Adaptive goal-oriented finite element computation of the energy release rate at crack growth

Per Heintz

Gothenburg, Sweden 2001

Abstract

This thesis presents the theoretical framework for *goal-oriented adaptivity* in finite element computations. The motivation for this approach is that in engineering practice, the solution to the mathematical problem might not be the reason for making the computations. It is rather a quantity derived from the solution that is of interest. By specifying the goal-quantity a priori the corresponding dual problem is solved. An error representation in the chosen quantity is formulated and a set of high quality element indicators for the refinement of the mesh is obtained.

Goal-oriented adaptivity is performed on a classic problem in nonlinear elastic fracture mechanics; the *energy release rate* at crack growth. The solution has a singularity at the crack tip making localized resolution appropriate. This singularity is usually resolved by the use of an a priori constructed mesh and special singularity elements, or by using an energy formulation that does not involve the crack tip condition. How well this is accomplished in the adaptive routine is investigated.

Two different energy formulations, the original *J-Contour integral* and the *generalized energy release rate*, are used in the simulations. The results are verified by computing the stress intensity factor using an asymptotic displacement approach.

The results show that the goal-oriented algorithm as a flexible and robust general framework for FE computations is appropriate for this type of problem.

Keywords: goal-oriented adaptivity, J integral, energy release rate

Preface

This master thesis is the final requirement for achieving the Master of Science degree at Chalmers. The work has been carried out at the Department of Applied Mechanics during the summer and autumn 2001.

There are some people that contributed to the progress of my work who I especially would like to thank:

I address my deepest gratitude to my advisor Dr Klas Samuelsson at Chalmers Finite Element Center for his enthusiasm and constructive critique during this period. Klas also shared his tremendous experience about theoretical and implementation issues. For this I will always be thankful.

PhD student Fredrik Larsson at the Department of Applied Mechanics for explaining theoretical details.

My examiner Peter Hansbo at the Department of Applied Mechanics for his guidance.

Senior researchers at the Department of Applied Mechanics with experience in evaluating crack tip parameters whom I consulted at the end of this period. Special thanks to Peter Möller for the last minute tip on how to check my calculations.

From my heart, I thank my fiance Anna for her love and support during these 4.5 years at Chalmers.

This thesis consists of the following chapters:

Chapter 1: A short review of the Galerkin method for computing an approximative solution to Partial Differential Equations using piecewise continouos polynomials. The equations of linear and nonlinear elasticity are formulated.

Chapter 2: An introduction to adaptive methods and error control by solving a corresponding dual problem. The theoretical framework for goal-oriented adaptivity is presented.

Chapter 3: A description of the adaptive algorithm used in the computations.

Chapter 4: Introduction to (non)Linear Elastic Fracture Mechanics (LEFM). Overview of the considered integrals computed with FEM.

Chapter 5: The FE implementation and plots from the results.

Chapter 6: Conclusions

Appendix: The path independence of the J-contour integral is proven. The derivation of J as an energy release rate and the derivation of the generalized energy release rate.

Contents

Chapter 1

Galerkin's Method

Galerkin's method is a technique to solve Partial Differential Equations (PDE:s) on the form

$$A\boldsymbol{u} = \boldsymbol{b}, \tag{1.1}$$

where A is a differential operator and $\boldsymbol{b}$ is the data to the problem. This is accomplished by seeking an approximate solution $\boldsymbol{u_h}$ in a finite dimensional space V_h spanned by a set of basis functions $\{\boldsymbol{v}\}_{i=1}^{n}$. The subscript h indicates that the vector space V_h has finite dimension. By choosing a finite set of basis functions we are, by an orthogonality argument, led to a system of equations that we can solve numerically. The basis functions should be linearly independent and vanish on the part of the boundary where essential boundary conditions are prescribed. The exact solution $\boldsymbol{u} \in V$ will be substituted by $\boldsymbol{u_h} \in V_h$ which is our approximation. We express $\boldsymbol{u_h}$ in terms of our basis functions

$$\boldsymbol{u_h} = \boldsymbol{\psi} + \sum_{i=1}^{n} \alpha_i \boldsymbol{v_i}. \tag{1.2}$$

$\alpha_i \in R$ are weights and $\boldsymbol{v_i}$ our basis functions. $\boldsymbol{\psi}$ is a function that describes the non homogeneous Dirichlet boundary conditions and is needed because the basis functions are defined to be zero on boundaries with prescribed values. When $\{\boldsymbol{v}\}_{i=1}^{n}$ has been chosen we must find the function $\boldsymbol{u_h}$ closest to $\boldsymbol{u}$ in V_h. This is equivalent to determine the weights α_i in (1.2). This is what we achieve with *Galerkin orthogonality*. Let us recall the L_2 scalar product

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{u}\boldsymbol{v}\, d\Omega, \tag{1.3}$$

and remember that two functions $\boldsymbol{u}$ and $\boldsymbol{v}$ are orthogonal w.r.t. the L_2-norm if their L_2-scalar product is zero. The unknown function $\boldsymbol{u}$ will be approximated by FEM

$$A\boldsymbol{u} = \boldsymbol{b}, \tag{1.4}$$

$$\boldsymbol{b} - A\boldsymbol{u} = \mathbf{0}. \tag{1.5}$$

As we insert our approximate solution $\boldsymbol{u_h}$ in (1.5) it changes to

$$\boldsymbol{b} - A\boldsymbol{u_h} = \boldsymbol{R}. \tag{1.6}$$

The Galerkin orthogonality is now expressed such that the residual $\boldsymbol{R}$ is orthogonal, w.r.t. L_2 norm, to every function in our chosen space V_h, see Fig.(1.1)

$$\int_\Omega \boldsymbol{R}\boldsymbol{v}\, d\Omega = 0 \quad \forall \boldsymbol{v} \in V_h. \tag{1.7}$$

Note that the true solution satisfies a stronger orthogonality condition

$$\int_\Omega \boldsymbol{R}\boldsymbol{v}\, d\Omega = 0 \quad \forall \boldsymbol{v} \in V, \tag{1.8}$$

since $\boldsymbol{R} = \mathbf{0}$. The set of functions where we seek our approximation is referred to as the *trial space* and the space of functions used for the orthogonality condition is referred to as the *test space*. The trial and test space differ when

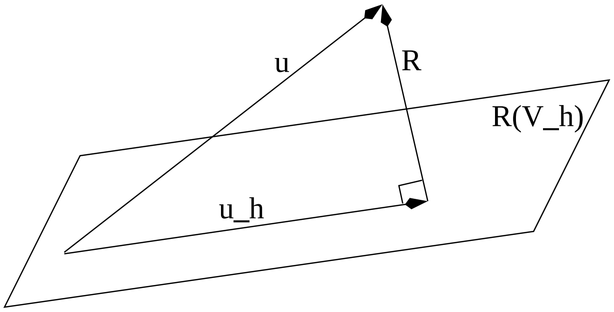

Figure 1.1: Graphical illustration of (1.7). $R(V_h) = \{\boldsymbol{y} \in V_h : A\boldsymbol{x} = \boldsymbol{y} \text{ for some } \boldsymbol{x} \in V_h\}$, $\boldsymbol{u}$ - the unknown solution in V, $\boldsymbol{u_h}$ - the approximate solution to be computed with FEM in V_h. $\boldsymbol{R}$ - Residual due to the approximation $\boldsymbol{R} \in V \ominus V_h$.

nonhomogeneous boundary conditions are present. The Galerkin method can, in a compact style, be expressed as

$$\text{Find } \boldsymbol{u_h} - \boldsymbol{\psi} \in V_h : \ (\boldsymbol{R}, \boldsymbol{v}) = \boldsymbol{0}. \tag{1.9}$$

If the operator A involves higher order derivatives, the Galerkin orthogonality is often rewritten in an equivalent form using integration by parts. The obtained corresponding form is called *weak formulation.*

The infinite space V, called Sobolev space, is different depending on the continuous problem. Typically

$$V = H_0^1 = \{\boldsymbol{v} : \int_\Omega \boldsymbol{v}^2 + |\nabla \boldsymbol{v}|^2 \, d\Omega < \infty, \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega_D\}, \tag{1.10}$$

for PDE:s involving the Laplace operator, $\Delta = \frac{\partial^2}{\partial x_i^2}$.

1.1 Linear Elasticity

We consider the deformation of an elastic continuum, where the fundamental equations are based on Newton's second law

$$\rho u_{i,tt} = \sigma_{ij,j} + f_i, \tag{1.11}$$

The term on the left hand side (LHS) represents the acceleration of the continuum. The first term on the right hand side (RHS) represents internal forces due to stresses and the last term represents body forces such as gravity. The acceleration term is very important in dynamical analysis but will be neglected here and we actually try to solve the vector PDE

$$-\sigma_{ij,j} = f_i. \tag{1.12}$$

This expression contains three equations and six unknowns (since σ_{ij} is symmetric) so we need more information to close the system of equations. Assuming isotropic elastic material the relation between the stress- and strain tensor can be expressed in Hookes generalized law

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu\varepsilon_{ij}. \tag{1.13}$$

λ and μ are the Lame's elasticity constants, which vary in space if the material is heterogeneous but are pure constants if the elastic material is homogeneous. The strain tensor is defined (neglecting thermal effects) as

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right). \tag{1.14}$$

(1.13) inserted into (1.12) yields a vector equation for the unknown displacement field $\boldsymbol{u}$.

$$((\lambda + \mu) u_{k,k})_{,i} + (\mu u_{i,k})_{,k} = f_i. \tag{1.15}$$

If Lame's coefficients are constant (1.15) changes to

$$(\lambda + \mu) u_{k,ki} + \mu u_{i,kk} = f_i. \tag{1.16}$$

The equations of elasticity are in mathematical style expressed, together with boundary conditions, as

$$\boldsymbol{\sigma} = \lambda(\boldsymbol{\nabla} \cdot \boldsymbol{u})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \qquad \text{in } \Omega, \tag{1.17}$$

$$-\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1.18}$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega_D, \tag{1.19}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} \qquad \text{on } \partial\Omega_N. \tag{1.20}$$

(1.17) and (1.18) are equivalent to (1.13) and (1.12). Equation (1.19) and (1.20) are boundary conditions that describes prescribed boundary movements and traction respectively.

1.2 Discretizing with Galerkin's Method

Applying Galerkin's method to (1.18) results in

$$\int_\Omega (-\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{v}\, d\Omega = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega. \tag{1.21}$$

Integration by part yields

$$\int_\Omega \boldsymbol{\sigma} \cdot \boldsymbol{\nabla} \boldsymbol{v}\, d\Omega - \int_{\Gamma_N} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{v}\; ds = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega. \tag{1.22}$$

Using the symmetry of $\boldsymbol{\sigma}$ and (1.20) we obtain the following weak form of the elasticity equations.

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega + \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v}\; ds. \tag{1.23}$$

The LHS physically represents internal strain energy while the RHS represents work done by applied body- and boundary forces.

We define the bilinear form $a(\boldsymbol{u}, \boldsymbol{v})$ and linear form $L(\boldsymbol{v})$ as

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega, && (1.24) \\ L(\boldsymbol{v}) &= \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega + \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v}\, ds. && (1.25) \end{aligned}$$

The variational formulation (VR) and the corresponding FEM are in abstract form written as

$$\begin{aligned} \text{Find } \boldsymbol{u} - \boldsymbol{\psi} \in V &: a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in V, && (1.26) \\ \text{Find } \boldsymbol{u_h} - \boldsymbol{\psi} \in V_h &: a(\boldsymbol{u_h}, \boldsymbol{v}) = L(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in V_h. && (1.27) \end{aligned}$$

V and V_h are defined as

$$\begin{aligned} V &= \{\boldsymbol{v} : \int_\Omega \boldsymbol{v}^2 + |\nabla \boldsymbol{v}|^2\, d\Omega < \infty, \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega_D\}, && (1.28) \\ V_h &= \{\boldsymbol{v} : \boldsymbol{v} \text{ c.p.w. in } \Omega, \ \ \boldsymbol{v} \in P^q(K), \ \ K \in T_h\}. && (1.29) \end{aligned}$$

$P^q(K)$ are continuous piecewise polynomials of maximum degree q defined on $K \in T_h$ (explained below).

