b^3_{5x} \\
-3.1255 & 6.3657 & 0.2535 & 4.6991 & 0.0102 & 7.8957 & 0.6848 & 2.8035 & b^3_{5y} \\
0.1337 & -0.4646 & 1.8688 & 0.9085 & 1.3863 & 0.6848 & 3.0809 & 0.8593 & b^4_{5x} \\
-0.5371 & 3.0697 & 0.9391 & 2.7293 & 0.6447 & 2.8035 & 0.8593 & 4.6936 & b^4_{5y} \\
1.0711 & -0.4211 & -2.7033 & -2.2896 & -3.8357 & -2.5456 & -3.5475 & -2.1844 & b^1_{8x} \\
0.4211 & -0.4646 & -2.2317 & -5.1104 & -2.3531 & -6.4674 & -2.2519 & -5.7623 & b^1_{8y} \\
2.7033 & -2.2317 & 1.6267 & -0.0624 & 1.3253 & -0.2536 & 0.8906 & 0.2809 & b^2_{8x} \\
-2.2896 & 5.1104 & 0.0624 & 2.1235 & -0.0524 & 3.3908 & 0.3556 & 1.8426 & b^2_{8y} \\
3.8357 & -2.3531 & 1.3253 & 0.0524 & 3.0119 & -0.1624 & 1.3991 & 0.4183 & b^3_{8x} \\
-2.5456 & 6.4674 & 0.2536 & 3.3908 & 0.1624 & 5.7528 & 0.5289 & 2.8592 & b^3_{8y} \\
-3.5475 & 2.2519 & -0.8906 & 0.3556 & -1.3991 & 0.5289 & -0.3275 & 0.0171 & b^4_{8x} \\
2.1844 & -5.7623 & 0.2809 & -1.8426 & 0.4183 & -2.8592 & -0.0171 & -2.0410 & b^4_{8y} \\
0.0348 & 0.5436 & 0.1492 & 0.2973 & 0.2621 & 0.0415 & 0.1608 & 0.4664 & b^1_{3x} \\
0.6363 & -2.1444 & 0.2335 & -1.2803 & -0.0611 & -1.9645 & 0.4244 & -1.8310 & b^1_{3y} \\
-0.2877 & 1.8585 & -0.4445 & 0.6033 & 0.0066 & 0.5878 & 0.4132 & 0.4480 & b^2_{3x} \\
1.3010 & 0.1977 & 0.6037 & 1.0283 & 0.5893 & 1.1667 & 0.4528 & 1.7991 & b^2_{3y} \\
-0.2494 & -0.8378 & 0.1581 & -0.1280 & 0.4067 & 0.3198 & 0.0727 & -0.2713 & b^3_{3x} \\
-0.9987 & 2.9134 & -0.1274 & 1.0023 & 0.3204 & 2.2895 & -0.2738 & 1.3633 & b^3_{3y} \\
1.5896 & -1.0412 & 0.5639 & 0.3344 & 0.9755 & -0.2223 & 0.6353 & 0.8866 & b^4_{3x} \\
-0.6412 & -1.1022 & 0.3323 & -1.1030 & -0.2200 & -1.5055 & 0.8864 & -1.6254 & b^4_{3y}
\end{array} \tag{6.10}$$

$$
\mathbf{k}_3^2 = 10^6 \left[\begin{array}{rrrrrrrr|l}
b_8^1 & b_{8y}^1 & b_{8x}^2 & b_{8y}^2 & b_{8x}^3 & b_{8y}^3 & b_{8x}^4 & b_{8y}^4 & \\
0.0348 & -0.5436 & -0.1492 & 0.2973 & -0.2621 & 0.0415 & 0.1608 & -0.4664 & b_{2x}^1 \\
-0.6363 & -2.1444 & 0.2335 & 1.2803 & -0.0611 & 1.9645 & -0.4244 & -1.8310 & b_{2y}^1 \\
0.2877 & 1.8585 & -0.4445 & -0.6033 & 0.0066 & -0.5878 & -0.4132 & 0.4480 & b_{2x}^2 \\
1.3010 & -0.1977 & -0.6037 & 1.0283 & -0.5893 & 1.1667 & 0.4528 & -1.7991 & b_{2y}^2 \\
0.2494 & -0.8378 & 0.1581 & 0.1280 & 0.4067 & -0.3198 & -0.0727 & -0.2713 & b_{2x}^3 \\
-0.9987 & -2.9134 & 0.1274 & 1.0023 & -0.3204 & 2.2895 & -0.2738 & -1.3633 & b_{2y}^3 \\
1.5896 & 1.0412 & -0.5639 & 0.3344 & -0.9755 & -0.2223 & 0.6353 & -0.8866 & b_{2x}^4 \\
0.6412 & -1.1022 & 0.3323 & 1.1030 & -0.2200 & 1.5055 & -0.8864 & -1.6254 & b_{2y}^4 \\
1.0711 & 0.4211 & 2.7033 & -2.2896 & 3.8357 & -2.5456 & -3.5475 & 2.1844 & b_{5x}^1 \\
-0.4211 & -0.4646 & -2.2317 & 5.1104 & -2.3531 & 6.4674 & 2.2519 & -5.7623 & b_{5y}^1 \\
-2.7033 & -2.2317 & 1.6267 & 0.0624 & 1.3253 & 0.2536 & -0.8906 & 0.2809 & b_{5x}^2 \\
-2.2896 & -5.1104 & -0.0624 & 2.1235 & 0.0524 & 3.3908 & 0.3556 & -1.8426 & b_{5y}^2 \\
-3.8357 & -2.3531 & 1.3253 & -0.0524 & 3.0119 & 0.1624 & -1.3991 & 0.4183 & b_{5x}^3 \\
-2.5456 & -6.4674 & -0.2536 & 3.3908 & -0.1624 & 5.7528 & 0.5289 & -2.8592 & b_{5y}^3 \\
-3.5475 & -2.2519 & 0.8906 & 0.3556 & 1.3991 & 0.5289 & -0.3275 & -0.0171 & b_{5x}^4 \\
-2.1844 & -5.7623 & 0.2809 & 1.8426 & 0.4183 & 2.8592 & 0.0171 & -2.0410 & b_{5y}^4 \\
12.4321 & 4.8960 & -0.8240 & -1.6902 & -3.1138 & -3.1255 & 0.1337 & 0.5371 & b_{8x}^1 \\
4.8960 & 17.0180 & -1.3594 & -3.3215 & -2.4441 & -6.3657 & 0.4646 & 3.0697 & b_{8y}^1 \\
-0.8240 & -1.3594 & 2.6386 & -0.4593 & 1.6477 & -0.2535 & -1.8688 & 0.9391 & b_{8x}^2 \\
-1.6902 & -3.3215 & -0.4593 & 3.9208 & -0.2142 & 4.6991 & 0.9085 & -2.7293 & b_{8y}^2 \\
-3.1138 & -2.4441 & 1.6477 & -0.2142 & 4.0634 & -0.0102 & -1.3863 & 0.6447 & b_{8x}^3 \\
-3.1255 & -6.3657 & -0.2535 & 4.6991 & -0.0102 & 7.8957 & 0.6848 & -2.8035 & b_{8y}^3 \\
0.1337 & 0.4646 & -1.8688 & 0.9085 & -1.3863 & 0.6848 & 3.0809 & -0.8593 & b_{8x}^4 \\
0.5371 & 3.0697 & 0.9391 & -2.7293 & 0.6447 & -2.8035 & -0.8593 & 4.6936 & b_{8y}^4 \\
-1.9820 & -0.9324 & -0.2764 & 0.2259 & 0.1235 & 0.7287 & 0.9031 & 0.2251 & b_{3x}^1 \\
-0.5447 & 2.2031 & 0.1857 & -1.4234 & 0.5936 & -1.7169 & 0.2499 & 2.7053 & b_{3y}^1 \\
-2.4985 & -0.0012 & -1.2176 & 0.3536 & 0.2680 & 0.5199 & 1.3446 & -0.3477 & b_{3x}^2 \\
-0.5732 & -3.1013 & 0.3677 & 0.5947 & 0.4765 & 1.9425 & -0.3298 & -0.0330 & b_{3y}^2 \\
1.8389 & -0.2271 & 0.9223 & -0.1449 & 0.4327 & -0.3908 & -1.2432 & -0.1557 & b_{3x}^3 \\
-0.3746 & -0.2816 & -0.1610 & 2.3371 & -0.3455 & 2.8466 & -0.1697 & -2.0542 & b_{3y}^3 \\
-1.7851 & -1.3215 & 0.4334 & 0.0597 & 1.4820 & 0.5918 & 0.1809 & 0.4898 & b_{3x}^4 \\
-0.9694 & 2.2579 & 0.0251 & -0.7102 & 0.6108 & -0.8339 & 0.5340 & 2.7632 & b_{3y}^4
\end{array}\right] \tag{6.11}
$$

$$\mathbf{k}_4^2 = 10^6 \begin{array}{cc}
\begin{array}{rrrrrrrr} b^1_{3x} & b^1_{3y} & b^2_{3x} & b^2_{3y} & b^3_{3x} & b^3_{3y} & b^4_{3x} & b^4_{3y} \end{array} & \\
\left[\begin{array}{rrrrrrrr}
-0.9639 & 0.0491 & 0.6394 & -0.4483 & 1.3047 & -0.0358 & -2.6648 & 0.5448 \\
-0.0491 & -3.3390 & -0.2940 & 1.7389 & 0.0584 & 2.9983 & 0.5041 & -4.6157 \\
-0.6394 & -0.2940 & 2.1119 & -0.0045 & -0.3063 & 0.4244 & -0.8268 & -0.3963 \\
-0.4483 & -1.7389 & 0.0045 & 0.7689 & 0.4526 & 1.4427 & -0.4192 & -2.0420 \\
-1.3047 & 0.0584 & -0.3063 & -0.4526 & 1.4819 & -0.0467 & -2.0428 & 0.6081 \\
-0.0358 & -2.9983 & -0.4244 & 1.4427 & 0.0467 & 2.0738 & 0.5281 & -4.4724 \\
-2.6648 & -0.5041 & 0.8268 & -0.4192 & 2.0428 & 0.5281 & -3.0614 & -0.0740 \\
-0.5448 & -4.6157 & -0.3963 & 2.0420 & 0.6081 & 4.4724 & 0.0740 & -5.6645 \\
0.0348 & 0.6363 & -0.2877 & 1.3010 & -0.2494 & -0.9987 & 1.5896 & -0.6412 \\
0.5436 & -2.1444 & 1.8585 & 0.1977 & -0.8378 & 2.9134 & -1.0412 & -1.1022 \\
0.1492 & 0.2335 & -0.4445 & 0.6037 & 0.1581 & -0.1274 & 0.5639 & 0.3323 \\
0.2973 & -1.2803 & 0.6033 & 1.0283 & -0.1280 & 1.0023 & 0.3344 & -1.1030 \\
0.2621 & -0.0611 & 0.0066 & 0.5893 & 0.4067 & 0.3204 & 0.9755 & -0.2200 \\
0.0415 & -1.9645 & 0.5878 & 1.1667 & 0.3198 & 2.2895 & -0.2223 & -1.5055 \\
0.1608 & 0.4244 & 0.4132 & 0.4528 & 0.0727 & -0.2738 & 0.6353 & 0.8864 \\
0.4664 & -1.8310 & 0.4480 & 1.7991 & -0.2713 & 1.3633 & 0.8866 & -1.6254 \\
-1.9820 & -0.5447 & -2.4985 & -0.5732 & 1.8389 & -0.3746 & -1.7851 & -0.9694 \\
-0.9324 & 2.2031 & -0.0012 & -3.1013 & -0.2271 & -0.2816 & -1.3215 & 2.2579 \\
-0.2764 & 0.1857 & -1.2176 & 0.3677 & 0.9223 & -0.1610 & 0.4334 & 0.0251 \\
0.2259 & -1.4234 & 0.3536 & 0.5947 & -0.1449 & 2.3371 & 0.0597 & -0.7102 \\
0.1235 & 0.5936 & 0.2680 & 0.4765 & 0.4327 & -0.3455 & 1.4820 & 0.6108 \\
0.7287 & -1.7169 & 0.5199 & 1.9425 & -0.3908 & 2.8466 & 0.5918 & -0.8339 \\
0.9031 & 0.2499 & 1.3446 & -0.3298 & -1.2432 & -0.1697 & 0.1809 & 0.5340 \\
0.2251 & 2.7053 & -0.3477 & -0.0330 & -0.1557 & -2.0542 & 0.4898 & 2.7632 \\
3.0196 & 0.5191 & 0.5159 & 0.4124 & -1.9575 & -0.7877 & 2.2076 & 0.3823 \\
0.5191 & 4.4572 & 0.5011 & -0.9909 & -0.7970 & -2.9112 & 0.2305 & 5.2198 \\
0.5159 & 0.5011 & 3.7414 & -0.0872 & -1.2502 & -0.1638 & -0.6077 & 1.0307 \\
0.4124 & -0.9909 & -0.0872 & 2.6385 & -0.1663 & 0.9452 & 1.0345 & -1.4335 \\
-1.9575 & -0.7970 & -1.2502 & -0.1663 & 3.0158 & 0.6434 & -1.4736 & -0.5134 \\
-0.7877 & -2.9112 & -0.1638 & 0.9452 & 0.6434 & 4.6826 & -0.3256 & -2.8348 \\
2.2076 & 0.2305 & -0.6077 & 1.0345 & -1.4736 & -0.3256 & 4.8646 & 0.0839 \\
0.3823 & 5.2198 & 1.0307 & -1.4335 & -0.5134 & -2.8348 & 0.0839 & 8.1103
\end{array}\right] &
\begin{array}{c}
b^1_{2x} \\ b^1_{2y} \\ b^2_{2x} \\ b^2_{2y} \\ b^3_{2x} \\ b^3_{2y} \\ b^4_{2x} \\ b^4_{2y} \\
b^1_{5x} \\ b^1_{5y} \\ b^2_{5x} \\ b^2_{5y} \\ b^3_{5x} \\ b^3_{5y} \\ b^4_{5x} \\ b^4_{5y} \\
b^1_{8x} \\ b^1_{8y} \\ b^2_{8x} \\ b^2_{8y} \\ b^3_{8x} \\ b^3_{8y} \\ b^4_{8x} \\ b^4_{8y} \\
b^1_{3x} \\ b^1_{3y} \\ b^2_{3x} \\ b^2_{3y} \\ b^3_{3x} \\ b^3_{3y} \\ b^4_{3x} \\ b^4_{3y}
\end{array}
\end{array} \tag{6.12}$$

For element 3, the enriched-standard submatrix is

$$\mathbf{K}_{es}^{3} = 10^{6}\begin{array}{cc}
\begin{array}{cccccccc} u_{1x} & u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{3y} & u_{4x} & u_{4y} \end{array} & \\
\left[\begin{array}{cccccccc}
-1.7149 & -0.8565 & 0.4495 & -0.1875 & 1.9864 & 1.4529 & -0.7210 & -0.4089 \\
-0.7524 & -3.1836 & 0.0565 & -1.2452 & 1.8416 & 1.9412 & -1.1456 & 2.4877 \\
-0.5859 & -0.0923 & 0.4937 & -0.3045 & 0.4322 & 0.3967 & -0.3399 & 0.0001 \\
-0.0062 & -0.3250 & -0.258 & 0.0022 & 0.3967 & 0.2623 & -0.1322 & 0.0605 \\
-0.8861 & -0.1059 & 0.7856 & -0.5179 & 0.6699 & 0.6183 & -0.5694 & 0.0055 \\
0.0518 & -0.4165 & -0.4677 & 0.0650 & 0.6183 & 0.3509 & -0.2025 & 0.0007 \\
-0.6002 & -0.1564 & 0.4595 & -0.2638 & 0.5208 & 0.4045 & -0.3801 & 0.0157 \\
-0.0867 & -0.4173 & -0.1934 & -0.0751 & 0.4045 & 0.3552 & -0.1244 & 0.1372 \\
-0.0962 & -0.0481 & 0.4808 & -0.2404 & 0.1923 & -0.1442 & -0.5769 & 0.4327 \\
0.0481 & 0.3846 & -0.2404 & 0.9615 & -0.3365 & -0.7692 & 0.5288 & -0.5769 \\
0.5769 & 0.4327 & -0.1923 & -0.1442 & -0.4808 & -0.2404 & 0.0962 & -0.0481 \\
0.5288 & 0.5769 & -0.3365 & 0.7692 & -0.2404 & -0.9615 & 0.0481 & -0.3846 \\
0.7210 & -0.4089 & -1.9864 & 1.4529 & -0.4495 & -0.1875 & 1.7149 & -0.8565 \\
-1.1456 & -2.4877 & 1.8416 & -1.9412 & 0.0565 & 1.2452 & -0.7524 & 3.1836 \\
-0.3399 & -0.0001 & 0.4322 & -0.3967 & 0.4937 & 0.3045 & -0.5859 & 0.0923 \\
0.1322 & 0.0605 & -0.3967 & 0.2623 & 0.2583 & 0.0022 & 0.0062 & -0.3250 \\
-0.5694 & -0.0055 & 0.6699 & -0.6183 & 0.7856 & 0.5179 & -0.8861 & 0.1059 \\
0.2025 & 0.0007 & -0.6183 & 0.3509 & 0.4677 & 0.0650 & -0.0518 & -0.4165 \\
0.3801 & 0.0157 & -0.5208 & 0.4045 & -0.4595 & -0.2638 & 0.6002 & -0.1564 \\
-0.1244 & -0.1372 & 0.4045 & -0.3552 & -0.1934 & 0.0751 & -0.0867 & 0.4173
\end{array}\right] &
\begin{array}{c}
b_{5x}^{1} \\ b_{5y}^{1} \\ b_{5x}^{2} \\ b_{5y}^{2} \\ b_{5x}^{3} \\ b_{5y}^{3} \\ b_{5x}^{4} \\ b_{5y}^{4} \\ c_{6x} \\ c_{6y} \\ c_{7x} \\ c_{7y} \\ b_{8x}^{1} \\ b_{8y}^{1} \\ b_{8x}^{2} \\ b_{8y}^{2} \\ b_{8x}^{3} \\ b_{8y}^{3} \\ b_{8x}^{4} \\ b_{8y}^{4}
\end{array}
\end{array} \tag{6.13}$$

The enriched-enriched submatrix is split for viewing convenience into three submatrices by dividing its columns

$$\mathbf{K}_{ee}^{3} = \left[\begin{array}{ccc} \mathbf{k}_{1}^{3} & \mathbf{k}_{2}^{3} & \mathbf{k}_{3}^{3} \\ (20 \times 8) & (20 \times 4) & (20 \times 8) \end{array}\right] \tag{6.14}$$

$$\mathbf{k}_1^3 = 10^6 \begin{array}{rrrrrrrrl}
b_{5x}^1 & b_{5y}^1 & b_{5x}^2 & b_{5y}^2 & b_{5x}^3 & b_{5y}^3 & b_{5x}^4 & b_{5y}^4 & \\
13.8705 & 5.1975 & 2.0502 & 0.7177 & 3.5374 & 1.2159 & 1.5609 & 0.6104 & b_{5x}^1 \\
5.1975 & 26.4649 & 0.6792 & 5.2796 & 1.1345 & 9.0905 & 0.7534 & 3.7132 & b_{5y}^1 \\
2.0502 & 0.6792 & 1.0978 & 0.1690 & 1.8224 & 0.2346 & 0.8431 & 0.2228 & b_{5x}^2 \\
0.7177 & 5.2796 & 0.1690 & 2.0517 & 0.2457 & 3.5826 & 0.2339 & 1.2216 & b_{5y}^2 \\
3.5374 & 1.1345 & 1.8224 & 0.2457 & 3.0754 & 0.3500 & 1.3168 & 0.3216 & b_{5x}^3 \\
1.2159 & 9.0905 & 0.2346 & 3.5826 & 0.3500 & 6.3480 & 0.3304 & 1.9734 & b_{5y}^3 \\
1.5609 & 0.7534 & 0.8431 & 0.2339 & 1.3168 & 0.3304 & 0.8690 & 0.2818 & b_{5x}^4 \\
0.6104 & 3.7132 & 0.2228 & 1.2216 & 0.3216 & 1.9734 & 0.2818 & 1.1805 & b_{5y}^4 \\
0.4821 & 0.3534 & 0.0516 & 0.1686 & 0.0696 & 0.3110 & 0.1284 & 0.0187 & c_{6x} \\
-0.3694 & -3.1250 & 0.0302 & -0.5783 & 0.0747 & -0.9925 & -0.0792 & -0.3827 & c_{6y} \\
0.0520 & 0.3002 & -0.2908 & -0.1317 & -0.4045 & -0.1302 & -0.3397 & -0.2405 & c_{7x} \\
0.3549 & 0.5303 & -0.2701 & 0.2323 & -0.3665 & 0.5778 & -0.3384 & -0.0906 & c_{7y} \\
-1.2547 & -0.3663 & -2.4493 & 0.4146 & -4.2686 & 0.8166 & -1.8219 & 0.0597 & b_{8x}^1 \\
0.3663 & -5.2272 & 0.6772 & -4.2286 & 1.1769 & -8.0132 & 0.3599 & -2.1031 & b_{8y}^1 \\
2.4493 & 0.6772 & 0.6863 & 0.0623 & 1.1889 & 0.1070 & 0.5433 & 0.0438 & b_{8x}^2 \\
0.4146 & 4.2286 & -0.0623 & 1.0571 & -0.0935 & 1.9883 & -0.0289 & 0.6293 & b_{8y}^2 \\
4.2686 & 1.1769 & 1.1889 & 0.0935 & 2.0743 & 0.1618 & 0.8866 & 0.0825 & b_{8x}^3 \\
0.8166 & 8.0132 & -0.1070 & 1.9883 & -0.1618 & 3.7371 & -0.0319 & 1.0742 & b_{8y}^3 \\
-1.8219 & -0.3599 & -0.5433 & -0.0289 & -0.8866 & -0.0319 & -0.5445 & -0.0180 & b_{8x}^4 \\
-0.0597 & -2.1031 & 0.0438 & -0.6293 & 0.0825 & -1.0742 & 0.0180 & -0.6988 & b_{8y}^4
\end{array} \tag{6.15}$$

$$\mathbf{k}_2^3 = 10^6 \begin{array}{rrrrl}
c_{6x} & c_{6y} & c_{7x} & c_{7y} & \\
0.4821 & -0.3694 & 0.0520 & 0.3549 & b_{5x}^1 \\
0.3534 & -3.1250 & 0.3002 & 0.5303 & b_{5y}^1 \\
0.0516 & 0.0302 & -0.2908 & -0.2701 & b_{5x}^2 \\
0.1686 & -0.5783 & -0.1317 & 0.2323 & b_{5y}^2 \\
0.0696 & 0.0747 & -0.4045 & -0.3665 & b_{5x}^3 \\
0.3110 & -0.9925 & -0.1302 & 0.5778 & b_{5y}^3 \\
0.1284 & -0.0792 & -0.3397 & -0.3384 & b_{5x}^4 \\
0.0187 & -0.3827 & -0.2405 & -0.0906 & b_{5y}^4 \\
0.9615 & -0.4808 & 0.0000 & 0.0000 & c_{6x} \\
-0.4808 & 1.9231 & 0.0000 & 0.0000 & c_{6y} \\
0.0000 & 0.0000 & 0.9615 & 0.4808 & c_{7x} \\
0.0000 & 0.0000 & 0.4808 & 1.9231 & c_{7y} \\
0.0520 & -0.3549 & 0.4821 & 0.3694 & b_{8x}^1 \\
-0.3002 & 0.5303 & -0.3534 & -3.1250 & b_{8y}^1 \\
0.2908 & -0.2701 & -0.0516 & 0.0302 & b_{8x}^2 \\
-0.1317 & -0.2323 & 0.1686 & 0.5783 & b_{8y}^2 \\
0.4045 & -0.3665 & -0.0696 & 0.0747 & b_{8x}^3 \\
-0.1302 & -0.5778 & 0.3110 & 0.9925 & b_{8y}^3 \\
-0.3397 & 0.3384 & 0.1284 & 0.0792 & b_{8x}^4 \\
0.2405 & -0.0906 & -0.0187 & -0.3827 & b_{8y}^4
\end{array} \tag{6.16}$$

$$\mathbf{k}_1^3 = 10^6 \begin{array}{|rrrrrrrr|l}
\multicolumn{1}{c}{b_{8x}^1} & b_{8y}^1 & b_{8x}^2 & b_{8y}^2 & b_{8x}^3 & b_{8y}^3 & b_{8x}^4 & \multicolumn{1}{c}{b_{8y}^4} & \\
\hline
-1.2547 & 0.3663 & 2.4493 & 0.4146 & 4.2686 & 0.8166 & -1.8219 & -0.0597 & b_{5x}^1 \\
-0.3663 & -5.2272 & 0.6772 & 4.2286 & 1.1769 & 8.0132 & -0.3599 & -2.1031 & b_{5y}^1 \\
-2.4493 & 0.6772 & 0.6863 & -0.0623 & 1.1889 & -0.1070 & -0.5433 & 0.0438 & b_{5x}^2 \\
0.4146 & -4.2286 & 0.0623 & 1.0571 & 0.0935 & 1.9883 & -0.0289 & -0.6293 & b_{5y}^2 \\
-4.2686 & 1.1769 & 1.1889 & -0.0935 & 2.0743 & -0.1618 & -0.8866 & 0.0825 & b_{5x}^3 \\
0.8166 & -8.0132 & 0.1070 & 1.9883 & 0.1618 & 3.7371 & -0.0319 & -1.0742 & b_{5y}^3 \\
-1.8219 & 0.3599 & 0.5433 & -0.0289 & 0.8866 & -0.0319 & -0.5445 & 0.0180 & b_{5x}^4 \\
0.0597 & -2.1031 & 0.0438 & 0.6293 & 0.0825 & 1.0742 & -0.0180 & -0.6988 & b_{5y}^4 \\
0.0520 & -0.3002 & 0.2908 & -0.1317 & 0.4045 & -0.1302 & -0.3397 & 0.2405 & c_{6x} \\
-0.3549 & 0.5303 & -0.2701 & -0.2323 & -0.3665 & -0.5778 & 0.3384 & -0.0906 & c_{6y} \\
0.4821 & -0.3534 & -0.0516 & 0.1686 & -0.0696 & 0.3110 & 0.1284 & -0.0187 & c_{7x} \\
0.3694 & -3.1250 & 0.0302 & 0.5783 & 0.0747 & 0.9925 & 0.0792 & -0.3827 & c_{7y} \\
13.8705 & -5.1975 & -2.0502 & 0.7177 & -3.5374 & 1.2159 & 1.5609 & -0.6104 & b_{8x}^1 \\
-5.1975 & 26.4649 & 0.6792 & -5.2796 & 1.1345 & -9.0905 & -0.7534 & 3.7132 & b_{8y}^1 \\
-2.0502 & 0.6792 & 1.0978 & -0.1690 & 1.8224 & -0.2346 & -0.8431 & 0.2228 & b_{8x}^2 \\
0.7177 & -5.2796 & -0.1690 & 2.0517 & -0.2457 & 3.5826 & 0.2339 & -1.2216 & b_{8y}^2 \\
-3.5374 & 1.1345 & 1.8224 & -0.2457 & 3.0754 & -0.3500 & -1.3168 & 0.3216 & b_{8x}^3 \\
1.2159 & -9.0905 & -0.2346 & 3.5826 & -0.3500 & 6.3480 & 0.3304 & -1.9734 & b_{8y}^3 \\
1.5609 & -0.7534 & -0.8431 & 0.2339 & -1.3168 & 0.3304 & 0.8690 & -0.2818 & b_{8x}^4 \\
-0.6104 & 3.7132 & 0.2228 & -1.2216 & 0.3216 & -1.9734 & -0.2818 & 1.1805 & b_{8y}^4
\end{array} \tag{6.17}$$

6.2.3 Displacements

After assembling the global stiffness matrix from the elementary sub-matrices of the previous section and calculating the global force vector, and the linear system is solved. The displacements along the standard and artificial degrees of freedom are:

$$
\tilde{\mathbf{u}} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ u_{4x} \\ u_{4y} \\ u_{5x} \\ u_{5y} \\ u_{6x} \\ u_{6y} \\ u_{7x} \\ u_{7y} \\ u_{8x} \\ u_{8y} \end{bmatrix} = 10^{-3} \begin{bmatrix} 0.0000 \\ 0.0000 \\ -0.2658 \\ 0.6944 \\ -0.2658 \\ -0.6944 \\ 0.0000 \\ 0.0000 \\ -8.1728 \\ -13.7686 \\ -8.1728 \\ -50.0000 \\ -8.1728 \\ 50.0000 \\ -8.7899 \\ 13.7686 \end{bmatrix}
\qquad
\tilde{\mathbf{a}} = \begin{bmatrix} b^1_{2x} \\ b^1_{2y} \\ b^2_{2x} \\ b^2_{2y} \\ b^3_{2x} \\ b^3_{2y} \\ b^4_{2x} \\ b^4_{2y} \\ b^1_{3x} \\ b^1_{3y} \\ b^2_{3x} \\ b^2_{3y} \\ b^3_{3x} \\ b^3_{3y} \\ b^4_{3x} \\ b^4_{3y} \\ b^1_{5x} \\ b^1_{5y} \\ b^2_{5x} \\ b^2_{5y} \\ b^3_{5x} \\ b^3_{5y} \\ b^4_{5x} \\ b^4_{5y} \\ c_{6x} \\ c_{6y} \\ c_{7x} \\ c_{7y} \\ b^1_{8x} \\ b^1_{8y} \\ b^2_{8x} \\ b^2_{8y} \\ b^3_{8x} \\ b^3_{8y} \\ b^4_{8x} \\ b^4_{8y} \end{bmatrix} = 10^{6} \begin{bmatrix} -0.4854 \\ 1.2571 \\ 0.0411 \\ -1.3658 \\ 0.1798 \\ -0.9780 \\ 0.0815 \\ -0.8043 \\ 0.4854 \\ 1.2571 \\ 0.0411 \\ 1.3658 \\ 0.1798 \\ 0.9780 \\ -0.0815 \\ -0.8043 \\ -2.7590 \\ -4.2708 \\ -1.5821 \\ 0.5124 \\ -1.8491 \\ -0.6714 \\ -0.6922 \\ 1.4914 \\ 15.6557 \\ 49.8864 \\ -15.6557 \\ 49.8864 \\ 2.7590 \\ -4.2708 \\ -1.5821 \\ -0.5124 \\ -1.8491 \\ 0.6714 \\ 0.6922 \\ 1.4914 \end{bmatrix} \tag{6.18}
$$

6.2.4 Crack propagation step

This section presents the computation of the stress intensity factors. The domain, material properties and XFEM parameters are exactly the same as in Section 6.2.1, with the following exceptions

- The mesh is refined. A uniform mesh consisting of Quad4 elements is used, as illustrated in Figure 6.2.
- To calculate the interaction integrals over all elements in the J-integral domain, the integration with sub-quads is employed (see Section 4.2). All elements are split into 8×8 sub-quads, and each sub-quad contains 4×4 integration points.