The finite element discretization uses subspaces V_h of V defined on decompositions $T_h = \{K\}$ of Ω where $\{K\}$ is the set of non overlapping elements (the mesh). T_h is characterized by the largest side of an element in the mesh, $h := \max_{K \in T_h}\{h_K\}$ where $h_K := \text{diam}(K)$, i.e., the largest side of the element, see Fig.(1.2). The mesh size h is often used in a priori analysis where the limiting behavior of the discretization is expressed in terms of the exact solution.

A mathematically oriented approach to Galerkin's method and the numerical solution to PDEs using piecewise polynomial approximations is found in [7, Eriksson,Estep,Hansbo,Johnson].

1.3 Nonlinear Elasticity

If the equations are nonlinear the bilinear form (1.24) changes to the semilinear form $a(\boldsymbol{u}; \boldsymbol{v})$ (nonlinear in the first and linear in the second argument, separated by a semicolon) and the VF and FEM changes to:

$$\begin{aligned} \text{Find } \boldsymbol{u} - \boldsymbol{\psi} \in V &: a(\boldsymbol{u}; \boldsymbol{v}) = L(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in V, && (1.30) \\ \text{Find } \boldsymbol{u_h} - \boldsymbol{\psi} \in V_h &: a(\boldsymbol{u_h}; \boldsymbol{v}) = L(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in V_h. && (1.31) \end{aligned}$$

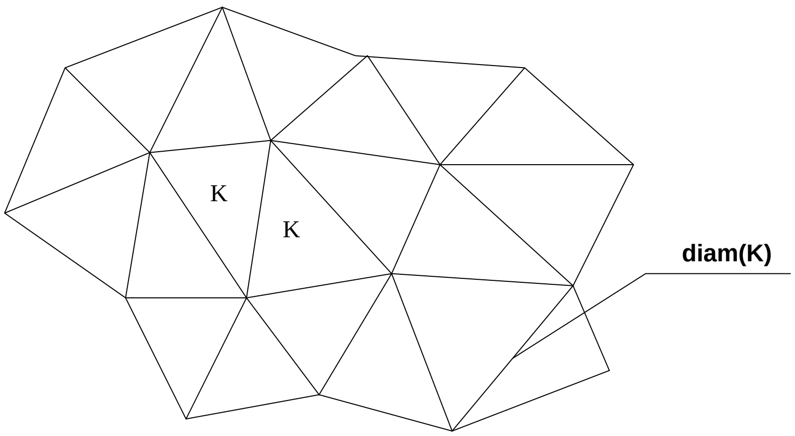

Figure 1.2: Illustration of diam(K), the largest side of element K.

The corresponding nonlinear Galerkin orthogonality is formulated using the secant form of $a(\boldsymbol{u};\boldsymbol{v})$ defined as

$$a_s(\boldsymbol{u},\boldsymbol{u_h};\boldsymbol{w},\boldsymbol{v}) = \int_0^1 a'(\bar{\boldsymbol{u}}(s);\boldsymbol{w},\boldsymbol{v})\ ds, \tag{1.32}$$

where $\bar{\boldsymbol{u}}(s) = \boldsymbol{u_h} + s\boldsymbol{e}$. In (1.32) we used the Gateaux derivative of $a(\boldsymbol{u};\boldsymbol{v})$ defined as

$$a'(\boldsymbol{u};\boldsymbol{w},\boldsymbol{v}) = \frac{\partial}{\partial\epsilon}a(\boldsymbol{u}+\epsilon\boldsymbol{w};\boldsymbol{v})_{|\epsilon=0}. \tag{1.33}$$

Choosing $\boldsymbol{w} := \boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u_h}$, the Fundamental Theorem of Calculus (FTC) leads to the result

$$a_s(\boldsymbol{u},\boldsymbol{u_h};\boldsymbol{e},\boldsymbol{v}) = \int_0^1 a'(\bar{\boldsymbol{u}}(s),\boldsymbol{e};\boldsymbol{v})\ ds = a(\boldsymbol{u};\boldsymbol{v}) - a(\boldsymbol{u_h};\boldsymbol{v}). \tag{1.34}$$

From the nonlinear VF (1.30) we can substitute the first term in the RHS to obtain the nonlinear Galerkin orthogonality

$$\begin{aligned} a_s(\boldsymbol{u},\boldsymbol{u_h};\boldsymbol{e},\boldsymbol{v}) &= L(\boldsymbol{v}) - a(\boldsymbol{u_h};\boldsymbol{v}) \qquad \forall\boldsymbol{v}\in V, & (1.35)\\ a_s(\boldsymbol{u},\boldsymbol{u_h};\boldsymbol{e},\boldsymbol{v}) &= 0 \qquad \forall\boldsymbol{v}\in V_h. & (1.36)\end{aligned}$$

These expressions will be used in the error analysis in the next chapter.

Chapter 2

The Dual Problem

Duality arguments and adaptive algorithms are widespread approaches when it comes to error control in finite elements. The global stability factor obtained from the dual solution have been used to bound the error in different norms [7]. In this thesis, the dual solution is computed and evaluated locally in every element and used as a weight factor on the element residual trying to compute the exact error from the discretization. This approach doesn't bound the error, nor is it in general able to compute the exact error due to the discretization. However it makes it possible to adapt a mesh according to a pre-specified goal i.e. *goal-oriented error adaptivity.*

This chapter explains how to specify the goal of the computations a priori and how this can be used in terms of a dual problem to obtain an mesh, that is adapted with respect to the quantity of interest. More on this type of a posteriori error estimate can be found in [14], [11], [4], [13] and [3].

2.1 The Idea Behind Duality Arguments

The error from the discretization is governed by

$$A\boldsymbol{e} = \boldsymbol{R}, \tag{2.1}$$

where A is a linear operator and $\boldsymbol{R}$ is the residual defined in (1.6). The effect of the residual $\boldsymbol{R}$ on the error $\boldsymbol{e}(\boldsymbol{y})$ at a point $\boldsymbol{y} \in \Omega$ is governed by the Green's function of the continuous problem. (Similar to influence functions in structural mechanics.) The Green's function is defined as

$$AG(\boldsymbol{x}, \boldsymbol{y}) = \delta(\boldsymbol{x} - \boldsymbol{y}). \tag{2.2}$$

Physically, the Green's function $G(\boldsymbol{x}, \boldsymbol{y})$ represents the effect at the point $\boldsymbol{x}$ of a Dirac delta function source at the point $\boldsymbol{x} = \boldsymbol{y}$. The solution to (2.1) can be written as

$$\boldsymbol{e}(\boldsymbol{x}) = \int_{\Omega} G(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{R}(\boldsymbol{y}) \, d\Omega. \tag{2.3}$$

A comprehensive treatment of the Green's function can be found in [6, Debnath, Mikusinski]

Green's function is suitable for pointwise error control of the solution. The analog to error control in more general quantities is the solution of a dual problem $\boldsymbol{\varphi}$. This can be thought of as a generalized influence function. We illustrate the idea of duality by considering the following problem

$$A\boldsymbol{u} = \boldsymbol{b}. \tag{2.4}$$

If we denote the exact solution $\boldsymbol{u}$ and the discretized solution $\boldsymbol{u_h}$ we can define the error due to the discretization by

$$\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u_h}. \tag{2.5}$$

We can now pose the continuous dual problem by

$$A^* \boldsymbol{\varphi} = \boldsymbol{e}, \tag{2.6}$$

where we consider $\boldsymbol{e}$ as the data. The adjoint operator A^* is defined as

$$(A\boldsymbol{w}, \boldsymbol{v}) = (\boldsymbol{w}, A^* \boldsymbol{v}). \tag{2.7}$$

Using this dual problem we derive the following error representation in L_2 norm

$$\begin{aligned} \parallel \boldsymbol{e} \parallel^2 &= (\boldsymbol{e}, A^* \boldsymbol{\varphi}) = (A\boldsymbol{e}, \boldsymbol{\varphi}) = (A\boldsymbol{u} - A\boldsymbol{u_h}, \boldsymbol{\varphi}) \\ &= (\boldsymbol{b} - A\boldsymbol{u_h}, \boldsymbol{\varphi}) = (\boldsymbol{R}, \boldsymbol{\varphi}). \end{aligned} \tag{2.8}$$

$\boldsymbol{R} = \boldsymbol{b} - A\boldsymbol{u_h}$ is the residual function obtained by exchanging $\boldsymbol{u}$ against $\boldsymbol{u_h}$ in the primal problem and $\boldsymbol{\varphi}$ is the solution to the dual problem. To give this a posteriori error estimate a quantitative meaning, the corresponding continuous dual problem has to be solved.

2.2 Goal Oriented Error Measures

Goal-oriented adaptivity can be described as how to adapt a mesh to compute a specific quantity as efficiently as possible, or how to obtain as accurate solution as possible with a pre-specified number of degrees of freedom in the problem. Let $Q(\boldsymbol{u})$ be a quantity of physical interest derived from the solution $\boldsymbol{u}$. To measure the error from the discretization we introduce the 'goal-quantity error measurer' $E(\boldsymbol{u}, \boldsymbol{u_h})$, defined as

$$E(\boldsymbol{u}, \boldsymbol{u_h}) \quad = \quad \lambda(\boldsymbol{e}). \tag{2.9}$$

$\lambda(\boldsymbol{e})$ is defined as one of two choices

$$\begin{aligned} \lambda(\boldsymbol{e}) &= Q(\boldsymbol{u}) - Q(\boldsymbol{u_h}), && (2.10) \\ \lambda(\boldsymbol{e}) &= Q(\boldsymbol{u} - \boldsymbol{u_h}) = Q(\boldsymbol{e}), && (2.11) \end{aligned}$$

depending on the quantity of interest. If Q is a linear operator they are equivalent.

Relevant quantities can be pointwise solutions (2.12), pointwise values of the gradient (2.13), contour integrals (2.14), global quantities as the L_2 norm of the error (2.15), or some other functional of interest.

$$\begin{aligned} Q(\boldsymbol{u}) &= \boldsymbol{u}(\boldsymbol{y}), && (2.12) \\ Q(\boldsymbol{u}) &= \nabla \boldsymbol{u}(\boldsymbol{y}), && (2.13) \\ Q(\boldsymbol{u}) &= \int_\Gamma \boldsymbol{u} \cdot \boldsymbol{n} \, d\Gamma, && (2.14) \\ Q(\boldsymbol{e}) &= \parallel \boldsymbol{e} \parallel = \left(\int_\Omega |\boldsymbol{e}|^2 \, d\Omega \right)^{1/2}. && (2.15) \end{aligned}$$

Following the theoretical concept presented by for instance [11, Larsson] we introduce the Gateaux derivative for $E(\boldsymbol{u}, \boldsymbol{u_h})$ with respect to its first argument

$$E'(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{w}) = \frac{\partial}{\partial \epsilon} E(\boldsymbol{u} + \epsilon \boldsymbol{w}, \boldsymbol{u_h}) \mid_{\epsilon=0} = \nabla Q(\boldsymbol{u}) \cdot \boldsymbol{w}. \tag{2.16}$$

The secant form of $E(\boldsymbol{u}, \boldsymbol{u_h})$ is defined as

$$\begin{aligned} E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{w}) &= \int_0^1 E'(\bar{\boldsymbol{u}}(s), \boldsymbol{u_h}; \boldsymbol{w}) ds \\ &= \int_0^1 \nabla Q(\bar{\boldsymbol{u}}(s)) \cdot \boldsymbol{w} \, ds \qquad \forall \boldsymbol{w} \in V, && (2.17) \end{aligned}$$

where $\bar{\boldsymbol{u}}(s) = \boldsymbol{u_h} + s\boldsymbol{e}$.