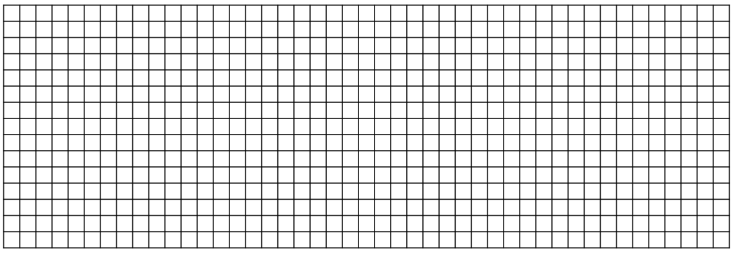


Fig. 6.2 Uniform mesh for the DCB.

The J-integral domain consists of the elements intersected by a circle centered at the crack tip and with radius r_J. The computation is performed for various ratios of this J-integral radius over the element size, as well as for various element sizes. The J-integral and Mode I stress SIF for each case are given in Table 6.1 and Table 6.2 respectively. The Mode II sif is not included as $K_{\text{II}} = 0$ in all cases. It can be observed that

- As the mesh is refined, both J and K_{I} converge.
- Increasing the ratio of the J-integral radius over the element size improves the accuracy until a certain point, around 2.0 - 3.0. After that the accuracy is increased much slower.

J-integral		**J-integral radius / element size**				
		1	2	3	4	5
Mesh	15x45	2.2701	2.1317	2.1190	2.1171	2.1153
	25x75	2.2552	2.1229	2.1137	2.1128	2.1115
	45x135	2.2359	2.1083	2.1013	2.1012	2.1003

Table 6.1 The values of J-integral using various meshes and ratios of J-integral radius per element size

$\mathbf{K_I}$		**J-integral radius / element size**				
		1	2	3	4	5
Mesh	15x45	2233.6804	2164.5234	2158.0247	2157.0788	2156.1421
	25x75	2226.2993	2160.0050	2155.3326	2154.9041	2154.2405
	45x135	2216.7518	2152.5914	2148.9863	2148.9365	2148.5228

Table 6.2 The values of K_{I} using various meshes and J-integral radius per element size ratios

6.3 Finite crack in an infinite plate

6.3.1 Problem formulation

The second example involves a finite crack inside an infinite plate subjected to a far field uniaxial tension, as illustrated in Figure 6.3. For this particular problem, an analytic solution is provided in [14]

$$K_{\mathrm{I}} = \sigma\sqrt{\pi\alpha}\cos^2\beta \tag{6.19a}$$

$$K_{\mathrm{II}} = \sigma\sqrt{\pi\alpha}\cos\beta\sin\beta \tag{6.19b}$$

where α half the crack length, β is the angle between global x axis and the crack and σ is the far field uniform tensile load.

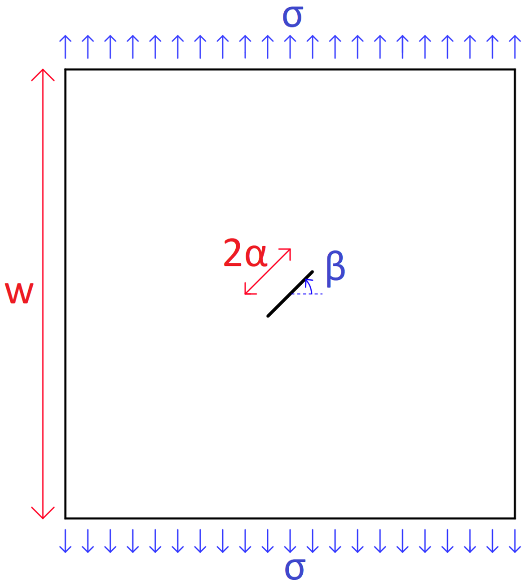


Fig. 6.3 Infinite plate with finite crack under tension.

Modeling the problem has been done according to the following:

- The width of the plate is $w = 200cm$, which is significantly larger than the crack length $2\alpha = 4cm$. There is no way to exactly model an infinite plate. At best a high w/α ratio can be selected.
- Plane stress problem with material properties: Young's modulus $E = 2.1 \cdot 10^6 \frac{kg}{cm^2}$, Poisson ratio $\upsilon = 0.3$ and thickness $t = 1cm$.
- To simulate the boundary conditions of this infinite plate, all nodes along the bottom horizontal edge of the plate are constrained, while the uniform tension $\sigma = 2000\frac{kg}{cm^2}$ is applied on the top horizontal edge. While this serves to replicate the stress conditions of the original problem, inevitably some errors are introduced, especially for lower w/α ratios.
- The structured rectilinear finite element mesh shown in Figure 6.4 is employed in order to reduce the computational cost of the analysis. This mesh is fine near the crack and coarse in the rest of the domain. The mesh size of the fine region can be controlled to study the convergence of the method.
- The numerical integration is performed according to Section 4.2. To construct the stiffness matrices, elements that are intersected by the crack or contain a crack tip, are split into 8×8 subquads and each subquad is integrated using 2×2 integration points. To compute the interaction integrals, all elements in the J-integral domain are split into 8×8 subquads and each subquad is integrated using 4×4 integration points.

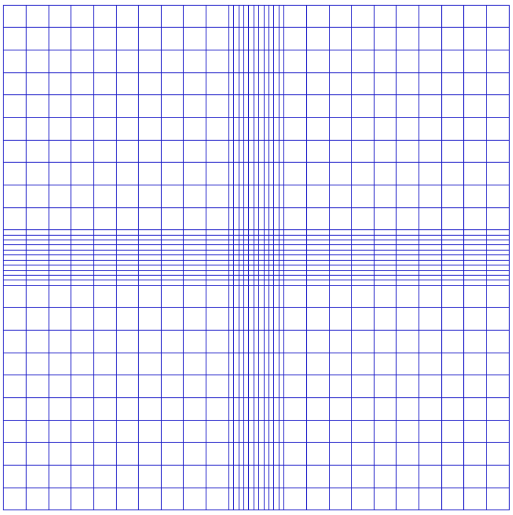


Fig. 6.4 Rectilinear mesh that gets finer near the crack.

6.3.2 Sensitivity to mesh size

At first the convergence of the method as the mesh gets refined is investigated. For a horizontal angle ($\beta = 0$) the analytic solution is

$$K_{\mathrm{I}}^{analytic} = \sigma\sqrt{\pi\alpha}\cos^2\beta = 5013.2565\frac{kg}{cm^{\frac{3}{2}}} \tag{6.20a}$$

$$K_{\mathrm{II}}^{analytic} = \sigma\sqrt{\pi\alpha}\cos\beta\sin\beta = 0 \tag{6.20b}$$

$$J^{analytic} = \frac{K_{\mathrm{I}}^2 + K_{\mathrm{II}}^2}{E^*} = 11.9680 \tag{6.20c}$$

To measure the convergence, the mesh is refined to different degrees quantified by the ratio of the crack length to the (finest) element size $2\alpha/h$. The radius of the J-integral outer contour to the element size is set to $r_J/h = 3$. Table 6.3 aggregates the computed J, K_{I}, K_{II} for various values of $2\alpha/h$. The relative error in the J-integral and in K_{I} are displayed in Figure 6.5. It can be observed that the accuracy of the method improves as the mesh is refined. Note that these results correspond to only one of the crack tips (the rightmost one). They are almost identical to the respective J, K_{I}, K_{II} computed at the other tip.

Crack length / element size	$\mathbf{J^{numeric}}$	$\mathbf{K_I^{numeric}}$	$\mathbf{K_{II}^{numeric}}$
1.0	57.7404	11011.5762	0.0000
1.5	41.4483	9329.5970	0.0000
2.0	27.7129	7628.7001	0.0000
2.5	22.9903	6948.3606	0.0000
3.0	14.5143	5520.8781	0.0000
3.5	12.9094	5206.7030	0.0000
4.0	12.0625	5033.0086	0.0000
4.5	12.0753	5035.6946	0.0000
5.0	11.9961	5019.1456	0.0000
5.5	12.0192	5023.9750	0.0000
6.0	11.9835	5016.5116	0.0000

Table 6.3 J-integral and SIFs for meshes of various size.

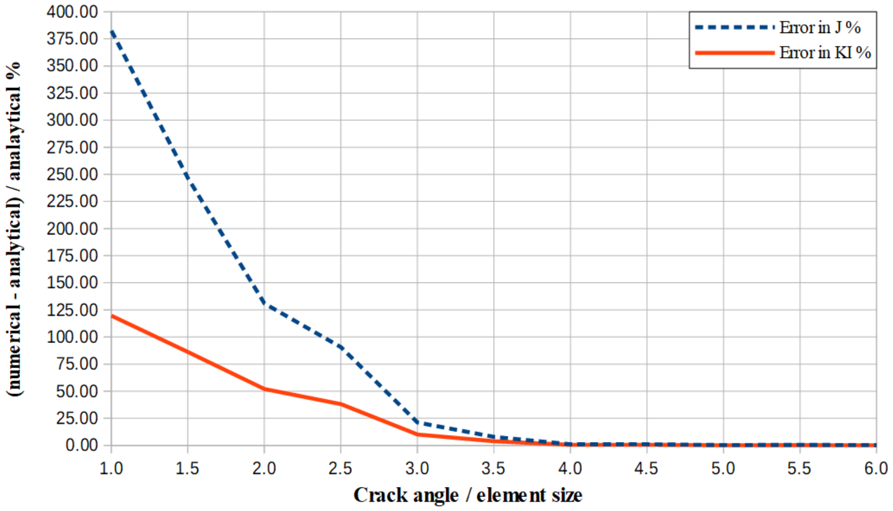


Fig. 6.5 Effect of mesh size on the accuracy of K_{I} and J-integral.

6.3.3 Sensitivity to J-integral radius

Next the effect of the J-integral radius is investigated, by setting the mesh size constant to $2\alpha/h = 4$ and the crack angle to $\beta = 0$. The resulting J, K_{I}, K_{II} for various values of r_J/h are shown in Table 6.4, while the relative errors are plotted in Figure 6.6. It can be observed that by increasing the the J-integral radius, the accuracy initially improves, then remains constant and then rapidly deteriorates. The latter is due to the J-integral radius exceeding the crack length, which causes the J-integral to be computed only from elements without any discontinuities. As mentioned in Section 3.5.2, the J-integral ceases to be path independent when the radius of the outer contour exceeds the crack length.

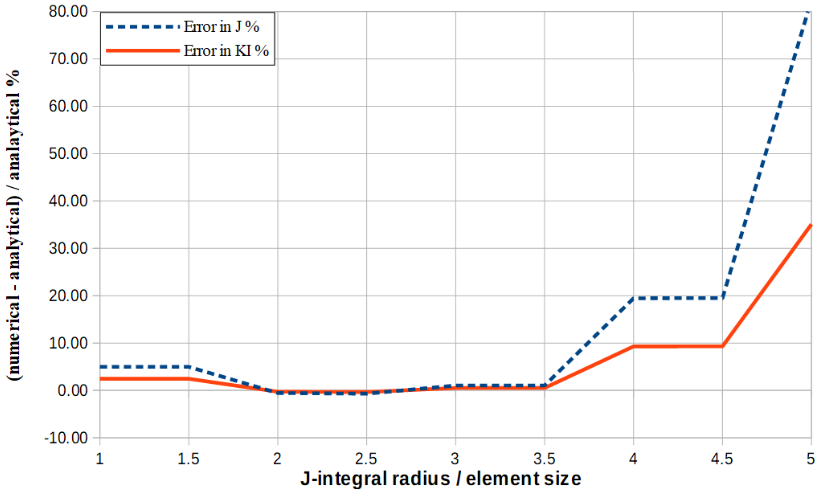


Fig. 6.6 Effect of the J-integral radius on the accuracy of K_{I} and J-intgral.

J-integral radius / element size	$\mathbf{J}^{\textbf{numeric}}$	$\mathbf{K}_{\mathrm{I}}^{\textbf{numeric}}$	$\mathbf{K}_{\mathrm{II}}^{\textbf{numeric}}$
1.0	12.4661	5116.5311	0.0000
1.5	12.4661	5116.5311	0.0000
2.0	11.8953	4998.0064	0.0000
2.5	11.8664	4991.9317	0.0000
3.0	12.0625	5033.0086	0.0000
3.5	12.0625	5033.0086	0.0000
4.0	14.3200	5483.7963	0.0000
4.5	14.3262	5484.9755	0.0000
5.0	21.8983	6781.3356	0.0000
5.5	21.9056	6782.4610	0.0000
6.0	27.7423	7632.7477	0.0000

Table 6.4 J-integral and SIFs for various J-integral domains.

6.3.4 Sensitivity to crack angle

Finally, different values of the crack angle β are considered, while the mesh size and J-integral radius are constant: $2\alpha/h = 4$ and $r_J/h = 3$. The evolution of the analytic and computed SIFs and J-integrals as the crack angle changes is shown in Table 6.5 and Figure 6.7.

Crack angle $(^{\mathbf{o}})$	$\mathbf{J}^{\textbf{analytic}}$	$\mathbf{K}_{\mathrm{I}}^{\textbf{analytic}}$	$\mathbf{K}_{\mathrm{II}}^{\textbf{analytic}}$	$\mathbf{J}^{\textbf{numeric}}$	$\mathbf{K}_{\mathrm{I}}^{\textbf{numeric}}$	$\mathbf{K}_{\mathrm{II}}^{\textbf{numeric}}$
0	11.9680	5013.2565	0.0000	12.3174	5085.9087	0.4609
10	11.6071	4862.0884	857.3174	11.9420	4933.6146	858.8236
20	10.5680	4426.8169	1611.2296	10.8958	4500.9774	1619.3447
30	8.9760	3759.9424	2170.8038	9.1877	3815.9458	2175.4961
40	7.0231	2941.8997	2468.5470	6.8380	2922.3058	2412.4633
50	4.9449	2071.3568	2468.5470	5.0125	2124.7158	2451.9174
60	2.9920	1253.3141	2170.8038	3.0999	1327.0431	2179.1658
70	1.4000	586.4396	1611.2296	1.4544	661.1053	1617.7553
80	0.3609	151.1682	857.3174	0.3776	227.5216	860.9657
90	0.0000	0.0000	0.0000	0.0028	77.0290	0.0000

Table 6.5 J-integral and SIFs for various J-integral domains.

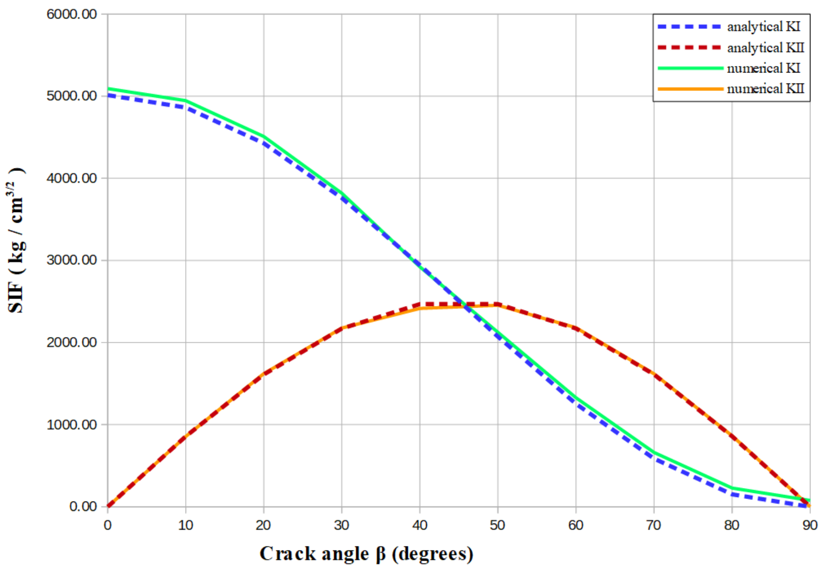


Fig. 6.7 Effect of the crack angle on the stress intensity factors.

6.4 Crack propagation from a fillet

6.4.1 Problem formulation

This example shows the propagation from a crack in an I-beam. The configuration of the problem is taken from experimental work found in [18] and is depicted in Figure 6.8. The computational domain is outlined by dashed lines. The model of the problem is characterized by

- The structure is in plane stress condition. It is subjected to a load $P = 20kN$, applied at the top boundary.
- Two sets of boundary conditions are considered:
  - The global y degree of freedom is constrained for all nodes along the bottom boundary, in order to simulate a very thick, rigid I-beam.
  - The global y degree of freedom is constrained only for the corner nodes of the bottom boundary, in order to simulate a very thin, flexible I-beam.
- The beam is made of steel: $E = 210GPa$, $\upsilon = 0.3$
- The initial crack has a length of $5mm$.

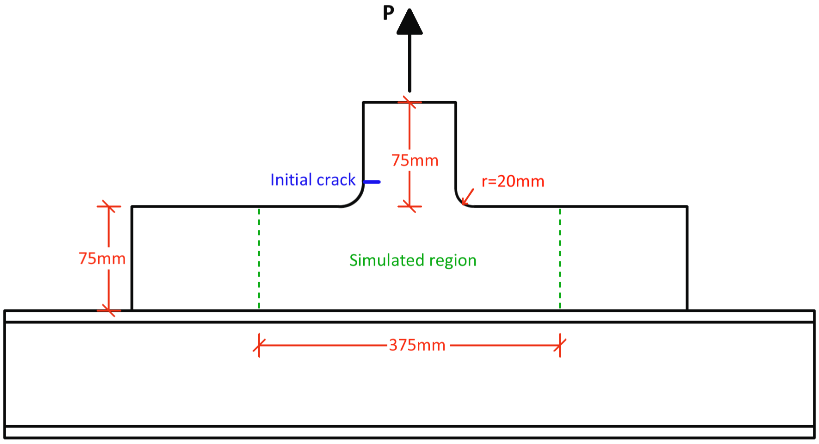


Fig. 6.8 Crack in the lower part of an I-beam.

Figure 6.9 illustrates the unstructured mesh used. The mesh is refined near the fillets to better conform to the curved geometry. Simultaneously the crack is expected to pass through that area, so the accuracy of XFEM is further increased. Although a triangular mesh is viable, it resulted in lower accuracy during testing, therefore 4-nodes quadrilateral finite elements were selected.

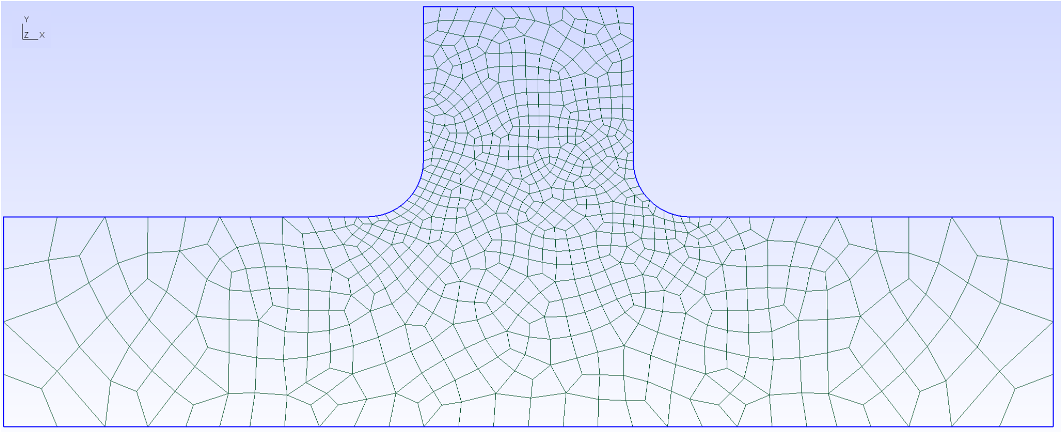


Fig. 6.9 Unstructured mesh for the I-beam.

6.4.2 Crack path

An explicit crack description (see Section 5.2) is employed to model the crack. The simulation progresses for 12 iterations, with a constant propagation length $\Delta L = 5mm$. The crack paths for the rigid (upper path) and flexible (lower path) I-beam are illustrated in Figure 6.10. In the same figure, the crack paths for the two I-beams from [24] are also plotted with dashed lines. It can be concluded that the results presented here are in agreement with the ones in [24].

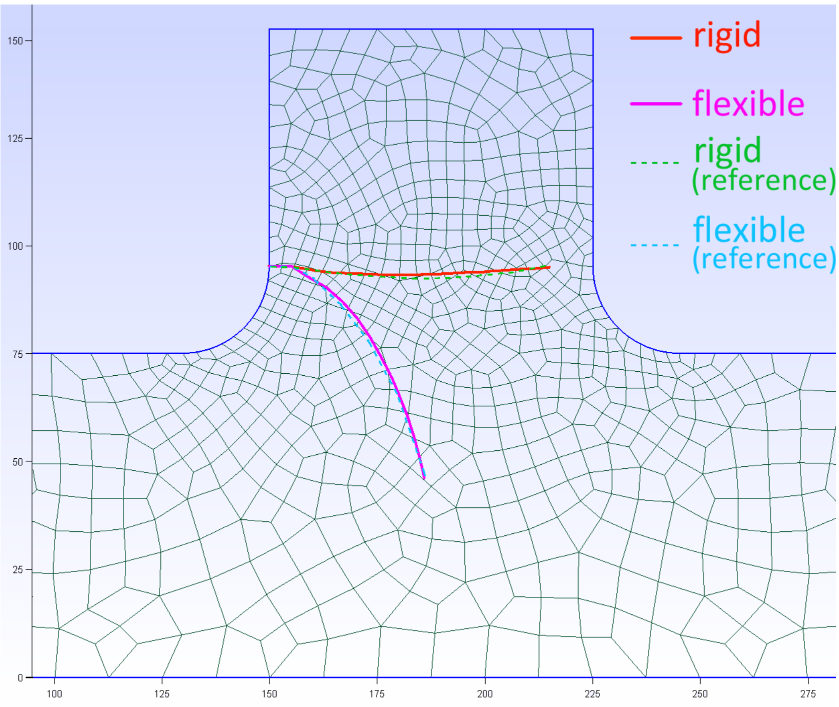


Fig. 6.10 Crack propagation for the rigid (upper path) and flexible (lower path) I-beam. The reference paths (dashed lines) were published in [24].

6.5 Crack growth in a double cantilever beam

6.5.1 Problem formulation

This example is taken from [21] and involves a crack propagating in a double cantilever beam, which is shown in Figure 6.11. The computational model is described in the following:

- The dimensions of the DCB are $h = 3.94$ *in* and $L = 3h = 11.82$ *in*.
- The right side of the DCB is fixed (constrained x,y dofs on all nodes with $x = L$), while a tensile load $P = 197$ *lbs* is applied on the top left and bottom left nodes.
- The structure is in plane stress conditions and the material properties are $E = 3 \cdot 10^7$ *psi* and $v = 0.3$.
- The first crack segment is horizontal at half the beam's height and has length $a = 3.95$ *in*. An extra crack segment is added with length $da = 0.5$ *in* and at a slight angle $d\theta = 5.71^o$. Without this kinking, the crack would propagate in pure Mode I, provided the mesh and boundary conditions are symmetric.

- The crack advances quasi-statically at various propagation lengths. An explicit crack description is used to model the crack interface, as described in Section 5.2.

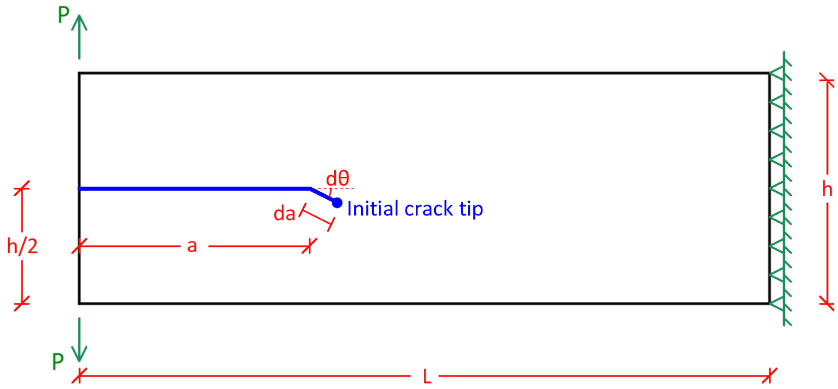


Fig. 6.11 Crack propagation in a double cantilever beam

At first, a sample analysis with a coarse uniform mesh was executed to obtain the general area where the crack is expected to pass. Then the mesh was refined in that area in order to improve the accuracy of XFEM and make sure that consecutive crack tips do not lie inside the same element. This significantly deteriorates the quality of the J-integral (see Section 3.5.2) and the crack geometry descriptions (see Sections 5.2 and 5.3) and should always be avoided. The resulting rectilinear mesh is shown in Figure 6.12.

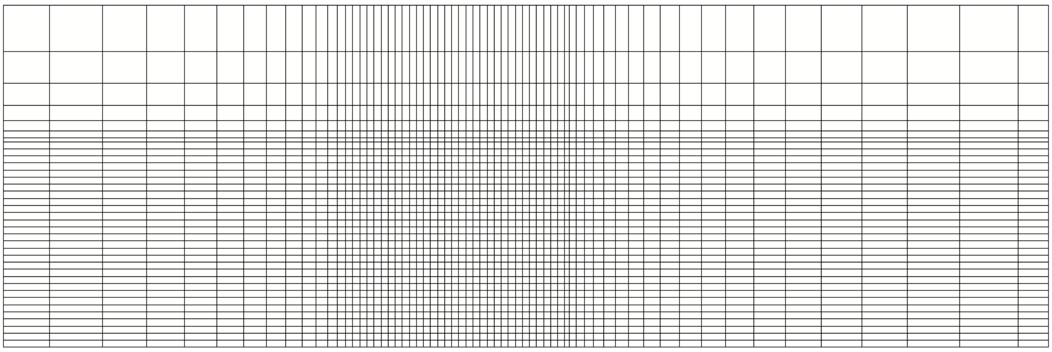


Fig. 6.12 Rectilinear mesh for the DCB. The area where the crack is expected to pass is refined.

6.5.2 Sensitivity to propagation length

This section investigates the effect of the growth length on the crack path. A constant mesh of 8777 elements is used, while the crack is allowed to propagate with various growth lengths. The ratio of the J-integral radius over the element size is fixed at $r_J/h_{el} = 2.0$. Figure 6.13

illustrates the resulting crack paths. It can be observed that the growth length significantly affects the crack path, with smaller lengths resulting in more accurate predictions.

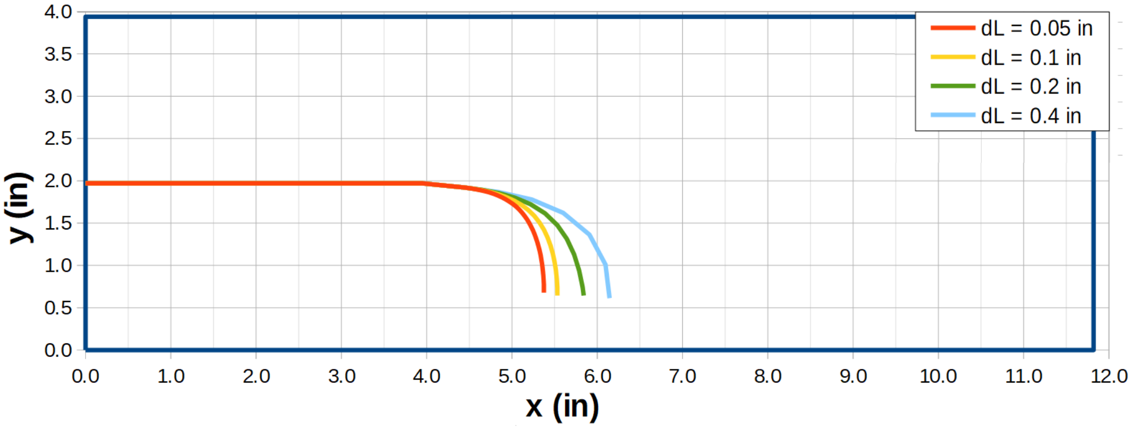


Fig. 6.13 Effect of the propagation length on the crack path

6.5.3 Sensitivity to mesh size

The crack path also depends on the element size, albeit at a lower degree. To examine the convergence of the crack paths when the mesh is refined, a series of independent crack propagation analyses were performed over different meshes. In all cases the growth length was fixed at $dL = 0.3\ in$. The ratio of the J-integral radius over the element size was also constant and equal to $r_J/h_{el} = 2.0$. The crack paths are compared in Figure 6.14. It can be seen that the predicted crack path quickly converges, and thus its sensitivity to the mesh size is not as high as the sensitivity to the growth length.

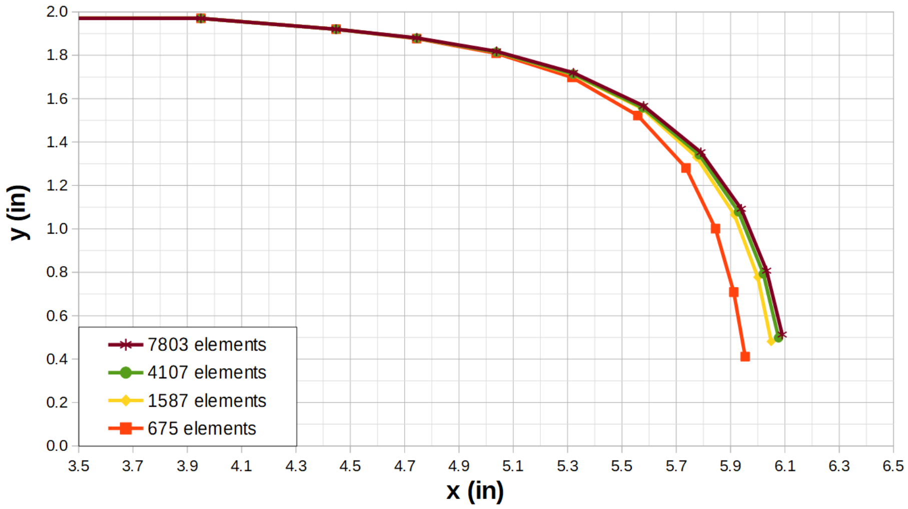


Fig. 6.14 Effect of the mesh size on the crack path

6.5.4 Implicit vs explicit crack geometry description

Finally the accuracy of LSM (see Section 5.3) is compared to that of the explicit crack description. The predicted paths of the two methods for two mesh sizes are depicted in Figure 6.15. The ”exact” path is computed on a very fine mesh of 7803 elements, using the explicit crack description, which does not introduce inaccuracies in the geometric operations, unlike LSM. As in the previous section, the growth length is fixed at $dL = 0.3$ *in* and the ratio of the J-integral radius over the element size at $r_J/h_{el} = 2.0$. It can be concluded from Figure 6.15, that both methods offer comparable accuracy, even on coarse meshes. As expected, the explicit description is more accurate, but this advantage is mitigated when the mesh is refined.