The fundamental theorem of calculus (FTC) leads to the result

$$E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{e}) = E(\boldsymbol{u}, \boldsymbol{u_h}) - E(\boldsymbol{u_h}, \boldsymbol{u_h}) = E(\boldsymbol{u}, \boldsymbol{u_h}). \tag{2.18}$$

For given $\boldsymbol{u}$, $\boldsymbol{u_h}$ we recall the secant form of the primal problem (1.32), and define the corresponding dual bilinear form $a_S^*(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{w}, \boldsymbol{v})$ such that

$$a_S^*(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{w}, \boldsymbol{v}) = a_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{v}, \boldsymbol{w}). \tag{2.19}$$

The abstract variational format of the dual problem is defined as

$$\text{Find } \boldsymbol{\varphi} \in V : \quad a_S^*(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{\varphi}, \boldsymbol{v}) = E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{v}) \qquad \forall \boldsymbol{v} \in V. \tag{2.20}$$

Note: Since we are dealing with linear elasticity we can use the linearity to drop the two leading terms in the LHS of (2.20), since the secant is equal to the tangent, and use the symmetry in the problem to obtain

$$a_S^*(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{w}, \boldsymbol{v}) = a^*(\boldsymbol{w}, \boldsymbol{v}) = a(\boldsymbol{v}, \boldsymbol{w}) = a(\boldsymbol{w}, \boldsymbol{v}). \tag{2.21}$$

Summing up, the following exact error representation holds where we have chosen $\boldsymbol{v} = \boldsymbol{e} := \boldsymbol{u} - \boldsymbol{u_h}$

$$\begin{aligned} E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{e}) &= a_S^*(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{\varphi}, \boldsymbol{e}) = a_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{e}, \boldsymbol{\varphi}) \\ &= a_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{e}, \boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}). \end{aligned} \tag{2.22}$$

The final equality is obtained by using the galerkin orthogonality (1.36) to insert an interpolant $\boldsymbol{\pi_h}\boldsymbol{\varphi}$ that belongs to V_h.

Using the FTC (2.18), (2.9) and (1.36) we obtain

$$\begin{aligned} \lambda(\boldsymbol{e}) &= L(\boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}) \\ &= \sum_{K \in T_h} (L(\boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}))_K. \end{aligned} \tag{2.23}$$

$$|\lambda(\boldsymbol{e})| = |\sum_{K \in T_h} (L(\boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi} - \boldsymbol{\pi_h}\boldsymbol{\varphi}))_K| \tag{2.24}$$

This is the exact error representation that can be discretized to formulate a goal-oriented adaptive routine for computing $Q(\boldsymbol{u})$.

Example

Consider a scalar PDE with the the chosen goal quantity

$$Q(u) = \int_\Omega u^2 \, d\Omega. \tag{2.25}$$

The 'goal-quantity error measurer' $E(u, u_h)$ is then defined as

$$\begin{aligned} E(u, u_h) &= Q(u) - Q(u_h) = \int_\Omega u^2 \, d\Omega - \int_\Omega u_h^2 \, d\Omega & (2.26) \\ &= \int_\Omega u^2 - u_h^2 \, d\Omega = \int_\Omega (u + u_h)\,(u - u_h)\; d\Omega. & (2.27) \end{aligned}$$

From the last equality we see that the data to the continuous dual problem reads $(u + u_h)$ and we will obtain the same result by proceeding as described above.

$$\begin{aligned} E_S(u, u_h; w) &= \int_0^1 \frac{\partial}{\partial \varepsilon}\Big(\int_\Omega (\bar{u} + \varepsilon w)^2 - {u_h}^2 \, d\Omega \Big)_{|\varepsilon=0} \, ds \\ &= \int_\Omega \int_0^1 2\,(u_h + se)\, w \; ds \;\; d\Omega \\ &= \int_\Omega 2 \left[u_h s + \frac{s^2}{2}\,(u - u_h) \right]_0^1 w \, d\Omega \\ &= \int_\Omega (u_h + u)\, w \, d\Omega \qquad \forall w \in V, & (2.28) \end{aligned}$$

with $w = e := u - u_h$.

2.3 Discretizing the Dual Problem

Solving (2.20) for all $\boldsymbol{v} \in V$ is the same type of problem as to compute the solution to the primal problem (1.26), and we must choose a subset of V to be able to compute an approximation of $\boldsymbol{\varphi}$. The Galerkin orthogonality (1.7) forces the approximation of the dual problem to be spanned by a larger function space than V_h. This refined function space will be called V_h^+. The following function spaces are now considered

$$V_h \subset V_h^+ \subset V \equiv H, \tag{2.29}$$

where H is a Sobolev space specified by the differential operator. (2.29) is a theoretical restriction to the choice of V so that the involved functionals make sense.

A straightforward way to choose V_h^+ is to use one degree higher polynomial than was used when discretizing the primal problem or a refined triangulation with the same order. Other methods proposed is solving the dual problem in V_h and then do some post processing to obtain a $\boldsymbol{\varphi}$ that belongs to an improved space. See for instance [11, Larsson] for developments in this direction.

The evaluation of the dual problem (2.20) first requires a linearization of the functional $E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{v})$ i.e. the data to the problem. Linear goal quantities result in functionals that does not involve the exact solution, and there is no need for a linearization. Nonlinear error measures can be handled by using the *rectangle* (2.30), *trapezoidal* (2.31) or *midpoint* (2.32) rule.

$$\int_0^1 f(\boldsymbol{u_h} + s\boldsymbol{e})\, ds \approx f(\boldsymbol{u_h}), \tag{2.30}$$

$$\int_0^1 f(\boldsymbol{u_h} + s\boldsymbol{e})\, ds \approx \frac{f(\boldsymbol{u_h}) + f(\boldsymbol{u_h}^+)}{2}, \tag{2.31}$$

$$\int_0^1 f(\boldsymbol{u_h} + s\boldsymbol{e})\, ds \approx f\left(\frac{\boldsymbol{u_h}}{2} + \frac{\boldsymbol{u_h}^+}{2}\right). \tag{2.32}$$

$\boldsymbol{u_h}^+$ is an improved solution closer to $\boldsymbol{u}$ computed in V_h^+. Equation (2.30) is equivalent to using the tangent form instead of the secant form. This is a well known approach in the nonlinear primal problem where the stiffness matrix is updated in every iteration.

Carrying out the discretization we finally obtain the discretized dual problem that reads

$$\text{Find } \boldsymbol{\varphi} \in V_h^+ : \quad a_S^*(\boldsymbol{u_h}^+, \boldsymbol{u_h}, \boldsymbol{\varphi}, \boldsymbol{v}) = E_S(\boldsymbol{u_h}^+, \boldsymbol{u_h}; \boldsymbol{v}) \qquad \forall v \in V_h^+ \tag{2.33}$$

When discretizing the dual problem the exact error representation changes to

$$|\tilde{\lambda}(\boldsymbol{e})| = |\sum_{K \in T_h} (L(\boldsymbol{\varphi_h}^+ - \boldsymbol{\pi_h \varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi_h}^+ - \boldsymbol{\pi_h \varphi}))_K|. \tag{2.34}$$

Chapter 3

Adaptivity

The adaptive algorithm can schematically be described as in Fig.(3.1).

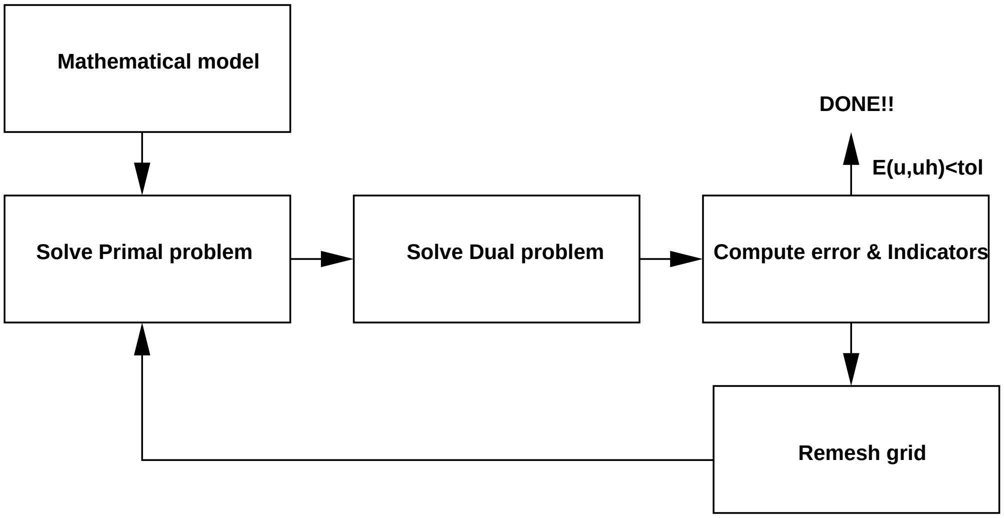

Figure 3.1: Graphical illustration of the adaptive process. Starting from a mathematical model the Primal and Dual problems are solved. Indicators and error are computed and finally, if the tolerance is not fulfilled, the mesh is refined.

The adaptive algorithm seeks to realize the criterion

$$|\lambda(\boldsymbol{e})| \leq \mathrm{tol} \tag{3.1}$$

where tol is a user specified tolerance of the error. The first step is to compute the primal solution $\boldsymbol{u_h} \in V_h$. The next step is to compute the dual solution $\boldsymbol{\varphi_{h^+}} \in V_h^+$. The solutions to the primal and dual problem are used to control whether the stop criterion is fulfilled or not. If not, we use element indicators to refine the mesh according to some strategy (will be explained later), and start over from the beginning.

3.1 Stop Criterion and Indicators

Starting from (2.24) the following estimates of the error can be used in the Adaptivity algorithm

$$\begin{aligned} |\lambda(\boldsymbol{e})| &= |\sum_{K \in T_h} (L(\boldsymbol{\varphi} - \boldsymbol{\pi_h \varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi} - \boldsymbol{\pi_h \varphi}))_K|, & (3.2) \\ &\leq \sum_{K \in T_h} |(L(\boldsymbol{\varphi} - \boldsymbol{\pi_h \varphi}) - a(\boldsymbol{u_h}, \boldsymbol{\varphi} - \boldsymbol{\pi_h \varphi}))|_K, & (3.3) \\ &\leq \sum_{K \in T_h} R_T(\boldsymbol{u_h}) \cdot W_T(\boldsymbol{\varphi} - \boldsymbol{\pi_h \varphi}). & (3.4) \end{aligned}$$

Equation (3.2) is an identity. In (3.3) the triangle inequality omits cancellation between different elements. In (3.4) integration by parts, the Cauchy-Schwarz inequality, and an estimation on the element level have been used to evaluate two bounding constants in every element. Example of this last inequality can be found in [7], and in [4, Becker, Rannacher].

Earlier works [10, Larson, Niklasson] propose that we use (3.2) as a stopping criteria and (3.3) to assign refinement indicators to the elements in T_h. This is a natural choice since elements with large contributions to (3.2) must be controlled. The weights used in (3.4) can be used to compute indicators and error, but the obtained error is usually too big [10].

3.2 Refinement Strategies

There are different strategies on how to use the set of indicators for choosing elements to refine. We list some strategies here together with some comments. We define η_K as the error contribution from element K and the corresponding indicator as $I_K = |\eta_K|$.

1. *Fixed fraction.* In each refinement cycle the elements are sorted according to their indicator's magnitude I_K and a fixed fraction are chosen to be refined. This strategy gives good control of the size of the refined mesh, but is insensitive to whether the problem has a singularity or not. Over refinements when singularities are present is common in practice.

2. *Relative fraction.* In each refinement cycle the elements are sorted according to their indicator's magnitude I_K. Every element whose indicator value is greater than, or equal to a fraction of the largest value are chosen to be refined. This strategy is more adaptive than strategy (1). In practice, when a singularity is present, the elements surrounding the singularity are chosen to be refined. This might lead to many refinement cycles with just a few more elements in every cycle.

3. *Equidistribution of errors.* The adaptive process tries to equilibrate the error contributions from every element in the mesh according to $\eta_K = \frac{tol}{N_f}$, where N_f is the number of elements in the final mesh. Since N_f is unknown it is common in practice to choose $N_f = N_c$, where N_c is the number of elements in the current mesh. This means that elements are allowed to transform from coarser to finer and back again in time dependent problems (hard to implement). If the tolerance is set too low (problem dependent) this strategy might lead to uniform refinements until the memory is exhausted. I.e. may not lead to $\lambda(\boldsymbol{e}) \leq$ tol.
   Note: In practice, strategy (1) and (2) also tries to do an equidistribution.