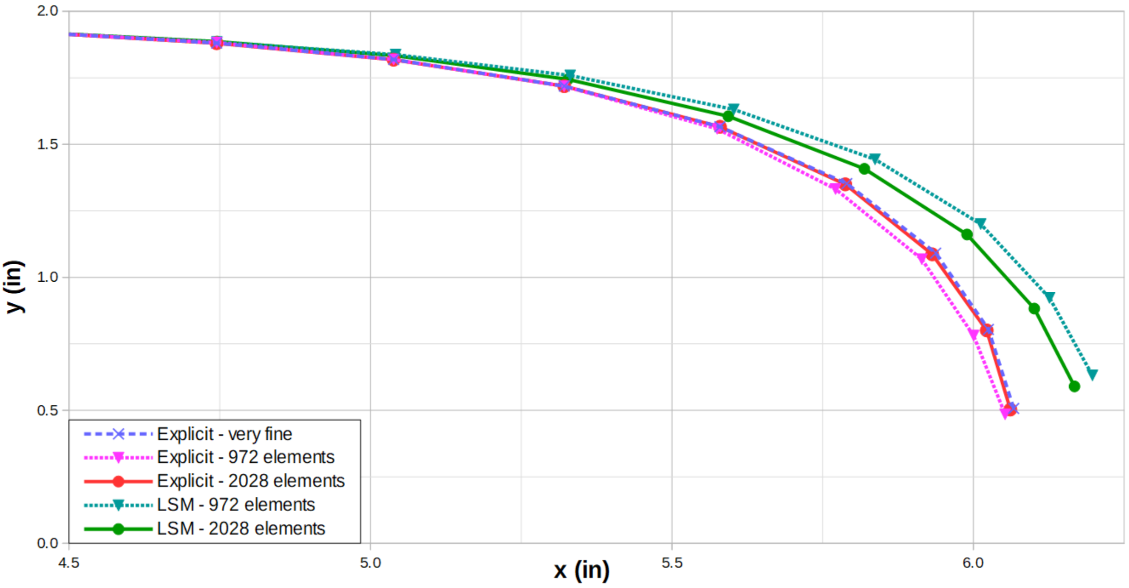


Fig. 6.15 LSM vs explicit crack geometry description

Chapter 7

Conclusions and future work

7.1 Summary

In this thesis the eXtended Finite Element Method was studied for crack propagation problems. By employing XFEM, there is no need to update the mesh at each time step, so that it conforms to the crack interface. Instead the polynomial basis of the classic FEM is enriched with special functions. These enrichment functions enable the approximated displacement field to incorporate the effects of the crack interface and crack tip.

The crack propagation model presented in this work is based on Linear Elastic Fracture Mechanics. LEFM provides a way to estimate the displacement and stress fields around the crack tip for brittle materials. To predict how the crack propagates, the stress intensity factors are needed. These can be accurately calculated using the J-integral method, which is well suited for computations over a finite element mesh.

XFEM is a versatile framework that allows the enrichment of the finite dimensional approximation with any problem specific function. By doing so, various non-polynomial solutions can be sought, as long as some information about the approximation field is known. In the case of a cracked body, the jumps in the displacement field and the near tip displacement field are modeled by enriching with the Heaviside and asymptotic tip functions respectively.

Discretizing the governing equations and arriving at the linear system describing the equilibrium is similar to traditional FEM. A novelty of XFEM is the introduction of artificial degrees of freedom for each enrichment function. In turn this results in the stiffness matrices and force vectors containing terms related to these dofs, after integrating the enriched basis functions. Another important difference is the need for special integration rules, since the integrands may contain discontinuous or singular terms.

XFEM removes the burden of needing a mesh that conforms to the crack interface, at the cost of introducing some degree of coupling between them. To implement XFEM for crack propagation problems, a significant amount of crack geometry-mesh interactions are required. Modeling the crack is performed using an explicit or implicit approach. Explicit crack descriptions tend to be more accurate, while implicit ones couple naturally with XFEM and are more efficient computationally.

XFEM has the potential to produce highly accurate solutions, without significantly deviating from the well known workflow of tradition FEM. Nevertheless, attention to some details is important, as concluded from the numerical examples. Selecting the appropriate mesh size needs to be considered alongside the selection of the crack propagation length and the radius of the J-integral domain. Although numerical integration is crucial to achieve the required accuracy, applying one of the specialized integration schemes is quite straightforward. Refining the mesh does indeed result in better approximations, however the convergence rate is often lower than expected.

For the needs of this thesis, XFEM and the crack propagation module has been in implemented in C# code. Object oriented programming is closer to the real domain and affords a high level of abstraction for all the details involved in XFEM and mesh-geometry interactions. This code is freely available as part of the open source MSolve software [36] for structural analysis and design, developed by the Institute of Structural Analysis and Antiseismic Research at the National Technical University of Athens.

7.2 Lines of future work

XFEM has met with increasing interest over the past years and has been applied successfully to various problems. The current code implements XFEM for crack propagation problems covering all basic aspects. Nevertheless, it has to be extended to accommodate more advanced cases.

3D problems: Extending the XFEM formulation, including the integration schemes, to three dimensions is straightforward. Some modification to the J-integral method are necessary, but the main hurdle is how to accurately and efficiently describe the interaction of 3D crack geometries with the finite element mesh. Although implicit methods are far easier to extended to 3D, special care must be taken to mitigate their many inherent inaccuracies.

Convergence: Convergence analysis of XFEM is more complicated than traditional FEM. In general, h-refinement (using finer meshes) exhibits slower convergence than expected. This can be attributed mainly to blending elements, that is elements where only some of the nodes are enriched and the partition of unity no longer holds. Different blending strategies need to be considered and implemented. Additionally, this thesis only dealt with first order finite element. Using higher order elements usually results in an increase of the error introduced by blending elements and reduced convergence rate. An approach to be considered is enriching only the nodes used in the first order interpolation (for enriched elements), while the standard shape functions are polynomials of higher degree.

Other fracture mechanics models: The fracture model presented in Chapter 3 is based on LEFM. LEFM is suitable for brittle material where the nonlinear zone ahead of the crack tip is negligible. As can be seen in Figure 7.1a, this zone is dominated by the singularity in the stress fields. Moreover, the crack faces are traction-free over their entire length.

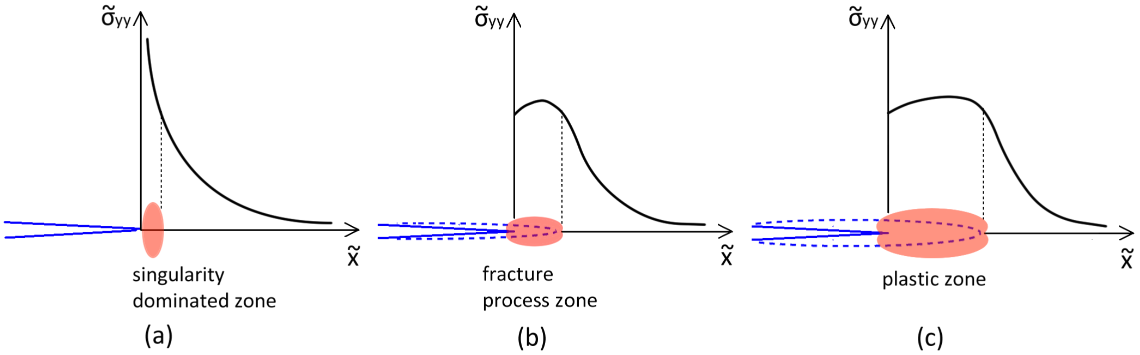


Fig. 7.1 Fracture models: a) Brittle fracture, b) Cohesive fracture, c) Ductile fracture

This model cannot accurately describe the behavior of quasi-brittle materials, such as concrete and geomaterials. In this case, the region in front of the crack tip is called fracture process zone (FPZ) and can no longer be ignored. The material exhibits nonlinear behavior in the FPZ, due to plasticity and microcracking, thus requiring a cohesive constitutive relation to properly model fracture phenomena. In the cohesive model, the damaged material can still still transfer stresses in the FPZ and the characteristic singularity of LEFM is avoided, as illustrated in Figure 7.1b. Another key difference is that cohesive forces are developed between the two crack faces, therefore a traction separation law is needed for the FPZ.

Crack propagation in ductile materials, such as steel, progresses according to Ductile Fracture Mechanics. In this case, fracture is preceded by large plastic deformations and the plastic energy release rate near the crack tip is not negligible as in LEFM. In Ductile Fracture Mechanics, the FPZ grows larger and a damage-plasticity model is required to simulate the material behavior. The damage-plasticity model can be applied over the whole domain or it can be localized at the FPZ, but the latter typically results in ill-posed governing equations. At any rate, both material and geometric nonlinearities must be considered, due to plastic constitutive laws and large deformations respectively.

Finally, a crack tends to branch as it propagates in reality. This can be modeled in XFEM using junction enrichment functions. Furthermore, bimaterial crack propagation problems can be solved with minor modifications using the XFEM framework. To do so, the crack is modeled using the Heaviside and tip functions, while the bimaterial interface can be modeled with the ramp function. The ramp function $\psi(\mathbf{x}) = \left|\varphi(\mathbf{x})\right|$ introduces a jump in the strain/stress field, while the displacement field remains continuous across the bimaterial interface.

References

[1] Inglis CE. “Stresses in a plate due to the presence of cracks and sharp corners”. In: *Transactions of Institute of Naval Architects* 55 (1913), pp. 219–241.

[2] Griffith AA. “The Phenomena of rupture and flow in solids”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering* 221 (1921), pp. 163–198.

[3] Westergard HM. “Bearing pressure and cracks”. In: *Journal of Applied Mechanics* 6 (1939), pp. 49–53.

[4] Eshelby JD. “The continuum theory of lattice defects”. In: *Solid State Physics* 3 (1956), pp. 79–141.

[5] Irwin GR. “Analysis of stresses and strains near the end of a crack traversing a plate”. In: *Journal of Applied Mechanics* 24 (1957), pp. 361–364.

[6] Irwin GR, Kies GA, and Smith HL. “Fracture strength relative to the onset and arrest of crack propagation”. In: *Proceedings of ASTM* 58 (1958), pp. 640–657.

[7] Erdogan F and Sih GC. “On the crack extension in plates under plane loading and transverse shear”. In: *Journal of Basic Engineering* 85.4 (1963), p. 519.

[8] Paris P and Erdogan F. “A critical analysis of crack propagation laws”. In: *Journal of Basic Engineering, Transactions of the American Society of Mechanical Engineers* (1963), pp. 528–534.

[9] Rice JR. “Path-independent integral and the approximate analysis of strain concentration by notches and cracks”. In: *Journal of Applied Mechanics, Transactions ASME* 35 (1968), pp. 379–386.

[10] Sih GC. “Strain-energy-density factor applied to mixed mode crack problems”. In: *International Journal of Fracture* 10.3 (1974), pp. 305–321.

[11] Tanaka K. “Fatigue crack propagation from a crack inclined to the cyclic tensile axis”. In: *Engineering Fracture Mechanics* 6.3 (1974), pp. 493–507.

[12] Atluri SN. “Path-independent integrals in finite elasticity and inelasticity, with body forces, inertiam and arbitrary crack-face conditions”. In: *Engineering Fracture Mechanics* 16 (1982), pp. 341–364.

[13] Li FZ, Shih CF, and Needleman A. “A comparison of methods for calculating energy release rates”. In: *Engineering Fracture Mechanics* 21.2 (1985), pp. 405–421.

[14] Broek D. *Elementary engineering fracture mechanics*. Dordrecht: Springer Netherlands, 1986.

[15] Osher S and Sethian JA. “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations”. In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49.

[16] Chew LP. “Constrained delaunay triangulations”. In: *Algorithmica* 4.1-4 (1989), pp. 97–108.

[17] Hua C. “An inverse transformation for quadrilateral isoparametric elements: Analysis and application”. In: *Finite Elements in Analysis and Design* 7.2 (1990), pp. 159–166.

[18] Sumi Y, Yang C, and Wang ZN. “Morphological aspects of fatigue crack propagation, Part II — effects of stress biaxiality and welding residual stress”. In: *International Journal of Fracture* 82.3 (1996), pp. 221–235.

[19] Dolbow JE. “An extended finite element method with discontinuous enrichment for applied mechanics”. PhD thesis. The address of the publisher: Northwestern University, USA, 1999.

[20] Moes N, Dolbow J, and Belytschko T. “A finite element method for crack growth without remeshing”. In: *International Journal for Numerical Methods in Engineering* 46.1 (1999), pp. 131–150.

[21] Belytschko T and Black T. “Elastic crack growth in finite elements with minimal remeshing”. In: *International Journal For Numerical Methods In Engineering* 45.5 (1999), pp. 601–620.

[22] Daux C, Moes N, Dolbow J, Sukumar N, and Belytschko T. “Arbitrary branched and intersecting cracks with the extended finite element method”. In: *International Journal for Numerical Methods in Engineering* 48.12 (2000), pp. 1741–1760.

[23] Lorentzon M and Eriksson K. “A path independent integral for the crack extension force of the circular arc crack”. In: *Engineering Fracture Mechanics* 66.5 (2000), pp. 423–439.

[24] Stolarska M, Chopp DL, Moës N, and Belytschko T. “Modelling crack growth by level sets in the extended finite element method”. In: *International Journal for Numerical Methods in Engineering* 51.8 (2001), pp. 943–960.

[25] Sukumar N, Chopp DL, Moës N, and Belytschko T. “Modeling holes and inclusions by level sets in the extended finite-element method”. In: *Computer Methods in Applied Mechanics and Engineering* 190.46-47 (2001), pp. 6183–6200.

[26] Ventura G, Budyn E, and Belytschko T. “Vector level sets for description of propagating cracks in finite elements”. In: *International Journal for Numerical Methods in Engineering* 58.10 (2003), pp. 1571–1592.

[27] Liang J, Huang R, Prévost JH, and Suo Z. “Evolving crack patterns in thin films with the extended finite element method”. In: *International Journal of Solids and Structures* 40.10 (2003), pp. 2343–2354.

[28] Huang R, Sukumar N, and Prevost JH. “Modeling quasi-static crack growth with the extended finite element method, Part II: Numerical applications”. In: *International Journal of Solids and Structures* 40 (2003), pp. 7539–7552.

[29] Laborde P, Pommier J, Renard Y, and Salaün M. “High-order extended finite element method for cracked domains”. In: *International Journal for Numerical Methods in Engineering* 64.3 (2005), pp. 354–381.

[30] Duflot M. “A study of the representation of cracks with level sets”. In: *International Journal for Numerical Methods in Engineering* 70.11 (2007), pp. 1261–1302.

[31] Khoei AR, Azadi H, and Moslemi H. “Modeling of crack propagation via an automatic adaptive mesh refinement based on modified superconvergent patch recovery technique”. In: *Engineering Fracture Mechanics* 75.10 (2008), pp. 2921–2945.

[32] Fries TP and Belytschko T. “The extended/generalized finite element method: An overview of the method and its applications”. In: *International Journal For Numerical Methods In Engineering* 84.3 (2010), pp. 1–3.

[33] Mohammadi S. *XFEM fracture analysis of composites*. John Wiley and Sons, 2012.

[34] Khoei AR. *Extended Finite Element Method: Theory and applications*. Wiley Series In Computational Mechanics. John Wiley and Sons, 2015.

[35] Geuzaine C and Remacle JF. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. url: http://gmsh.info.

[36] NTUA ISAAR. *MSolve*. url: https://github.com/isaar/MSolve.

Appendix A

Coordinate systems and transformations

There are four coordinate systems involved in crack propagation analysis with XFEM:

- A global cartesian system, which is the original system where all physical quantities are defined. Its coordinates will be denoted as $\mathbf{x} = (x, y)$.

- A natural system for each finite element, which is used in the isoparametric formulation. Its coordinates will be denoted as $\boldsymbol{\xi} = (\xi, \eta)$.

- A local cartesian system defined at the crack tip, whose $\tilde{x}$ axis is aligned with the crack and points outwards. Its coordinates will be denoted as $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$.

- A local polar system defined at the crack tip, which is a polar transformation of the tip cartesian system listed above. Its coordinates will be denoted as $\mathbf{r} = (r, \theta)$.

This appendix presents all necessary transformations of points and fields between those systems.

A.1 Isoparametric formulation

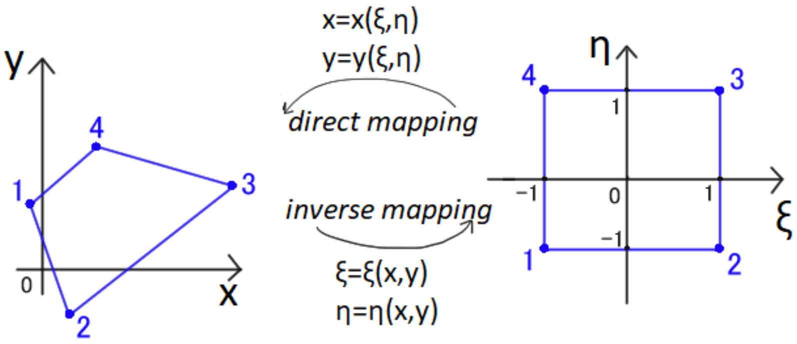


Fig. A.1 The isoparametric mapping for a quadrilateral element with 4 nodes. Left: the global cartesian coordinate system. Right: the element's natural coordinate system.

A.1.1 The isoparametring mapping

Figure A.1 displays a quadrilateral finite element with 4 nodes, the global cartesian and the element's natural coordinate system. The direct mapping is defined for transforming the natural coordinates $\boldsymbol{\xi} = (\xi, \eta)$ of a point to global coordinates $\mathbf{x} = (x, y)$:

$$x = x(\boldsymbol{\xi}) = \sum_{i=1}^{n_{nodes}} N_i(\boldsymbol{\xi}) x_i \tag{A.1a}$$

$$y = y(\boldsymbol{\xi}) = \sum_{i=1}^{n_{nodes}} N_i(\boldsymbol{\xi}) y_i \tag{A.1b}$$

where

- n_{nodes} is the number of the element's nodes.
- x_i and y_i are the global coordinates of node i.
- $N_i(\boldsymbol{\xi})$ are the shape functions used for interpolating nodal values over the element's domain. They are drawn from the space of Lagrange polynomials.
- The element's nodes must be numbered either clockwise or anti-clockwise in both coordinate systems. Here an anti-clockwise numbering is adopted.

The shape functions of the 4-noded quadrilateral element (Quad4) depicted in Figure A.1 are:

$$N_1(\boldsymbol{\xi}) = \frac{1}{4}(1-\xi)(1-\eta) \tag{A.2a}$$

$$N_2(\boldsymbol{\xi}) = \frac{1}{4}(1+\xi)(1-\eta) \tag{A.2b}$$

$$N_3(\boldsymbol{\xi}) = \frac{1}{4}(1+\xi)(1+\eta) \tag{A.2c}$$

$$N_4(\boldsymbol{\xi}) = \frac{1}{4}(1-\xi)(1+\eta) \tag{A.2d}$$

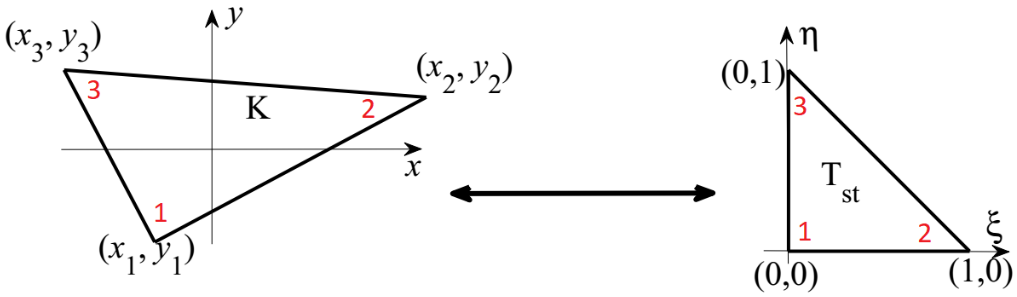


Fig. A.2 The isoparametric mapping for a triangular element with 3 nodes. Left: the global cartesian coordinate system. Right: the element's natural coordinate system.

The shape functions of the 3-noded triangular element (Tri3) depicted in Figure A.2 are:

$$N_1(\boldsymbol{\xi}) = 1-\xi-\eta \tag{A.3a}$$

$$N_2(\boldsymbol{\xi}) = \xi \tag{A.3b}$$

$$N_3(\boldsymbol{\xi}) = \eta \tag{A.3c}$$

Note that for both Quad4 and Tri3 the shape functions are (bi)linear functions of the natural coordinates, hence the terms linear or first order finite elements.

A.1.2 Jacobian of the isoparametric mapping

The Jacobian matrix of the direct mapping is

$$\mathbf{J}_{NG} = \begin{bmatrix} x,_\xi & x,_\eta \\ y,_\xi & y,_\eta \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{n_{nodes}} N_{i},_\xi(\boldsymbol{\xi})\,x_i & \sum\limits_{i=1}^{n_{nodes}} N_{i},_\eta(\boldsymbol{\xi})\,x_i \\ \sum\limits_{i=1}^{n_{nodes}} N_{i},_\xi(\boldsymbol{\xi})\,y_i & \sum\limits_{i=1}^{n_{nodes}} N_{i},_\eta(\boldsymbol{\xi})\,y_i \end{bmatrix} \tag{A.4}$$

where $f,_\xi$ and $f,_\eta$ denote the differentiation of f with respect to ξ and η respectively. Its determinant will be called Jacobian determinant henceforth and can be calculated as

$$\left|\mathbf{J}_{NG}\right| = x,_\xi\, y,_\eta - x,_\eta\, y,_\xi \tag{A.5}$$

The inverse mapping (global to natural coordinates) does not generally have analytic formulas. Even though the mapping itself is rarely used in classic FEM, its Jacobian is important and typically obtained by inversing $\mathbf{J}_{NG}$

$$\mathbf{J}_{GN} = \mathbf{J}_{NG}^{-1} = \begin{bmatrix} \xi,_x & \xi,_y \\ \eta,_x & \eta,_y \end{bmatrix} = \frac{1}{\left|\mathbf{J}_{NG}\right|} \begin{bmatrix} y,_\eta & -x,_\eta \\ -y,_\xi & x,_\xi \end{bmatrix} \tag{A.6}$$

and its determinant is

$$\left|\mathbf{J}_{GN}\right| = \frac{1}{\left|\mathbf{J}_{NG}\right|} \tag{A.7}$$

A.1.3 Transformation of fields and their derivatives

Scalar fields

A scalar field f is independent of the coordinate system used, thus there is no transformation necessary $f(\mathbf{x}) = f(\boldsymbol{\xi})$.

Scalar field derivatives

To differentiate a scalar field f with respect to the global coordinates x, y:

- If the analytic expression of $f(\mathbf{x})$ is known, simply calculate

$$_{\mathbf{x}}\nabla f(\mathbf{x}) = \begin{bmatrix} f,_x(\mathbf{x}) & f,_y(\mathbf{x}) \end{bmatrix} \tag{A.8}$$

  where $_{\mathbf{x}}\nabla$ denotes the gradient operator with respect to global coordinates

- Otherwise differentiate with respect to natural coordinates and apply the chain rule

$$\begin{aligned}\frac{\vartheta f}{\vartheta x} &= \frac{\vartheta f}{\vartheta \xi}\frac{\vartheta \xi}{\vartheta x} + \frac{\vartheta f}{\vartheta \eta}\frac{\vartheta \eta}{\vartheta x}\\ \frac{\vartheta f}{\vartheta y} &= \frac{\vartheta f}{\vartheta \xi}\frac{\vartheta \xi}{\vartheta y} + \frac{\vartheta f}{\vartheta \eta}\frac{\vartheta \eta}{\vartheta y}\end{aligned} \tag{A.9}$$

or

$$\begin{aligned} {}_{\mathbf{x}}\nabla f(\mathbf{x}) = \begin{bmatrix} f,_x(\mathbf{x}) & f,_y(\mathbf{x})\end{bmatrix} &= \begin{bmatrix} f,_\xi(\boldsymbol{\xi}) & f,_\eta(\boldsymbol{\xi})\end{bmatrix} \cdot \begin{bmatrix} \xi,_x & \xi,_y \\ \eta,_x & \eta,_y\end{bmatrix}\\ \Leftrightarrow\ {}_{\mathbf{x}}\nabla f(\mathbf{x}) &= {}_{\boldsymbol{\xi}}\nabla f(\boldsymbol{\xi}) \cdot \mathbf{J}_{NG}^{-1}\end{aligned} \tag{A.10}$$

As seen in the previous equations, the chain rule is conveniently written as a right multiplication with the inverse of the direct mapping's Jacobian matrix, given that the gradient is written as a row vector. This notation will be kept throughout the thesis.

Vector fields

Unlike scalar fields, the representation of a vector field **F** is defined with respect to a coordinate system. In fact, the components of a vector field are the scalar coefficients of the coordinate system's basis vectors. Therefore the same vector will have different representations (components) in different coordinate systems. Fortunately in the case of the isoparametric formulation, all vector fields of interest are typically represented only in the global cartesian system, so there is not any need to transform them. The following notation will be adopted:

- $\mathbf{F}(\mathbf{x})$ denotes the analytic expression with respect to the global coordinates that produces the global components of the vector field.

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_x(\mathbf{x}) \\ F_y(\mathbf{x}) \end{bmatrix} \tag{A.11}$$

- $\bar{\mathbf{F}}(\boldsymbol{\xi})$ denotes the analytic expression with respect to the natural coordinates that produces the natural components of the vector field. As stated before, this will not be used, since only the representation of the vector in the global system is needed.

- $\mathbf{F}(\boldsymbol{\xi})$ denotes the analytic expression with respect to the natural coordinates that produces the global components of the vector field. Often an analytic expression is known only with respect to the natural coordinates, even though the representation of the vector refers to the global cartesian system.

$$\mathbf{F}(\boldsymbol{\xi}) = \begin{bmatrix} F_x(\boldsymbol{\xi}) \\ F_y(\boldsymbol{\xi}) \end{bmatrix} \tag{A.12}$$

Vector field derivatives

The gradients of a vector field with respect to (w.r.t. in short) the two coordinate systems are

- Global cartesian components and derivatives w.r.t global cartesian coordinates:

$$_{\mathbf{x}}\nabla\mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_{x,x}(\mathbf{x}) & F_{x,y}(\mathbf{x}) \\ F_{y,x}(\mathbf{x}) & F_{y,y}(\mathbf{x}) \end{bmatrix} \tag{A.13}$$

- Global cartesian components but derivatives w.r.t natural coordinates:

$$_{\xi}\nabla\mathbf{F}(\boldsymbol{\xi}) = \begin{bmatrix} F_{x,\xi}(\boldsymbol{\xi}) & F_{x,\eta}(\boldsymbol{\xi}) \\ F_{y,\xi}(\boldsymbol{\xi}) & F_{y,\eta}(\boldsymbol{\xi}) \end{bmatrix} \tag{A.14}$$

Contrary to the field itself, the gradient of a vector field must be transformed between the two coordinate systems if its analytic expression in the desired system is unknown. To transform the gradient of a vector field from the natural to the global system, the chain rule is applied again. Note that each row contains the derivatives of the same vector component. This is similar to scalar field gradients being written as row vectors. Therefore the chain rule can be implemented by right multiplying each row of the vector field's gradient with the inverse of the (direct) mapping's Jacobian matrix.

$$\begin{aligned} _{\mathbf{x}}\nabla\mathbf{F}(\mathbf{x}) &= {}_{\xi}\nabla\mathbf{F}(\boldsymbol{\xi}) \cdot \mathbf{J}_{NG}^{-1} \\ \Leftrightarrow \begin{bmatrix} F_{x,x}(\mathbf{x}) & F_{x,y}(\mathbf{x}) \\ F_{y,x}(\mathbf{x}) & F_{2,y}(\mathbf{x}) \end{bmatrix} &= \begin{bmatrix} F_{x,\xi}(\boldsymbol{\xi}) & F_{x,\eta}(\boldsymbol{\xi}) \\ F_{y,\xi}(\boldsymbol{\xi}) & F_{y,\eta}(\boldsymbol{\xi}) \end{bmatrix} \cdot \begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix} \end{aligned} \tag{A.15}$$

Tensor fields

The two tensors that appear most often are the strain and stress tensor. Of course the gradient of the displacement field is also a second order tensor, but it is covered in the previous section about vector fields. Similar to vectors, these tensors are represented (their components) in the *global* cartesian system. In addition, note that the strain tensor and indirectly the stress tensor are derived from the gradient of the displacement field with respect to the *global* cartesian coordinates. Thus there is no need to transform them between the two systems. One usually calculates the components of the strain or stress tensor in the global cartesian system, expressed as functions of the natural coordinates, and uses those (e.g. in the weak form, for stress recovery, etc.).

A.1.4 The inverse isoparametric mapping

The inverse mapping (global to natural coordinates) does not generally have analytic formulas. Unlike traditional FEM, in XFEM it may be used for some integration schemes (see Section 4.3). Typically numerical methods are used to obtain an approximation of the inverse transformation. However, for first order finite elements (3-node triangle, 4-node quadrilateral in 2D and 4-node tatrahedron, 8-node hexahedron in 3D) the direct mapping is linear and there exist unique inverse mappings with analytic formulas.