4. *Fixed degrees of freedom.* Order the cells according to their indicators magnitude I_K and refine a fraction of the cells with largest contributions to the error. Coarse the same number of elements whose indicator magnitude is smallest. This makes the degrees of freedom in the problem and the amount of CPU time needed to solve the problem in every iteration more or less constant.

5. *Optimized mesh.* A proposed error representation including a smooth mesh-size function $h(x)$ and an h-independent error density function $\Phi(x)$ is used for generating a formula for an optimal mesh-size distribution. This is achieved a priori under very restrictive conditions but is in general supported by computational experience. See [4] for details.

6. *Mixed refinement* This strategy is implemented in Diffpack [9]. Using this strategy a refinement flag between 0 and 3 is assigned to every element in the mesh. The flag indicates how many sides that is to be refined in the element. If an element is chosen to be refined, the longest side is always refined (the largest element angel is divided). This strategy leads to approximately $15 - 20\%$ more elements in every refinement cycle.

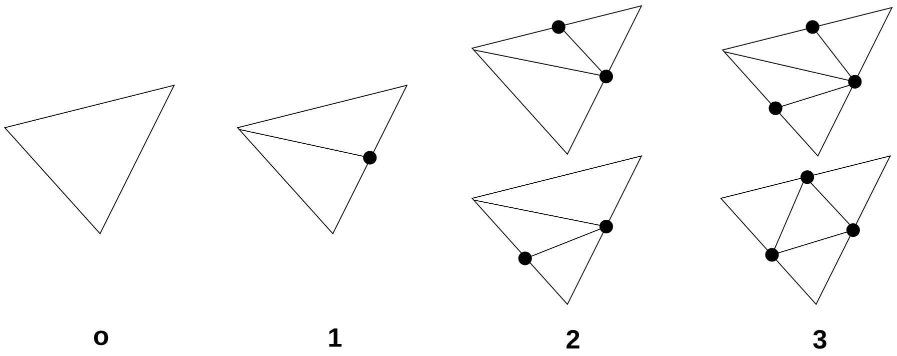

Figure 3.2: The flag in the interval 0-3 indicates how many sides that is to be refined. Flag 3 can be refined uniformly or by bisection of the largest angel.

One way to decide the flag on the element is by using the following table where $I_K^* = \max_{K \in T_h} I_K$ i.e. the largest element indicator in the mesh.

Table 3.1: One strategy to put flags on the element where $I_K^* = \max_{K \in T_h} I_K$.

Element Indicator $\mathbf{I_K}$	**Flag**
$I_K^* \geq I_K \geq 0.5 \cdot I_K^*$	3
$0.5 \cdot I_K^* > I_K \geq 0.5^2 \cdot I_K^*$	2
$0.5^2 \cdot I_K^* > I_K \geq 0.5^3 \cdot I_K^*$	1
$0.5^3 \cdot I_K^* > I_K$	0

3.3 Putting it all Together

We demonstrate the material presented so far by considering a console in plane strain with a distributed load, see Fig.(3.3).

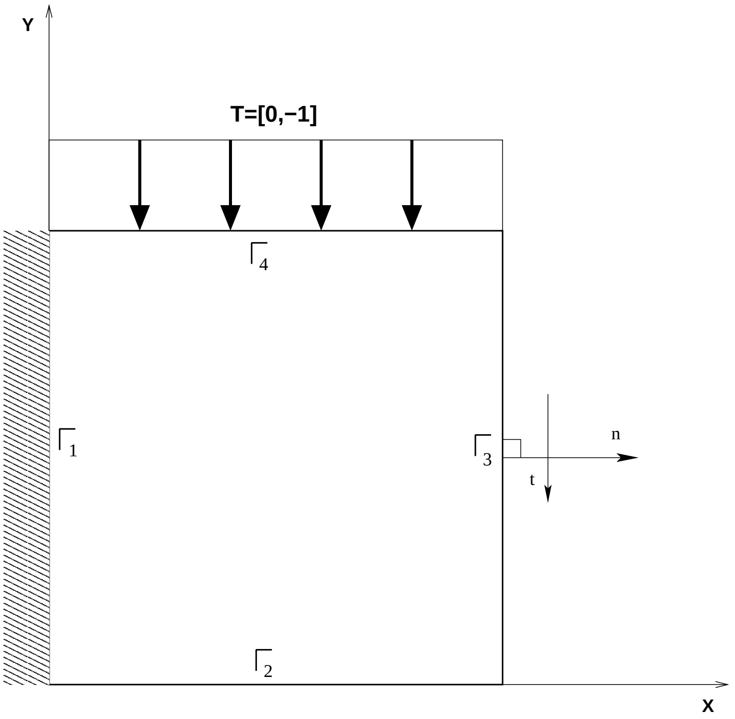

Figure 3.3: A console in plane strain. The left boundary have prescribed displacements $u_i = 0$. $\boldsymbol{n}$ and $\boldsymbol{t}$ are normal- and tangent-vectors for the right boundary Γ_3. $\boldsymbol{T} = [0, -1]$ is the load on top of the console.

The equations of linear elasticity together with boundary conditions define the primal problem

$$\boldsymbol{\sigma} = \lambda(\boldsymbol{\nabla} \cdot \boldsymbol{u})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \quad \text{in } \Omega, \tag{3.5}$$

$$-\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \boldsymbol{0} \quad \text{in } \Omega, \tag{3.6}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_1, \tag{3.7}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{T} \quad \text{on } \Gamma_4. \tag{3.8}$$

The Galerkin method leads to a weak form that corresponds to

$$\int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega + \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v}\, ds. \tag{3.9}$$

If we define the bilinear form $a(\boldsymbol{u}, \boldsymbol{v})$ and linear form $L(\boldsymbol{v})$ as

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega, \tag{3.10}$$

$$L(\boldsymbol{v}) = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega + \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v}\, ds, \tag{3.11}$$

the corresponding VF and FEM reads

$$\text{Find } \boldsymbol{u} - \boldsymbol{\psi} \in V \;:\; a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}) \;\; \forall \boldsymbol{v} \in V, \tag{3.12}$$

$$\text{Find } \boldsymbol{u_h} - \boldsymbol{\psi} \in V_h \;:\; a(\boldsymbol{u_h}, \boldsymbol{v}) = L(\boldsymbol{v}) \;\; \forall \boldsymbol{v} \in V_h. \tag{3.13}$$

To define the dual problem we choose the goal-quantity as the mean vertical displacements over the right boundary Γ_3

$$Q(\boldsymbol{u}) = \frac{1}{m(\Gamma_3)} \int_{\Gamma_3} \boldsymbol{t} \cdot \boldsymbol{u}\, ds, \tag{3.14}$$

where $m(\Gamma_3)$ is the length of boundary Γ_3. This leads to the following discretized dual problem

$$\text{Find } \boldsymbol{\varphi_h}^+ \in V_h{}^+ : \quad a(\boldsymbol{v}, \boldsymbol{\varphi_h}^+) = E_S(\boldsymbol{u_h}^+, \boldsymbol{u_h}; \boldsymbol{v}) \qquad \forall \boldsymbol{v} \in V_h{}^+ \tag{3.15}$$

where

$$E_S(\boldsymbol{u_h}^+, \boldsymbol{u_h}; \boldsymbol{v}) = \frac{1}{m(\Gamma_3)} \int_{\Gamma_3} \boldsymbol{t} \cdot \boldsymbol{v}\, ds. \tag{3.16}$$

Figure(3.4) shows the adapted mesh obtained with the tolerance set to 1%. The convergence rates for uniform and adaptive refinements are plotted in Fig.(3.5). The effectivity index in Fig.(3.6) is defined as

$$\eta = \frac{Q(\boldsymbol{u_h}^+) - Q(\boldsymbol{u_h})}{Q(\boldsymbol{u}) - Q(\boldsymbol{u_h})}, \tag{3.17}$$

where $Q(\boldsymbol{u})$ is approximated with a reference solution obtained from a converged adaptive computation.

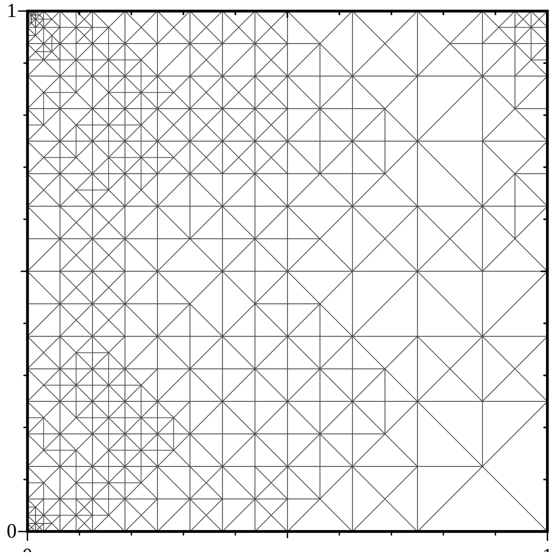

Figure 3.4: Adapted mesh, obtained with the tolerance for the relative error $tol = 1\%$. Note the local refinements in the corners.

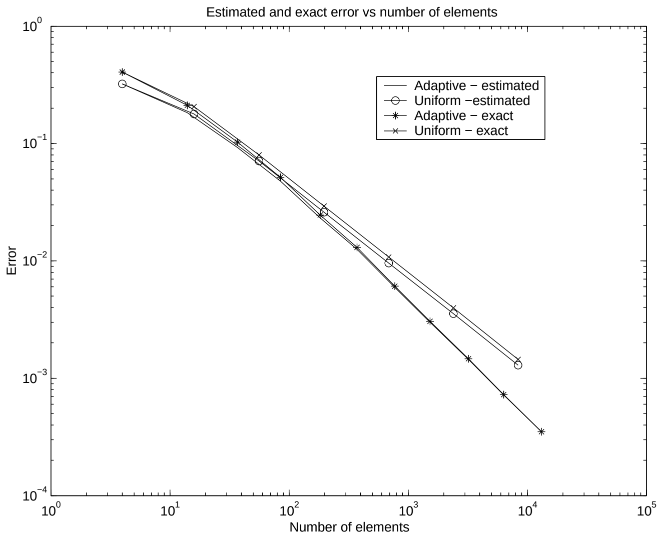

Figure 3.5: Convergence rates for uniform and adaptive refinements.

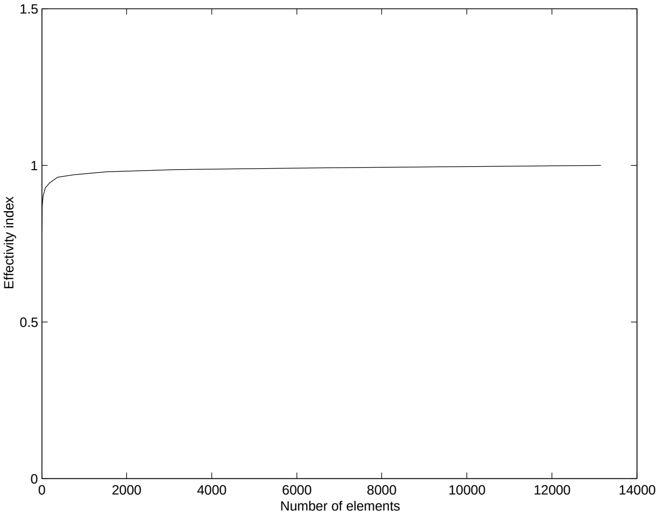

Figure 3.6: Effectivity index defined as $\eta = \frac{Q(\boldsymbol{u_h}^+)-Q(\boldsymbol{u_h})}{Q(\boldsymbol{u})-Q(\boldsymbol{u_h})}$.

In Fig.(3.6) we notice that the effectivity index is close to unity after a few iterations. I.e. the error representation is equal to the difference between two primal solutions. This behaviour is typical for linear goal quantities.

Chapter 4

Nonlinear Elastic Fracture Mechanics

There are many developed theories which account for various types of nonlinear material behavior in fracture mechanics. Most of them are extensions to Linear Elastic Fracture Mechanics (LEFM). A solid background in LEFM is thus essential to understand more advanced concepts. In this chapter some fundamentals about the *J-contour integral* and the *energy domain integral* are described and then the focus is on the computations. (Both methods are equivalent in nonlinear elastic materials and can be used for computing the energy release rate J.) We refer to [1, Anderson] for a comprehensive treatment of the physical and theoretical background of the considered equations.

In Appendix, the derivation of the J-contour integral as an energy release rate, its path independence and the derivation of the energy domain integral can be found.

4.1 The J-Contour Integral

Consider an arbitrary counter-clockwise path Γ around a crack-tip as in Fig.(4.1). (The normal vector is pointing out from the crack-tip.) The J-contour integral is defined as

$$J = \int_{\Gamma} \left(W \, dx_2 - T_i \frac{\partial u_i}{\partial x_1} \, ds \right) \tag{4.1}$$

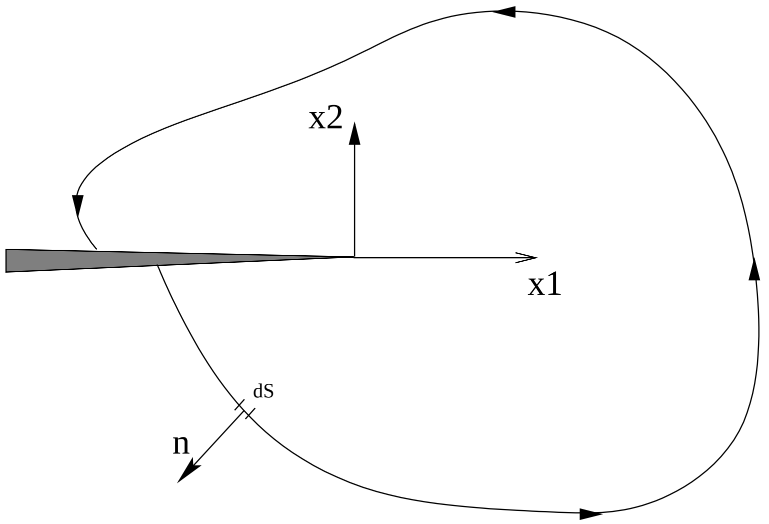

Figure 4.1: The J-contour integral. x_1, x_2 are the local coordinate axis attached to the crack tip. The normal vector $\boldsymbol{n}$ is pointing out from the crack tip.

where W is the strain energy density, T_i are the components of the traction vector and u_i is the displacement vector.

$$W = \int_0^{\varepsilon_{ij}} \sigma_{ij} d\varepsilon_{ij} \tag{4.2}$$

$$T_i = \sigma_{ij} n_j \tag{4.3}$$

This J-contour integral has been widely used as a fracture characterizing parameter for nonlinear elastic materials.

In 1968 [15] Rice showed that the value of this integral, J (the energy release rate in a nonlinear elastic body at crack growth), could be written as a path independent line integral (see Appendix). The Elastic-plastic material follows a linear unloading with a slope equal to Young's modulus, while the nonlinear elastic material unloads along the same path as it was loaded, illustrated in Fig.(4.2). Thus an analysis that assumes nonlinear elastic material may be valid for an elastic-plastic material, provided no unloading occurs.

The path independence (proven in the Appendix), in elastic materials, makes it possible to evaluate the J-contour integral at a remote contour avoiding singularity zones close to the crack tip.

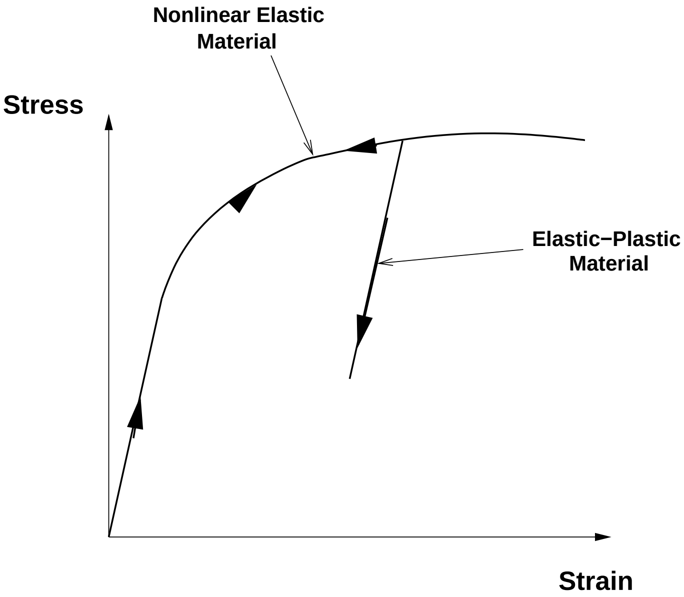

Figure 4.2: The stress-strain relationship for a nonlinear elastic material and an elastic plastic material. The materials behave equal during loading but when unloading occurs they follow different paths.

4.2 The Energy Domain Integral

More recent formulations of the energy release rate apply an area integration for 2D problems and a volume integral for 3D problems. The more recently formulated *energy domain integral* methodology [1] is a method for numerical analysis of the energy release rate. It can be applied to quasi static and dynamic problems with elastic, plastic or viscoplastic material response as well as thermal loading.

The energy domain integral is derived from a generalized energy balance (see Appendix for the derivation) and reads

$$J \;=\; \int_{A^*} \left(\left[\sigma_{ij} \frac{\partial u_j}{\partial x_1} - W\delta_{1i} \right] \frac{\partial q}{\partial x_i} \right) \, dA \qquad (4.4)$$

Assuming $W = W^e$ makes it equivalent to Rice's path-independent J contour integral. The function q is explained in the Appendix.

Chapter 5

Finite Element Implementation

We solve the equations of linear elasticity with a prescribed boundary traction in plane stress for a body with a central crack, see Fig.(5.1). The obtained displacement field is used to compute the energy release rate at crack growth with the J-contour and the energy domain integral.

The stress intensity factor K_I is evaluated using an asymptotic displacement approach at the crack tip (explained later). K_I is evaluated as an insurance that the obtained energy release rates are correct. The stress intensity factor, for cracks in Modus I, see Fig.(5.2), is related to the energy release rate through (5.1).

$$J = \frac{K_I^2}{E}, \tag{5.1}$$

See [5], [8] and [18] for the relation between the energy release rate and the stress intencity factors under different modus loads.

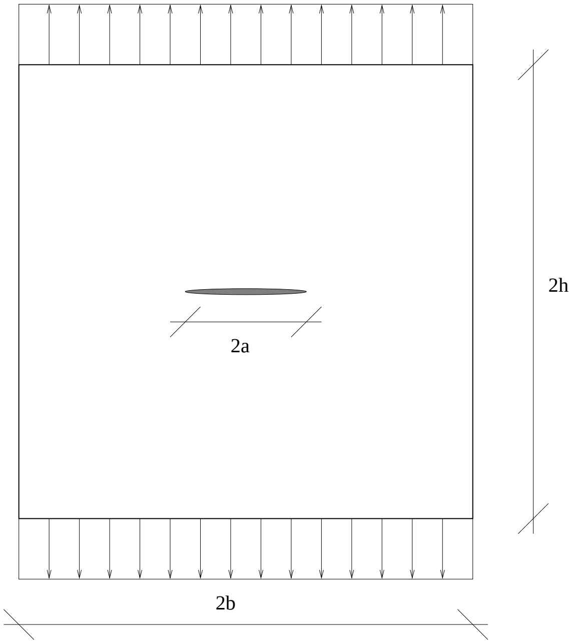

Figure 5.1: Central crack loaded in modus I. $2a$ is the crack length, $2h$ is the height of the body and $2b$ is the width.

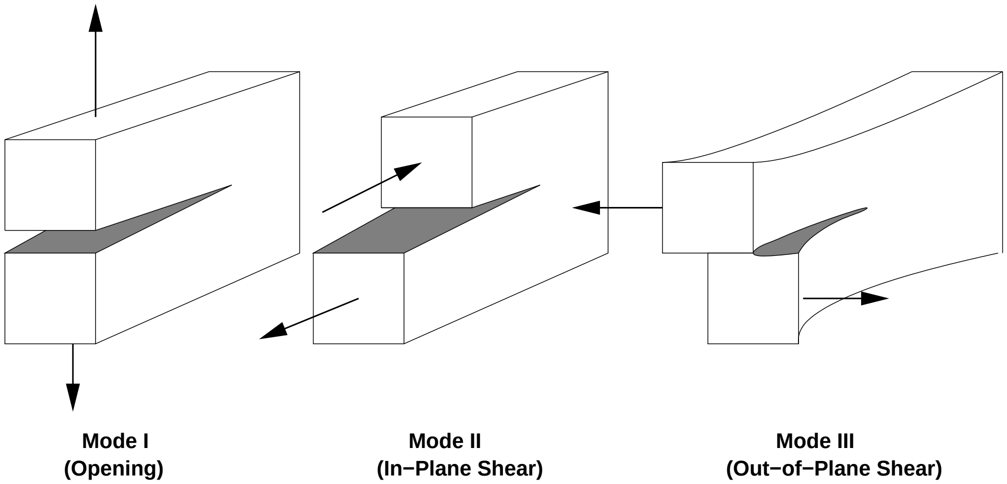

Figure 5.2: Definition of modus I, modus II and modus III loading that can be applied to a crack.

5.1 The Primal Problem

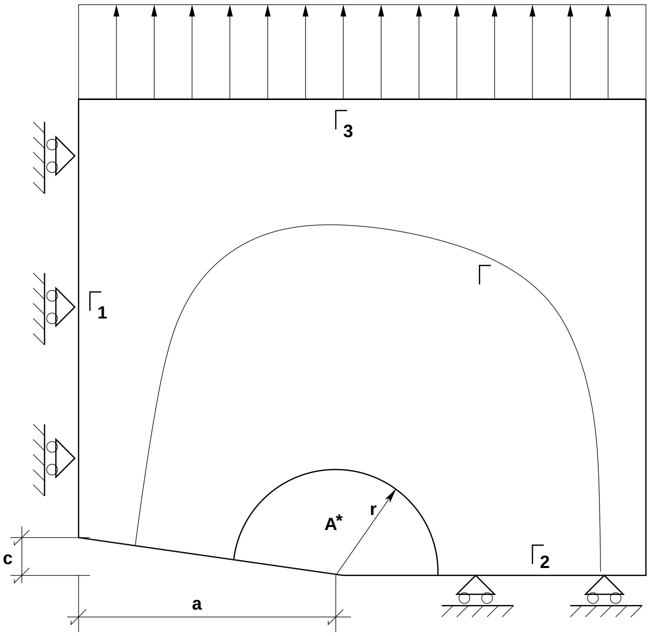

Figure 5.3: The primal problem. Γ represents the contour where the J-contour integral is evaluated. A^*, defined by $\boldsymbol{r}$, represents the area where the energy domain integral is evaluated. Half the length and width of the crack are denoted by a and c respectively.

Using the symmetry to reduce the problem to one quadrant Fig.(5.3), we solve

$$\boldsymbol{\sigma} = \lambda(\boldsymbol{\nabla} \cdot \boldsymbol{u})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \quad \text{in } \Omega, \tag{5.2}$$

$$-\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = 0 \quad \text{in } \Omega, \tag{5.3}$$

$$u_1 = 0 \quad \text{on } \Gamma_1, \tag{5.4}$$

$$u_2 = 0 \quad \text{on } \Gamma_2, \tag{5.5}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{\sigma_0} \quad \text{on } \Gamma_3, \tag{5.6}$$

with the finite element method using linear ($\boldsymbol{u_h}$) and quadratic ($\boldsymbol{u_h}^+$) basis functions. The singularity was obtained through the boundary conditions and the crack width c was actually set to zero. The two solution fields were used as data to the dual problem.