Inverse mapping of an isoparametric 3-noded triangular element

The inverse mapping of an isoparametric triangular element with 3 nodes can be easily derived by expanding the direct mapping and solving for the the natural coordinates $\boldsymbol{\xi} = (\xi, \eta)$:

$$x = (1 - \xi - \eta)x_1 + \xi x_2 + \eta x_3 \Leftrightarrow (x_2 - x_1)\xi + (x_3 - x_1)\eta = x - x_1 \tag{A.16a}$$

$$y = (1 - \xi - \eta)y_1 + \xi y_2 + \eta y_3 \Leftrightarrow (y_2 - y_1)\xi + (y_3 - y_1)\eta = y - y_1 \tag{A.16b}$$

Using Cramer's rule for this 2×2 linear system

$$\xi = \frac{D_\xi}{D} \tag{A.17a}$$

$$\eta = \frac{D_\eta}{D} \tag{A.17b}$$

$$D = \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix} \tag{A.17c}$$

$$D_\xi = \begin{vmatrix} x - x_1 & x_3 - x_1 \\ y - y_1 & y_3 - y_1 \end{vmatrix} \tag{A.17d}$$

$$D_\eta = \begin{vmatrix} x_2 - x_1 & x - x_1 \\ y_2 - y_1 & y - y_1 \end{vmatrix} \tag{A.17e}$$

Therefore

$$\xi = \frac{(x-x_1)(y_3-y_1)-(x_3-x_1)(y-y_1)}{(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)} \tag{A.18a}$$

$$\eta = \frac{(x_2-x_1)(y-y_1)-(x-x_1)(y_2-y_1)}{(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)} \tag{A.18b}$$

Note that above $D \neq 0$ is assumed. This is the case for any non degenerate triangle, since

- No two vertices coincide (precondition 1)
- The three vertices do not lie on the same line (precondition 2)

Suppose $D = 0 \Leftrightarrow (x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1) = 0$

- Case 1: $x_2 - x_1 \neq 0$ AND $x_3 - x_1 \neq 0$

$$D = 0 \Leftrightarrow \frac{y_3-y_1}{x_3-x_1} = \frac{y_2-y_1}{x_2-x_1} \Rightarrow (\mathbf{x}_3-\mathbf{x}_1) \parallel (\mathbf{x}_2-\mathbf{x}_1) \quad \text{which violates precondition 2}$$

- Case 2: $x_2 - x_1 = 0$

$$D = 0 \Leftrightarrow (x_3-x_1)(y_2-y_1) = 0 \Leftrightarrow \begin{cases} x_3 - x_1 = 0 & \text{which violates precondition 2} \\ \quad OR & \\ y_2 - y_1 = 0 & \text{which violates precondition 1} \end{cases}$$

- Case 3: $x_3 - x_1 = 0$

$$D = 0 \Leftrightarrow (x_2-x_1)(y_3-y_1) = 0 \Leftrightarrow \begin{cases} x_2 - x_1 = 0 & \text{which violates precondition 2} \\ \quad OR & \\ y_3 - y_1 = 0 & \text{which violates precondition 1} \end{cases}$$

Inverse mapping of an isoparametric 4-noded quadrilateral element

The inverse mapping of an isoparametric quadrilateral element with 4 nodes (Quad4) is developed by Hua. Here only the necessary formulas will be presented. For their detailed derivation, the reader is referred to [17]. To transform the global coordinates $\mathbf{x} = (x, y)$ of a point to the natural coordinates $\boldsymbol{\xi} = (\xi, \eta)$:

1. Calculate the following coefficients from the global coordinates of the point and the element's nodes

$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \tag{A.19a}$$

$$d_1 = 4x - (x_1 + x_2 + x_3 + x_4) \tag{A.19b}$$

$$d_2 = 4y - (y_1 + y_2 + y_3 + y_4) \tag{A.19c}$$

2. Calculate the following (2×2) determinants

$$a_b = -b_a = a_1 \cdot b_2 - a_2 \cdot b_1 \tag{A.20a}$$

$$a_c = a_1 \cdot c_2 - a_2 \cdot c_1 \tag{A.20b}$$

$$a_d = -d_a = a_1 \cdot d_2 - a_2 \cdot d_1 \tag{A.20c}$$

$$b_c = -c_b = b_1 \cdot c_2 - b_2 \cdot c_1 \tag{A.20d}$$

$$b_d = -d_b = b_1 \cdot d_2 - b_2 \cdot d_1 \tag{A.20e}$$

$$d_c = d_1 \cdot c_2 - d_2 \cdot c_1 \tag{A.20f}$$

$$\tag{A.20g}$$

3. Depending on the above coefficients and determinants, refer to Table A.1, in order to calculate the natural coordinates (ξ, η)

Case	Condition	Solution
1	$a_1 a_2 a_b a_c \neq 0$	ξ is the solution of: $a_b\xi^2 + (c_b + d_a)\xi + d_c = 0$ that lies in the interval [-1.0, 1.0] $\eta = (a_d + b_a\xi)/a_c$
2	$a_1 = 0$ AND $a_2 c_1 \neq 0$	
3	$a_2 = 0$ AND $a_1 b_2 \neq 0$	
4	$a_1 a_2 \neq 0$ AND $a_b = 0$	$\xi = (a_1 d_c)/(b_1 a_c + a_1 a_d)$ $\eta = a_d/a_c$
5	$a_1 a_2 \neq 0$ AND $a_b = 0$	$\xi = a_d/a_b$ $\eta = (a_1 d_b)/(c_1 a_b + a_1 a_d)$
6	All other cases	$\xi = d_c/(a_1 d_2 + b_c)$ $\eta = b_d/(a_2 d_1 + b_c)$

Table A.1 Solutions of the inverse Quad 4 transformation.

Note that the shape functions (A.2) are assumed to implement the mapping illustrated in Figure A.1, where (ξ_1, η_1) is mapped to (x_1, y_1) and so forth. In addition the node numbering is anti-clockwise. If the nodes are numbered clockwise or if (ξ_1, η_1) is mapped to a node with different coordinates than (x_1, y_1) (and so forth), then (A.19) needs to be modified too.

A.2 Crack tip coordinate systems

At each crack tip a local cartesian system and its corresponding local polar system are defined, such that:

- The local $\tilde{x}$ axis is parallel to the line tangent to the crack at the crack tip and oriented away from the crack.
- The local $\tilde{y}$ axis can be obtained by rotating the local $\tilde{x}$ axis 90^o counter-clockwise.
- $\alpha \in (-\pi, \pi]$ is the counter-clockwise angle from global x axis to local $\tilde{x}$ axis.

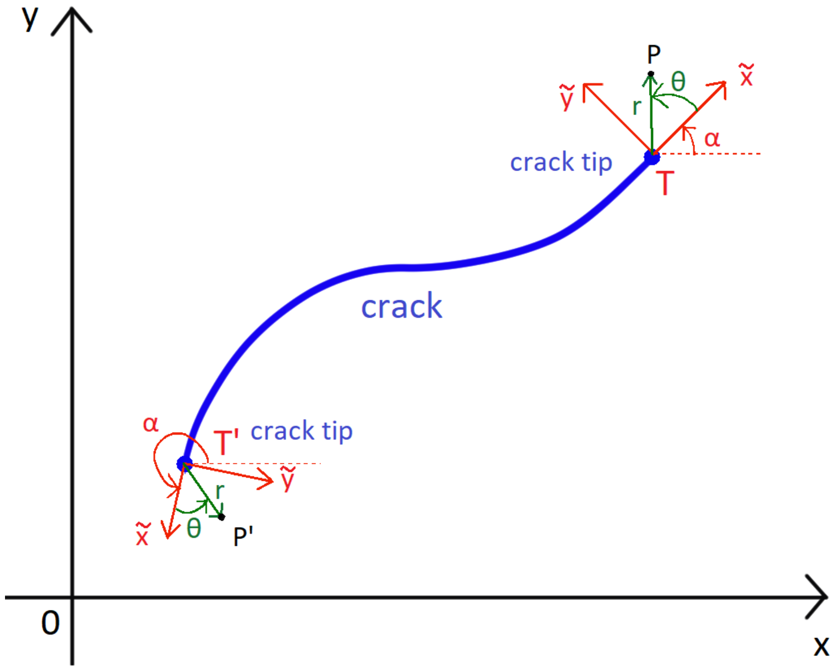


Fig. A.3 The local cartesian (red) and local polar (green) coordinate systems defined at each tip of an interior crack.

A.2.1 The local coordinate systems around the crack tip

To transform the *global* coordinates of a point to the *local cartesian* system of the tip, the following mapping is used

$$\tilde{\mathbf{x}} = \mathbf{Q} \cdot \mathbf{x} + \mathbf{b} \tag{A.21}$$

where

- $\mathbf{Q}$ is the following orthonormal rotation matrix:

$$\mathbf{Q} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \tag{A.22a}$$

$$\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{-1} \tag{A.22b}$$

- $\mathbf{b} = \begin{bmatrix} \tilde{x}_o \\ \tilde{y}_o \end{bmatrix}$ are the *local cartesian* coordinates of the *global* system's origin $O(0,0)$.

The inverse mapping, that is from the tip's *local cartesian* system to the *global cartesian* system, is defined similarly to (A.21)

$$\begin{aligned} &\mathbf{x} = \mathbf{Q}^{-1} \cdot \tilde{\mathbf{x}} + \mathbf{x}_T \\ \Leftrightarrow\ &\mathbf{x} = \mathbf{Q}^{T} \cdot \tilde{\mathbf{x}} + \mathbf{x}_T \end{aligned} \tag{A.23}$$

or it can be derived by multiplying (A.21) with $\mathbf{Q}^T$

$$\begin{aligned} &\mathbf{Q}^T \cdot \tilde{\mathbf{x}} = \mathbf{Q}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{Q}^T \cdot \mathbf{b} \\ \Leftrightarrow\ &\mathbf{Q}^{-1} \cdot \mathbf{Q} \cdot \mathbf{x} = \mathbf{Q}^T \cdot \tilde{\mathbf{x}} - \mathbf{Q}^T \cdot \mathbf{b} \\ \Leftrightarrow\ &\mathbf{x} = \mathbf{Q}^T \cdot \tilde{\mathbf{x}} - \mathbf{Q}^T \cdot \mathbf{b} \end{aligned} \tag{A.24}$$

where $\mathbf{x}_T = \begin{bmatrix} x_T \\ y_T \end{bmatrix}$ are the *global* coordinates of the *local cartesian* system's origin, i.e. the crack tip T, which are *known*. Thus $\mathbf{b}$ can now be obtained from

$$\begin{aligned} &-\mathbf{Q}^T \cdot \mathbf{b} = \mathbf{x}_T \\ \Leftrightarrow\ &\mathbf{b} = -\mathbf{Q} \cdot \mathbf{x}_T \end{aligned} \tag{A.25}$$

The local polar coordinate system defined at the crack tip uses the usual polar conversion. Define

- The direct mapping: from the *local polar* to the *local cartesian* coordinate system

$$\tilde{x} = r \cdot cos\theta \tag{A.26a}$$

$$\tilde{y} = r \cdot sin\theta \tag{A.26b}$$

- The inverse mapping: from the *local cartesian* to the *local polar* coordinate system

$$r = \sqrt{\tilde{x}^2 + \tilde{y}^2} \quad \in [0, \infty) \tag{A.27a}$$

$$\theta = \text{atan2}\left(\frac{\tilde{y}}{\tilde{x}}\right) \quad \in (-\pi, \pi] \tag{A.27b}$$

where atan2(y/x) is a common variation of the arctangent function, that returns angles covering a whole circle and is provided in most programming languages.

A.2.2 Jacobians of the mappings between crack tip systems

The Jacobian matrix of the direct mapping from *global* to *local cartesian* coordinate system is

$$\mathbf{J}_{GL} = \begin{bmatrix} \tilde{x},_x & \tilde{x},_y \\ \tilde{y},_x & \tilde{y},_y \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} = \mathbf{Q} \tag{A.28}$$

while the Jacobian matrix of the inverse mapping from *global* to *local cartesian* coordinate system is

$$\begin{aligned} \mathbf{J}_{LG} &= \begin{bmatrix} x,_{\tilde{x}} & x,_{\tilde{y}} \\ y,_{\tilde{x}} & y,_{\tilde{y}} \end{bmatrix} = \mathbf{J}_{GL}^{-1} = \mathbf{Q}^{-1} \\ \Leftrightarrow \mathbf{J}_{LG} &= \mathbf{J}_{GL}^{-1} = \mathbf{Q}^T = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \end{aligned} \tag{A.29}$$

and the determinants are

$$\left|\mathbf{J}_{GL}\right| = \left|\mathbf{J}_{LG}\right| = cos^2\alpha + sin^2\alpha = 1 \tag{A.30}$$

Note that these Jacobian matrices and their determinants are constants, since *a* is constant for each crack configuration. For all other mappings examined in this section and the previous one, the Jacobian matrices and their determinants depend on the coordinates of the point they are evaluated at.

The Jacobian matrix of the direct mapping from *local polar* to *local cartesian* coordinate system is

$$\mathbf{J}_{PL} = \begin{bmatrix} \tilde{x},_r & \tilde{x},_\theta \\ \tilde{y},_r & \tilde{y},_\theta \end{bmatrix} = \begin{bmatrix} cos\theta & -rsin\theta \\ sin\theta & rcos\theta \end{bmatrix} \tag{A.31}$$

and its determinant is

$$\left|\mathbf{J}_{PL}\right| = r\left(cos^2\theta + sin^2\theta\right) = r \tag{A.32}$$

The Jacobian matrix of the inverse mapping from *local cartesian* to *local polar* coordinate system can be calculated by inversing $\mathbf{J}_{PL}$

$$\begin{aligned}\mathbf{J}_{LP} &= \begin{bmatrix} r,_{\tilde{x}} & r,_{\tilde{y}} \\ \theta,_{\tilde{x}} & \theta,_{\tilde{y}} \end{bmatrix} = \mathbf{J}_{PL}^{-1} = \frac{1}{\left|\mathbf{J}_{PL}\right|} \begin{bmatrix} rcos\theta & -(-rsin\theta) \\ -sin\theta & cos\theta \end{bmatrix} \\ \Leftrightarrow \mathbf{J}_{LP} &= \mathbf{J}_{PL}^{-1} = \begin{bmatrix} cos\theta & sin\theta \\ \dfrac{-sin\theta}{r} & \dfrac{cos\theta}{r} \end{bmatrix}\end{aligned} \tag{A.33}$$

and its determinant is

$$\left|\mathbf{J}_{LP}\right| = \frac{1}{\left|\mathbf{J}_{PL}\right|} = \frac{1}{r} \tag{A.34}$$

A.2.3 Transformation of fields and their derivatives

Only the transformations that are actually applied will be presented here.

Scalar fields

A scalar field f is independent of the coordinate system used, thus there is no transformation necessary $f(\mathbf{x}) = f(\tilde{\mathbf{x}}) = f(\mathbf{r})$.

Scalar field derivatives

Let

- ${}_{\mathbf{x}}\nabla f(\mathbf{x}) = \begin{bmatrix} f,_x(\mathbf{x}) & f,_y(\mathbf{x}) \end{bmatrix}$
- ${}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) = \begin{bmatrix} f,_{\tilde{x}}(\tilde{\mathbf{x}}) & f,_{\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix}$
- ${}_{\mathbf{r}}\nabla f(\mathbf{r}) = \begin{bmatrix} f,_r(\mathbf{r}) & f,_\theta(\mathbf{r}) \end{bmatrix}$

be the gradients (row vectors as before) of the scalar field with respect to the global cartesian, local cartesian and local polar coordinates accordingly. To convert the gradient between two of these systems, the chain rule is applied or equivalently a multiplication with the Jacobian matrix of the inverse mapping (see Appendix A.1.3):

- Global cartesian to local cartesian $\left({}_{\mathbf{x}}\nabla f(\mathbf{x}) \rightarrow {}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) \right)$. Multiply with $\mathbf{J}_{GL}^{-1} = \mathbf{Q}^T$:

$$\begin{aligned} {}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) &= {}_{\mathbf{x}}\nabla f(\mathbf{x}) \cdot \mathbf{J}_{GL}^{-1} \\ \Leftrightarrow \begin{bmatrix} f_{,\tilde{x}}(\tilde{\mathbf{x}}) & f_{,\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} &= \begin{bmatrix} f_{,x}(\mathbf{x}) & f_{,y}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \end{aligned} \tag{A.35}$$

- Local cartesian to global cartesian $\left({}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) \rightarrow {}_{\mathbf{x}}\nabla f(\mathbf{x}) \right)$. Multiply with $\mathbf{J}_{LG}^{-1} = \mathbf{J}_{GL} = \mathbf{Q}$:

$$\begin{aligned} {}_{\mathbf{x}}\nabla f(\mathbf{x}) &= {}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) \cdot \mathbf{J}_{LG}^{-1} \\ \Leftrightarrow \begin{bmatrix} f_{,x}(\mathbf{x}) & f_{,y}(\mathbf{x}) \end{bmatrix} &= \begin{bmatrix} f_{,\tilde{x}}(\tilde{\mathbf{x}}) & f_{,\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \end{aligned} \tag{A.36}$$

- Local polar to local cartesian $\left({}_{\mathbf{r}}\nabla f(\mathbf{r}) \rightarrow {}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) \right)$. Multiply with $\mathbf{J}_{PL}^{-1}$:

$$\begin{aligned} {}_{\tilde{\mathbf{x}}}\nabla f(\tilde{\mathbf{x}}) &= {}_{\mathbf{r}}\nabla f(\mathbf{r}) \cdot \mathbf{J}_{PL}^{-1} \\ \Leftrightarrow \begin{bmatrix} f_{,\tilde{x}}(\tilde{\mathbf{x}}) & f_{,\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} &= \begin{bmatrix} f_{,r}(\mathbf{r}) & f_{,\theta}(\mathbf{r}) \end{bmatrix} \cdot \begin{bmatrix} cos\theta & sin\theta \\ \frac{-sin\theta}{r} & \frac{cos\theta}{r} \end{bmatrix} \end{aligned} \tag{A.37}$$