5.2 The Dual Problem

We use two different dual problems because we will evaluate both the J-contour and the energy domain integral for computing the energy release rate at crack growth. Using the linearity and symmetry in the problem the dual problem is defined as

$$\text{Find } \boldsymbol{\varphi} \in V : \quad a(\boldsymbol{\varphi}, \boldsymbol{v}) = E_S(\boldsymbol{u}, \boldsymbol{u_h}; \boldsymbol{v}) \qquad \forall \boldsymbol{v} \in V. \tag{5.7}$$

The RHS will change depending on the choice of goal function. The secant form is evaluated with the midpoint rule (2.32) using the two primal solutions $\boldsymbol{u_h}$ and $\boldsymbol{u_h}^+$. This linearization will be exact when $\boldsymbol{u_h}^+ \to \boldsymbol{u}$.

5.2.1 The Dual Problem for the J-Contour Integral

Choosing the goal quantity as the J-contour integral

$$Q(\boldsymbol{u}) = \int_\Gamma \left(W \, dx_2 - T_i \frac{\partial u_i}{\partial x_1} \, ds \right), \tag{5.8}$$

we have

$$E_S(\boldsymbol{u}, \boldsymbol{u_h}, \boldsymbol{v}) = \int_\Omega \int_0^1 \nabla Q(\bar{\boldsymbol{u}}) \cdot \boldsymbol{v} \, dS \, d\Omega, \tag{5.9}$$

where

$$\begin{aligned} \nabla Q_1 = & \\ & \left(\frac{\partial \bar{u}_2}{\partial x_2} n_1 \lambda - n_2 \left(\frac{\partial \bar{u}_1}{\partial x_2} \mu + \frac{\partial \bar{u}_2}{\partial x_1} (\lambda + \mu) \right) \right) \frac{\partial(\cdot)}{\partial x_1} + \\ & \left(\left(2 \frac{\partial \bar{u}_1}{\partial x_2} n_1 + \frac{\partial \bar{u}_2}{\partial x_1} n_1 - \frac{\partial \bar{u}_1}{\partial x_1} n_2 \right) \right) \mu \frac{\partial(\cdot)}{\partial x_2} \end{aligned} \tag{5.10}$$

$$\begin{aligned} \nabla Q_2 = & \\ & \left(\frac{\partial \bar{u}_1}{\partial x_2} n_1 \mu - \frac{\partial \bar{u}_1}{\partial x_1} n_2 (\lambda + \mu) - \frac{\partial \bar{u}_2}{\partial x_2} n_2 (\lambda + 2\mu) \right) \frac{\partial(\cdot)}{\partial x_1} + \\ & \left(\frac{\partial \bar{u}_1}{\partial x_1} n_1 \lambda + \left(2 \frac{\partial \bar{u}_2}{\partial x_2} n_1 - \frac{\partial \bar{u}_2}{\partial x_1} n_2 \right) (\lambda + 2\mu) \right) \frac{\partial(\cdot)}{\partial x_2} \end{aligned} \tag{5.11}$$

5.2.2 The Dual Problem for the Energy Domain Integral

Choosing the goal quantity as the energy domain integral

$$Q(\boldsymbol{u}) = \int_{A^*} \left(\left[\sigma_{ij} \frac{\partial u_j}{\partial x_1} - W\delta_{1i} \right] \frac{\partial q}{\partial x_i} \right) \, dA, \tag{5.12}$$

we have

$$E_S(\boldsymbol{u}, \boldsymbol{u_h}, \boldsymbol{v}) = \int_\Omega \int_0^1 \nabla Q(\bar{\boldsymbol{u}}) \cdot \boldsymbol{v} \; dS \, d\Omega, \tag{5.13}$$

where

$$\begin{aligned} \nabla Q_1 = & \\ & \left(\frac{\partial q}{\partial x_2} \left(\mu \frac{\partial \bar{u}_1}{\partial x_2} + (\mu + \lambda) \frac{\partial \bar{u}_2}{\partial x_1} \right) - \lambda \frac{\partial \bar{u}_2}{\partial x_2} \frac{\partial q}{\partial x_1} \right) \frac{\partial (\cdot)}{\partial x_1} + \\ & + \left(\mu \frac{\partial \bar{u}_1}{\partial x_1} \frac{\partial q}{\partial x_2} - \mu \frac{\partial q}{\partial x_1} \left(\frac{\partial \bar{u}_2}{\partial x_1} + 2 \frac{\partial \bar{u}_1}{\partial x_2} \right) \right) \frac{\partial (\cdot)}{\partial x_2} \end{aligned} \tag{5.14}$$

$$\begin{aligned} \nabla Q_2 = & \\ & \left(\frac{\partial q}{\partial x_2} \left(\frac{\partial \bar{u}_1}{\partial x_1} (\lambda + \mu) + \frac{\partial \bar{u}_2}{\partial x_2} (\lambda + 2\mu) \right) - \frac{\partial q}{\partial x_1} \frac{\partial \bar{u}_1}{\partial x_2} \mu \right) \frac{\partial (\cdot)}{\partial x_1} + \\ & + \left(\frac{\partial q}{\partial x_2} \frac{\partial \bar{u}_2}{\partial x_1} (\lambda + 2\mu) - \frac{\partial q}{\partial x_1} \left(\frac{\partial \bar{u}_1}{\partial x_1} \lambda + 2 \frac{\partial \bar{u}_2}{\partial x_2} (\lambda + 2\mu) \right) \right) \frac{\partial (\cdot)}{\partial x_2} \end{aligned} \tag{5.15}$$

The function $\boldsymbol{q}$ was defined as a tent function that equals unity on the crack tip and zero at the outer radie i.e. a cone.

5.3 Diffpack

Diffpack is a class library in C++ containing building blocks for solving PDE:s in computational science. The use of Diffpack makes it possible to have a higher degree of abstraction in the programming code. The goal-oriented adaptivity was not supported (before this thesis) by Diffpack so it had to be implemented. The assembly process, iterative solvers, refinement routines and visualization were carried out with Diffpack objects. The PDE:s obtained from the primal and dual problem are implemented in a variational style close to the mathematical formulations. To learn C++ and Diffpack, we refer to [17], [2] and [9].

5.4 Computational Results

Both methods for computing the energy release rate at crack growth were evaluated. In both methods the initial grid was as shown in Fig.(5.4). The grid was generated by Triangle (freeware). Triangle is a mesh generator in 2D. We refer to [16] for the program.

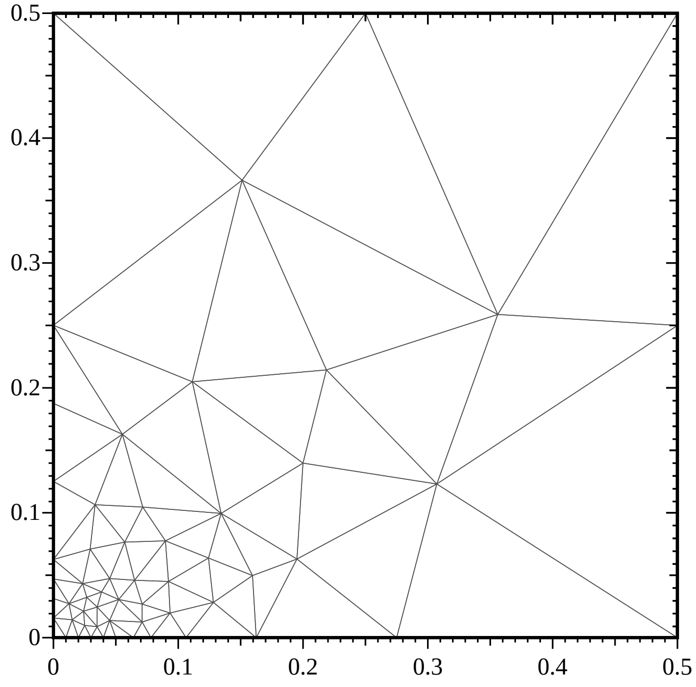

Figure 5.4: Start grid for the computations.

In the plots we use the effectivity index η and the average size of the elements $\bar{h}$ defined as

$$\eta = \frac{Q(\boldsymbol{u_h}^+) - Q(\boldsymbol{u_h})}{Q(\boldsymbol{u}) - Q(\boldsymbol{u_h})}, \qquad \bar{h} = \frac{1}{\sqrt{N}}. \tag{5.16}$$

$Q(\boldsymbol{u})$ was obtained from a reference solution, i.e. a converged computation.

5.4.1 J-Contour Integral

The J-contour integral was evaluated over four different contours to investigate its path independence and how different paths affected the adapted grid. The adapted grids are found in Fig.(5.5). Figure(5.6) to Fig.(5.8) are from the computations on path 4. Table(5.1) shows how the converged solution changed for different paths.

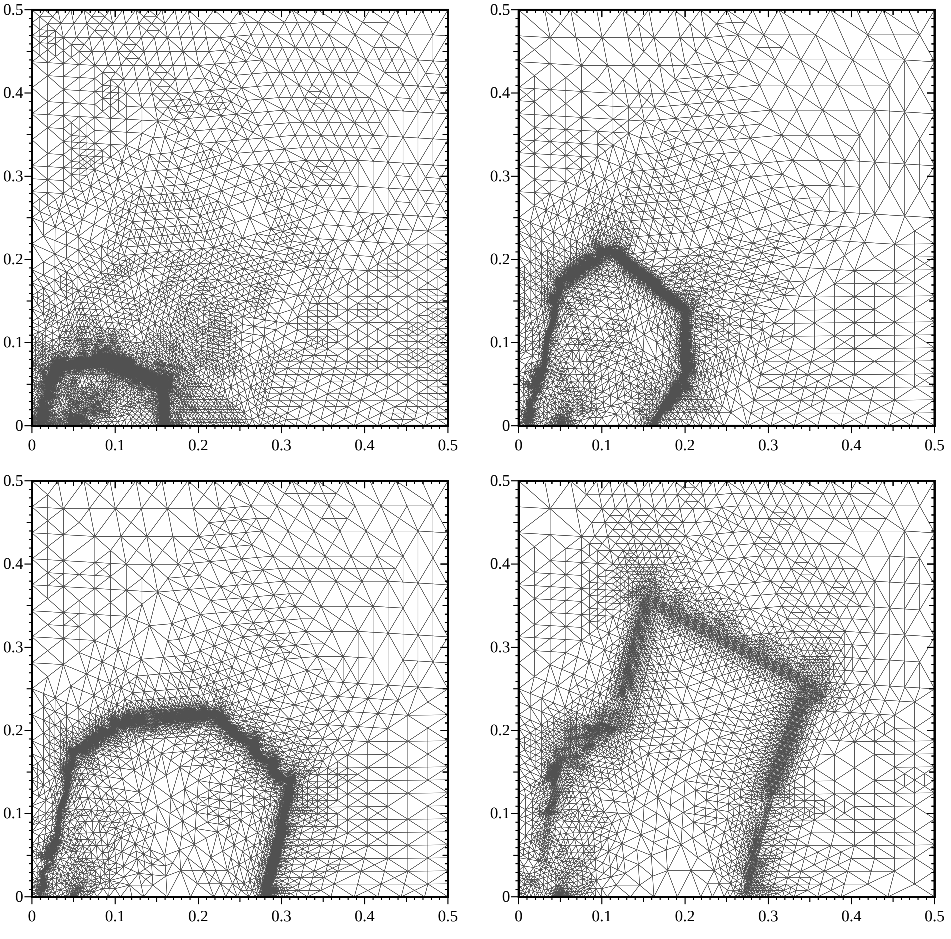

Figure 5.5: The adapted grids for the J-contour integral evaluated on path 1-4.

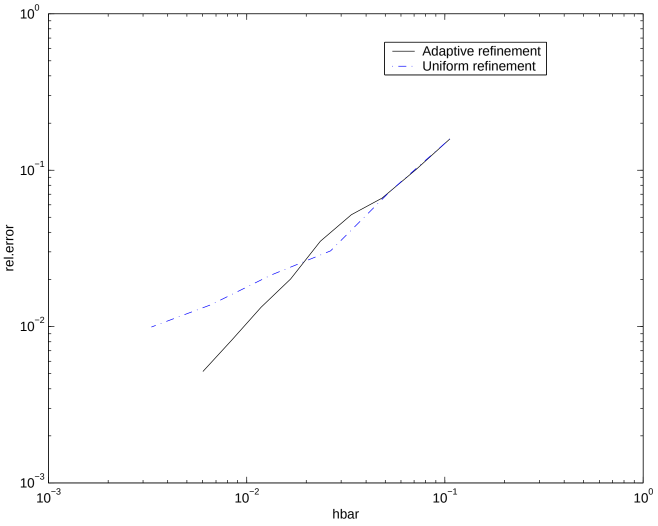

Figure 5.6: The convergence for the relative error using adaptive and uniform refinements.

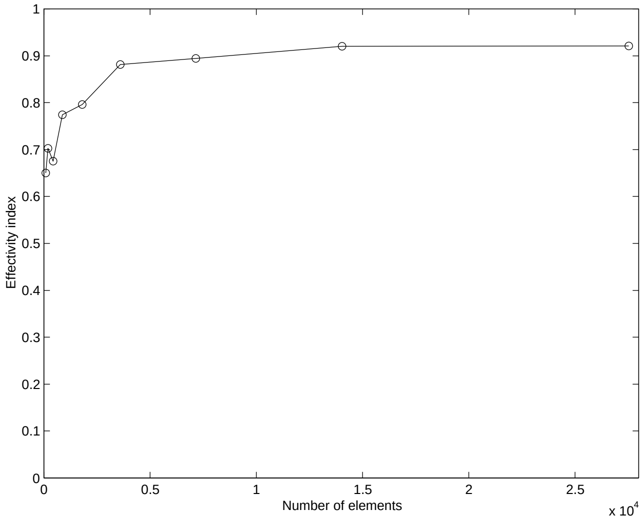

Figure 5.7: The effectivity index for the J-contour integral.

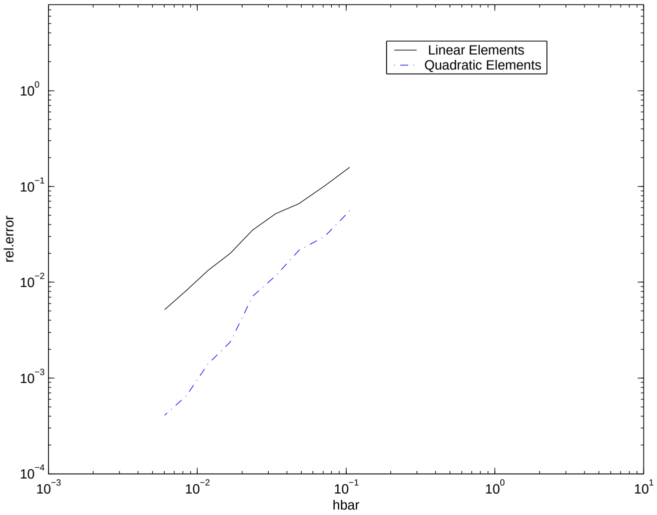

Figure 5.8: The convergence for linear and quadratic elements.

Table 5.1: The converged solution evaluated on path 1, 2, 3 and 4.

Path	$\mathbf{J}[\frac{\partial\pi}{\partial\mathbf{a}}]$	**Relative Error**
1	1008.09	0.0035
2	1134.63	0.0058
3	1124.28	0.0083
4	1131.42	0.0105

5.4.2 Energy Domain Integral

The energy domain integral was evaluated in five different areas to investigate its area independence and how the adapted grid was affected. Two of the adapted grids are found in Fig.(5.9). Figures(5.10-5.12) are from the computations with $r = 0.050$. Figure(5.2) shows how the converged solution changed for different values of the radie.

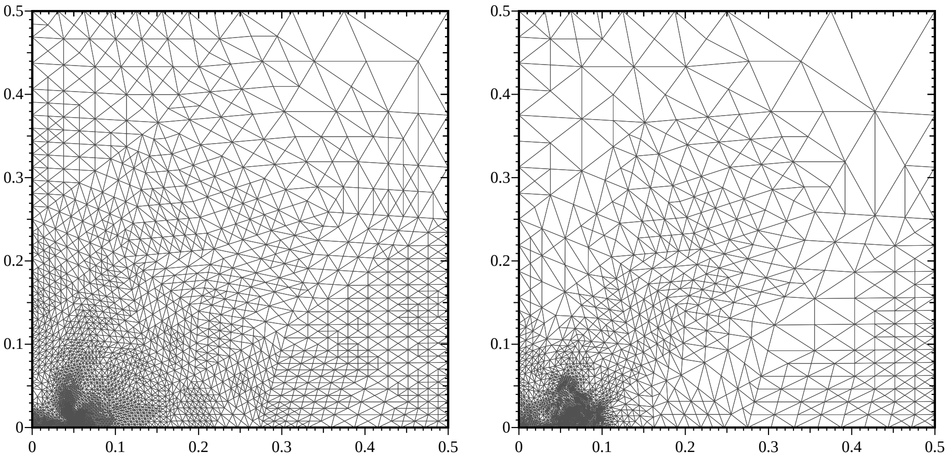

Figure 5.9: Adapted grids for the energy domain integral with $r = 0.050$ (to the left) and $r = 0.010$ (to the right).

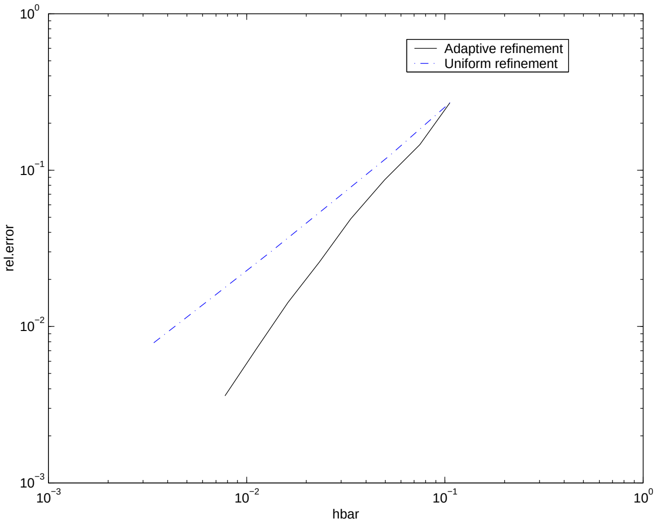

Figure 5.10: Convergence for the relative error using adaptive and uniform refinements.

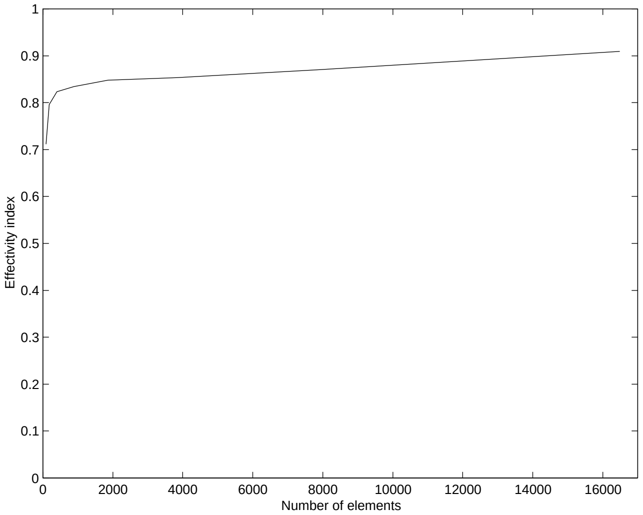

Figure 5.11: Effectivity index for the energy domain integral.

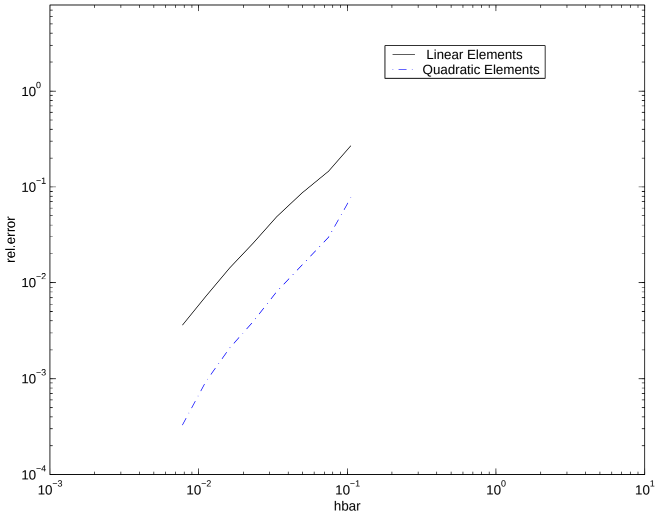

Figure 5.12: Convergence for linear and quadratic elements

Table 5.2: Computed energy release rate for different areas.

Radie[m]	$\mathbf{J}[\frac{\partial \pi}{\partial \mathbf{a}}]$	**Relative Error**
0.050	906.58	0.0015
0.040	890.434	0.0012
0.030	876.177	0.0013
0.020	863.508	0.0015
0.010	852.018	0.0020

5.4.3 Asymptotic Displacements

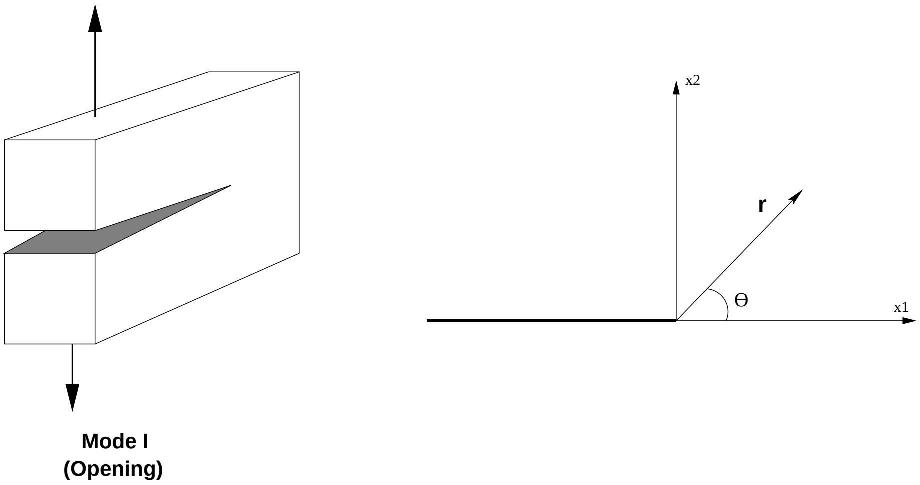

Figure 5.13: Crack in Modus I (left). Polar coordinate system attached to the crack tip (right).

The displacements of the crack face can be used to compute the stress intensity factor K_I. When K_I has been approximated, the energy release rate can be calculated through (5.1).

The asymptotic displacement field in $2D$, (u, v), at the crack face subjected to a Modus I load can be described by a polar coordinate system (r, θ) attached to the crack tip.

$$u = K_I \frac{(1+\nu)}{4\pi E} \sqrt{2\pi r} \left((2\kappa - 1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right) + O(r), \tag{5.17}$$

$$v = K_I \frac{(1+\nu)}{4\pi E} \sqrt{2\pi r} \left((2\kappa + 1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right) + O(r), \tag{5.18}$$

where

$$\kappa = 3 - 4\nu \qquad \text{plane strain,} \tag{5.19}$$

$$\kappa = \frac{(3-\nu)}{(1+\nu)} \qquad \text{plane stress.} \tag{5.20}$$

Here, E is young's modulus, ν is Poisson's ratio.

Using the above equations, the stress intensity K_I can be expressed as

$$K_I = C(r,\theta) \cdot \frac{v}{\sqrt{r}}, \tag{5.21}$$

where the constant C is defined through the choice of r and θ. If one choose $\theta = \pi$ and plot K_I as a function of the radius r, the obtained graph can be used to evaluate K_I at the crack tip. This has

been done in Fig.(5.14). According to [5, Carlsson] the extrapolated value for $r = 0$ can be used to obtain the stress intensity factor. The error is usually below 5% depending on how well the crack tip singularity is resolved. The displacement field used for the extrapolation was taken from the adapted grid when evaluating the J-contour integral (path 4). The crack tip singularity is quite well resolved even though the goal-quantity did not include the displacements around the crack tip. Much better results should be obtained if the displacement field around the crack tip was the goal quantity.