- Local polar to global cartesian $\left({}_{\mathbf{r}}\nabla f(\mathbf{r}) \rightarrow {}_{\mathbf{x}}\nabla f(\mathbf{x}) \right)$. A useful shortcut:

$$\begin{aligned} {}_{\mathbf{x}}\nabla f(\mathbf{x}) &= {}_{\mathbf{r}}\nabla f(\mathbf{r}) \cdot \mathbf{J}_{PL}^{-1} \cdot \mathbf{J}_{LG}^{-1} \\ \Leftrightarrow \begin{bmatrix} f_{,x}(\mathbf{x}) & f_{,y}(\mathbf{x}) \end{bmatrix} &= \begin{bmatrix} f_{,r}(\mathbf{r}) & f_{,\theta}(\mathbf{r}) \end{bmatrix} \cdot \begin{bmatrix} cos\theta & sin\theta \\ \frac{-sin\theta}{r} & \frac{cos\theta}{r} \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} f_{,x}(\mathbf{x}) & f_{,y}(\mathbf{x}) \end{bmatrix} &= \begin{bmatrix} f_{,r}(\mathbf{r}) & f_{,\theta}(\mathbf{r}) \end{bmatrix} \cdot \begin{bmatrix} cos\alpha \cdot cos\theta - sin\alpha \cdot sin\theta & sin\alpha \cdot cos\theta + cos\alpha \cdot sin\theta \\ -cos\alpha\frac{sin\theta}{r} - sin\alpha\frac{cos\theta}{r} & -sin\alpha\frac{sin\theta}{r} + cos\alpha\frac{cos\theta}{r} \end{bmatrix} \end{aligned} \tag{A.38}$$

Vectors

Let

- P_1 and P_2 be two points
- $P_1\left(x_1, y_1\right)$ and $P_2\left(x_2, y_2\right)$ be their representations in the global cartesian system

- $P_1\left(\tilde{x}_1, \tilde{y}_1\right)$ and $P_2\left(\tilde{x}_2, \tilde{y}_2\right)$ be their representations in the local cartesian system
- $\mathbf{v} = \overrightarrow{P_1 P_2}$ be a vector

The vector's components are defined with respect to the coordinates of its end points

- $\mathbf{v} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} = \mathbf{x}_2 - \mathbf{x}_1$ is its representation in the global cartesian system.
- $\tilde{\mathbf{v}} = \begin{bmatrix} \tilde{x}_2 - \tilde{x}_1 \\ \tilde{y}_2 - \tilde{y}_1 \end{bmatrix} = \tilde{\mathbf{x}}_2 - \tilde{\mathbf{x}}_1$ is its representation in the local cartesian system.

To convert the vector's components from the local cartesian to the global cartesian system, we need to convert the coordinates of its end points with (A.21):

$$\begin{aligned} \tilde{\mathbf{v}} = \tilde{\mathbf{x}}_2 - \tilde{\mathbf{x}}_1 = \mathbf{Q} \cdot \mathbf{x}_2 + \mathbf{b} - \left(\mathbf{Q} \cdot \mathbf{x}_1 + \mathbf{b}\right) &= \mathbf{Q} \cdot \left(\mathbf{x}_2 - \mathbf{x}_1\right) \\ \Leftrightarrow \tilde{\mathbf{v}} &= \mathbf{Q} \cdot \mathbf{v} \end{aligned} \tag{A.39}$$

Conversely, to transform the vector's components from the global cartesian to the local cartesian system

$$\mathbf{v} = \mathbf{Q}^T \cdot \tilde{\mathbf{v}} \tag{A.40}$$

Vector fields

Define $\mathbf{F}$ be a vector field and its representations:

- $\mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_x(\mathbf{x}) \\ F_y(\mathbf{x}) \end{bmatrix}$ are its components in the global cartesian coordinate system, expressed as functions of global cartesian coordinates.
- $\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{F}_x(\mathbf{x}) \\ \tilde{F}_y(\mathbf{x}) \end{bmatrix}$ are its components in the local cartesian coordinate system, expressed as functions of local cartesian coordinates.
- $\tilde{\mathbf{F}}(\mathbf{r}) = \begin{bmatrix} \tilde{F}_x(\mathbf{r}) \\ \tilde{F}_y(\mathbf{r}) \end{bmatrix}$ are its components in the local cartesian coordinate system, expressed as functions of local polar coordinates.

No vector of interest needs to be represented in the local polar system, however its analytic expression may only be known in the local polar system. This is similar to Appendix A.1.3. Since the components are identical we can write

$$\tilde{\mathbf{F}}(\mathbf{r}) = \tilde{\mathbf{F}}(\tilde{\mathbf{x}}) \tag{A.41}$$

On the other hand, the vector representation needs to be converted from the global cartesian to the local cartesian system, according to (A.39)

$$\begin{aligned} &\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) = \mathbf{Q} \cdot \mathbf{F}(\mathbf{x}) \\ \Leftrightarrow &\begin{bmatrix} \tilde{F}_x(\mathbf{x}) \\ \tilde{F}_y(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \cdot \begin{bmatrix} F_x(\mathbf{x}) \\ F_y(\mathbf{x}) \end{bmatrix} \end{aligned} \tag{A.42}$$

and from the local cartesian to the global cartesian system according to (A.40):

$$\begin{aligned} &\mathbf{F}(\mathbf{x}) = \mathbf{Q}^T \cdot \tilde{\mathbf{F}}(\tilde{\mathbf{x}}) \\ \Leftrightarrow &\begin{bmatrix} F_x(\mathbf{x}) \\ F_y(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \tilde{F}_x(\mathbf{x}) \\ \tilde{F}_y(\mathbf{x}) \end{bmatrix} \end{aligned} \tag{A.43}$$

Vector field derivatives

The gradients of the vector field involved are

- Global cartesian components, derivatives with respect to global cartesian coordinates (w.r.t in short):
$$_{\mathbf{x}}\nabla\mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_{x},_{x}(\mathbf{x}) & F_{x},_{y}(\mathbf{x}) \\ F_{y},_{x}(\mathbf{x}) & F_{y},_{y}(\mathbf{x}) \end{bmatrix} \tag{A.44}$$
- Local cartesian components, derivatives w.r.t. local cartesian coordinates:
$$_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{F}_{x},_{\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{x},_{\tilde{y}}(\tilde{\mathbf{x}}) \\ \tilde{F}_{y},_{\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{y},_{\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} \tag{A.45}$$
- Local cartesian components, derivatives w.r.t local polar coordinates:
$$_{\mathbf{r}}\nabla\tilde{\mathbf{F}}(\mathbf{r}) = \begin{bmatrix} \tilde{F}_{x},_{r}(\mathbf{r}) & \tilde{F}_{x},_{\theta}(\mathbf{r}) \\ \tilde{F}_{y},_{r}(\mathbf{r}) & \tilde{F}_{y},_{\theta}(\mathbf{r}) \end{bmatrix} \tag{A.46}$$

To convert the gradient of a vector field between two coordinate systems:

- Each row of the vector field's gradient contains the derivatives of the same component. To apply the chain rule, right multiply with the direct mapping's inverse Jacobian matrix.

- Each column of the vector field's gradient can be seen as a vector itself. To transform its components between to cartesian systems, left multiply with the appropriate rotation matrix.

The following two transformations are useful:

- ${}_{\mathbf{x}}\nabla\mathbf{F}(\mathbf{x}) \;\rightarrow\; {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{F}}(\tilde{\mathbf{x}})$:

$$
\begin{aligned}
& {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) = \mathbf{Q}\cdot\; {}_{\mathbf{x}}\nabla\mathbf{F}(\mathbf{x})\cdot\mathbf{J}_{GL}^{-1} \\
\Leftrightarrow & \begin{bmatrix} \tilde{F}_{x,\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{x,\tilde{y}}(\tilde{\mathbf{x}}) \\ \tilde{F}_{y,\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{y,\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix}\cdot\begin{bmatrix} F_{x,x}(\mathbf{x}) & F_{x,y}(\mathbf{x}) \\ F_{y,x}(\mathbf{x}) & F_{y,y}(\mathbf{x}) \end{bmatrix}\cdot\begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix}
\end{aligned}
\tag{A.47}
$$

- ${}_{\mathbf{r}}\nabla\tilde{\mathbf{F}}(\mathbf{r}) \;\rightarrow\; {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{F}}(\tilde{\mathbf{x}})$. Only the chain rule needs to be applied here, since the components of the two vector field gradients refer to the same coordinate system :

$$
\begin{aligned}
& {}_{\tilde{\mathbf{x}}}\nabla\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) = {}_{\mathbf{r}}\nabla\tilde{\mathbf{F}}(\mathbf{r})\cdot\mathbf{J}_{PL}^{-1} \\
\Leftrightarrow & \begin{bmatrix} \tilde{F}_{x,\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{x,\tilde{y}}(\tilde{\mathbf{x}}) \\ \tilde{F}_{y,\tilde{x}}(\tilde{\mathbf{x}}) & \tilde{F}_{y,\tilde{y}}(\tilde{\mathbf{x}}) \end{bmatrix} = \begin{bmatrix} \tilde{F}_{x,r}(\mathbf{r}) & \tilde{F}_{x,\theta}(\mathbf{r}) \\ \tilde{F}_{y,r}(\mathbf{r}) & \tilde{F}_{y,\theta}(\mathbf{r}) \end{bmatrix}\cdot\begin{bmatrix} cos\theta & sin\theta \\ \frac{-sin\theta}{r} & \frac{cos\theta}{r} \end{bmatrix}
\end{aligned}
\tag{A.48}
$$

Tensor fields

Assume a *symmetric* second order tensor $\mathbf{T}$. Its components in the global cartesian and local cartesian system are respectively

$$
\mathbf{T}(\mathbf{x}) = \begin{bmatrix} T_{xx}(\mathbf{x}) & T_{xy}(\mathbf{x}) \\ T_{yx}(\mathbf{x}) & T_{yy}(\mathbf{x}) \end{bmatrix} \tag{A.49a}
$$

$$
\tilde{\mathbf{T}}(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{T}_{xx}(\tilde{\mathbf{x}}) & \tilde{T}_{xy}(\tilde{\mathbf{x}}) \\ \tilde{T}_{yx}(\tilde{\mathbf{x}}) & \tilde{T}_{yy}(\tilde{\mathbf{x}}) \end{bmatrix} \tag{A.49b}
$$

The only such tensors of interest in our problem domain are the strain and stress tensor. Their components are always in the global cartesian or local cartesian system. Nevertheless, it is possible that analytic formulas for those components only exist w.r.t. local polar coordinates. Therefore, conversions are needed between the global cartesian and local cartesian sytems, but not between the local cartesian and local polar system.

The gradient of a vector field, such as the displacement field, is a second order tensor, albeit not a symmetric one. Equation (A.47) is also used for tensor conversions between global and local cartesian systems. Especially for symmetric tensors, (A.47) can be simplified into the well known formulas for rotating tensors:

$$\begin{aligned}
&\tilde{\mathbf{T}}(\tilde{\mathbf{x}}) = \mathbf{Q} \cdot \mathbf{T}(\mathbf{x}) \cdot \mathbf{J}_{GL}^{-1} \\
\Leftrightarrow &\begin{bmatrix} \tilde{T}_{11}(\tilde{\mathbf{x}}) & \tilde{T}_{xy}(\tilde{\mathbf{x}}) \\ \tilde{T}_{xy}(\tilde{\mathbf{x}}) & \tilde{T}_{yy}(\tilde{\mathbf{x}}) \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \cdot \begin{bmatrix} T_{xx}(\mathbf{x}) & T_{xy}(\mathbf{x}) \\ T_{xy}(\mathbf{x}) & T_{yy}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \\
\Leftrightarrow &\begin{bmatrix} \tilde{T}_{xx}(\tilde{\mathbf{x}}) & \tilde{T}_{xy}(\tilde{\mathbf{x}}) \\ \tilde{T}_{xy}(\tilde{\mathbf{x}}) & \tilde{T}_{yy}(\tilde{\mathbf{x}}) \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \cdot \begin{bmatrix} T_{xx}(\mathbf{x})cos\alpha + T_{xy}(\mathbf{x})sin\alpha & -T_{xx}(\mathbf{x})sin\alpha + T_{xy}(\mathbf{x})cos\alpha \\ T_{xy}(\mathbf{x})cos\alpha + T_{yy}(\mathbf{x})sin\alpha & -T_{xy}(\mathbf{x})sin\alpha + T_{yy}(\mathbf{x})cos\alpha \end{bmatrix}
\end{aligned} \tag{A.50}$$

These are equivalent to

$$\begin{aligned}
\tilde{T}_{xx}(\tilde{\mathbf{x}}) &= T_{xx}(\mathbf{x}) \cdot cos^2\alpha + T_{xy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha + T_{xy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha + T_{yy}(\mathbf{x}) \cdot sin^2\alpha \\
\tilde{T}_{xy}(\tilde{\mathbf{x}}) &= -T_{xx}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha + T_{xy}(\mathbf{x}) \cdot cos^2\alpha - T_{xy}(\mathbf{x}) \cdot sin^2\alpha + T_{yy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha \\
\tilde{T}_{xy}(\tilde{\mathbf{x}}) &= -T_{xx}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha - T_{xy}(\mathbf{x}) \cdot sin^2\alpha + T_{xy}(\mathbf{x}) \cdot cos^2\alpha + T_{yy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha \\
\tilde{T}_{yy}(\tilde{\mathbf{x}}) &= T_{xx}(\mathbf{x}) \cdot sin^2\alpha - T_{xy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha - T_{xy}(\mathbf{x}) \cdot cos\alpha \cdot sin\alpha + T_{yy}(\mathbf{x}) \cdot cos^2\alpha
\end{aligned} \tag{A.51}$$

By using the following trigonometric transformations

$$cos^2\alpha = \frac{1 + cos2\alpha}{2} \tag{A.52a}$$

$$sin^2\alpha = \frac{1 - cos2\alpha}{2} \tag{A.52b}$$

$$cos\alpha \cdot sin\alpha = \frac{sin2\alpha}{2} \tag{A.52c}$$

the well known formulas for transforming a tensor from the global to the local cartesian system are obtained

$$\begin{aligned}
\tilde{T}_{xx}(\tilde{\mathbf{x}}) &= \frac{T_{xx}(\mathbf{x}) + T_{yy}(\mathbf{x})}{2} + \frac{T_{xx}(\mathbf{x}) - T_2(\mathbf{x})}{yy} cos2\alpha + T_{xy}(\mathbf{x}) sin2\alpha \\
\tilde{T}_{yy}(\tilde{\mathbf{x}}) &= \frac{T_{xx}(\mathbf{x}) + T_{yy}(\mathbf{x})}{2} + \frac{T_{xx}(\mathbf{x}) - T_2(\mathbf{x})}{yy} cos2\alpha - T_{xy}(\mathbf{x}) sin2\alpha \\
\tilde{T}_{xy}(\tilde{\mathbf{x}}) &= \tilde{T}_{yx}(\tilde{\mathbf{x}}) = T_{xy}(\mathbf{x}) cos2\alpha - \frac{T_{xx}(\mathbf{x}) - T_{yy}(\mathbf{x})}{2} sin2\alpha
\end{aligned} \tag{A.53}$$