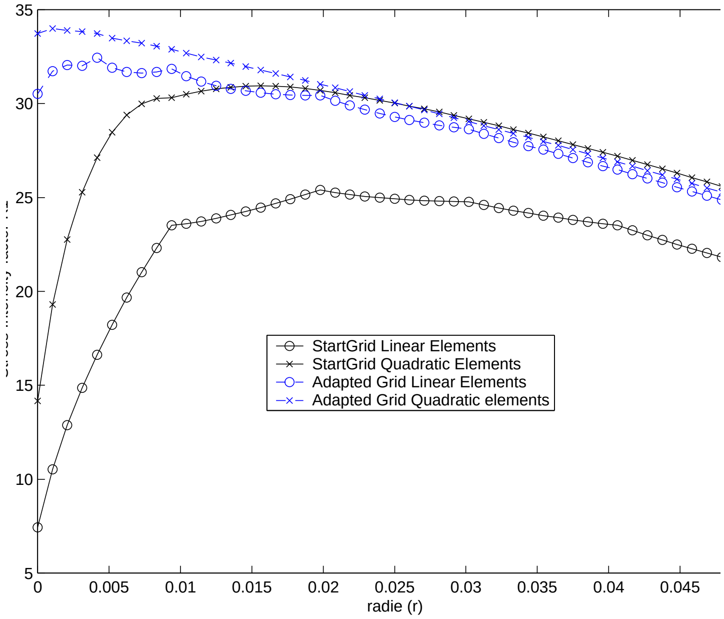

Figure 5.14: Extrapolation of the stress intensity factor K_I. The displacement field is from the J-contour integral path 4.

Chapter 6

Conclusions

6.1 Computing the Energy Release Rate

A number of techniques have been suggested for evaluating stress intensity factors from finite element results [12, Owen, Fawkes], [8, Knott], [5]. An adequate representation of the crack tip singularity is a problem common to most of these methods. The use of energy methods has the advantage that an exact modeling of the crack-tip behavior is not necessary.

In this thesis we have compared two different energy approaches for computing nonlinear elastic energy release rates at crack growth. The original *J-contour integral* and the more recently formulated *energy domain integral* have been evaluated with FEM using goal-oriented adaptivity. Their claimed path independence has been investigated numerically by evaluating the integrals on different contours and areas. The obtained results have been compared with an asymptotic displacement approach.

6.1.1 The J-contour integral

The J-contour integral approach has been successful. The obtained energy release rates have been confirmed with the asymptotic displacement approach. The difference between the two methods are below 1% for path 2-4. Figure(5.14) shows that the crack tip singularity is rather well resolved and there is almost no need for an extrapolation. The graph is comparable with graphs obtained using special crack tip singularity elements.

The path independence has been noticed for path 2-4. The difference compared to one another is below 1%. Path 1 differ compared to the other paths with approximately 6%.

6.1.2 The Energy Domain Integral

The energy domain integral approach has not been successful. This might bee due to the weight function q. In the literature [12] it is claimed that it is sufficient to use an arbitrary function that equals one at the crack tip and zero at the outer contour. In this thesis, the q function has been represented with continuous piecewise functions (quadratic basis functions) over the integration domain. The nodal values were set to zero outside the domain and to $1 - \frac{r}{r_0}$ within the outer radie r_0.

This way of defining q is maybe not appropriate. Other ways to define q should be investigated.

Instead of rewriting the contour integral as an area integral one can compute the contour integral on different layers of elements attatched to the crack tip. An extrapolation can be used to get the limiting quantity when $r \rightarrow 0$.

6.2 Limitations of Computational Fracture Mechanics

FEM can compute crack tip parameters such as the energy release rate, but these analysis cannot alone predict when and how the crack will grow. The obvious limitation that comes to mind is that the computations rely on continuum theory. A continuum doesn't contain grain boundaries, particles and obstacles which are present in a real material. Thus experiments are still important in fracture mechanics.

The development of multi-scale modeling (MSM) is interesting since that concept couples macro and micro scales in the material and it is at these length scales the dislocations occur.

The adaptive concept is applicable to any model that involves mathematical formulations that can be computed with FEM. The limitations is rather in the models describing the phenomena than in the computations itself. New and better theories for crack growth can be computed within the same framework presented in this thesis.

6.3 Error Control and Adaptivity

In this thesis two primal solutions ($\boldsymbol{u}$ and $\boldsymbol{u_h}^+$ in two different function spaces) have been calculated. The dual problem has been approximated in the same function space as $\boldsymbol{u_h}^+$. This approach makes the error representation formula (2.24) in a sense useless as an estimation of the error. Why estimate the error representation, when the best we can get -in absense of quadrature errors- is the difference between the two primal problems $Q(\boldsymbol{u})$-$Q(\boldsymbol{u_h}^+)$? The answer is that we do need the error representation formula to put indicators for refinements on the elements; but a more consistent way to represent the error is the very simple way

$$\lambda(\boldsymbol{e}) \approx Q(\boldsymbol{u_h}^+) - Q(\boldsymbol{u_h}). \tag{6.1}$$

This can be used during the implementation to test the code. Solve two primal problems in ${V_h}^p$ and ${V_h}^{p+1}$ and define

$$\begin{aligned} \boldsymbol{u_h} &:= \boldsymbol{u_h}^p \in {V_h}^p, && (6.2)\\ \boldsymbol{u_h}^+ &:= \boldsymbol{u_h}^{p+1} \in {V_h}^{p+1}. && (6.3) \end{aligned}$$

Solving the dual problem in ${V_h}^{p+1}$ results in

$$\frac{\tilde{\lambda}(\boldsymbol{e})}{Q(\boldsymbol{u_h}^+) - Q(\boldsymbol{u_h})} = 1, \tag{6.4}$$

where $\tilde{\lambda}(\boldsymbol{e})$ is the discretized error representation obtained from the computations. If this doesn't hold and the code isn't wrong, underlying reasons can be

- Wrong treatment of $\lambda(\boldsymbol{e})$. Instead of using $\boldsymbol{u_h}^+ + \boldsymbol{u_h}$ in the data to the dual problem one uses $2\boldsymbol{u_h}$ (see the example in chapter 3).
- Wrong quadrature - raise the quadrature order.
- Decrease the tolerance on the residual in the iterative solver.
- Floating point representation in the computer. (8:th decimal)

We do realise that solving two primal problems and a dual problem is extremely expensive. The amount of work can be reduced using fast iterative solvers such as *multi-grid*. The benefits are good indicators for the refinement of the next level.

One can also view the computations as using the result of the finer primal problems. This is clearly more accurate than the coarser primal but then we have not as correct error bound as above.

6.4 Mesh Design

In fracture mechanics the design of the finite element mesh has been much of an art form and a science in itself. Conventional linear elements have led to extremely fine meshes [12] in order to represent the $1/\sqrt(r)$ crack tip singularity accurately. This has led to the development of special crack-tip singularity elements by manipulation of the mid-side node positions. Higher order elements in the iso-parametric family has led to coarser meshes but the use of these special purpose elements has been advantageous [12].

The goal-oriented adaptive algorithm, using quadratic elements, is a way to avoid the a priori mesh constructions and special purpose elements. This leads to a more flexible and general solver for more complex problems.

6.5 Future Work and Development

Other methods proposed for computing the energy release rate at crack growth are based on virtual crack methods [1, 12]. A finite element analysis is performed for the original crack configuration and the strain energy U_1 is evaluated. The crack is then extended δa in some direction $\boldsymbol{r}$ and the strain energy U_2 is again calculated. The energy release rate is then

$$J = \frac{\delta U}{\delta a} = \frac{U_1 - U_2}{\delta a} \tag{6.5}$$

The proposed advantage of this method is that the crack will tend to grow in the direction $\boldsymbol{r}$ that results in the highest value of J.

This method demands that the strain energy is computed extremely accurate. The most economic and reliable method should be to use goal-oriented adaptivity with the strain energy as goal quantity.

The interpolant subtracted in the error representation can be exchanged from the nodal interpolant (used in this thesis) to the L_2 projection into V_h. Better indicators might be obtained.

The error representation can be taken further by applying integration by parts. Then one obtains an error representation on the form

$$\lambda(\boldsymbol{e}) = (\boldsymbol{R}, \boldsymbol{\varphi} - \boldsymbol{\varphi_h}) + \text{Jump terms from the element boundaries.} \quad (6.6)$$

How the spatial jump terms should be distributed is still under discussion.

Even though the J-contour integral was successful one can try to apply the divergence theorem to obtain an area integral similar to the generalized energy domain integral.

Bibliography

[1] T.L. Anderson. *Fracture Mechanics - Fundamentals and Applications.* CRC Press, 1991.

[2] J.J. Barton and L.R. Nackman. *Scientific and Engineering C++ -An introduction with Advanced Techniques and Examples.* Addison-Wesley, 1994.

[3] R. Becker and R. Rannacher. Weighted a posteriori error control in fe methods. Technical report, Institut für Angewandte Mathematik, Universität Heidelberg, 1996.

[4] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. Technical report, Institut für Angewandte Mathematik, Universität Heidelberg, 2001.

[5] J. Carlsson. *Brottmekanik.* Fingraf AB, 1985.

[6] L. Debnath and P. Mikusinski. *Introduction to Hilbert Spaces with Applications.* Academic Press, 1999.

[7] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations.* Studentlitteratur, 1996.

[8] J.F. Knott. *Fundamentals of Fracture Mechanics.* Butterworths, 1973.

[9] H.P. Langtanen. *Computational Partial Differential Equations.* Springer, 1999.

[10] M. Larson and J. Niklasson. A posteriori error estimation of functionals in elliptic problems: Experiments. Technical report, Chalmers Finite Element Center, 2001. PREPRINT 2001-04.

[11] F. Larsson. *Goal-oriented error measures and adaptive finite element procedures in solid mechanics.* lic. thesis, Chalmers, 2001.

[12] D.R.J. Owen and A.J. Fawkes. *Engineering Fracture Mechanics - Numerical Methods and Applications.* Pineridge Press Limited, 1983.

[13] R. Rannacher. Error control in finite element computations. Technical report, Institut für Angewandte Mathematik, Universität Heidelberg, 2000.

[14] R. Rannacher and F.-T. Suttmeier. A feed-back approach to error control in finite element methods: application to linear elasticity. *Computational Mechanics*, (19):434–446, 1997.

[15] J.R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.

[16] J.R. Shewchuk. *Triangle: A 2D quality mesh generator and delaunay triangulator.* http://www.cs.cmu.edu/ quake/tripaper/triangle0.html, 1996.

[17] J. Skansholm. *C++ direkt.* Studentlitteratur, 2000.

[18] B. Sundström. *Handbok och formelsamling i Hållfasthetslära.* Fingraf AB, Södertälje, 1998.

Appendix

Path Independence

The mathematical proof of path independence for the J-contour integral was presented by [15, Rice] By evaluating the J-contour integral along a closed

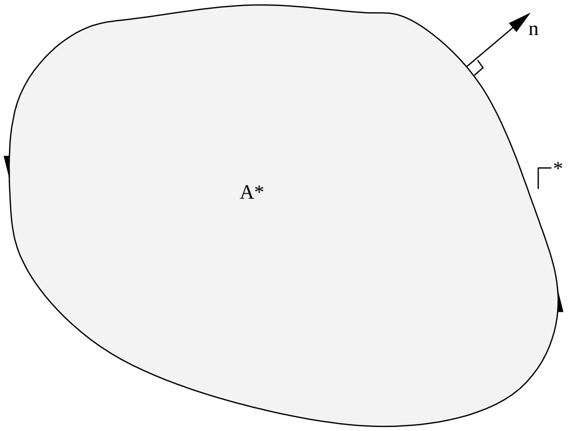

Figure 1: The J-contour integral over a closed contour Γ^*. The normal vector $\boldsymbol{n}$ is pointing out from the crack-tip. A^* is the area enclosed by Γ^*
.

contour Γ^* Fig.(1) the obtained J^*-contour integral reads

$$J^* = \int_{\Gamma^*} \left(W \; dx_2 - T_i \frac{\partial u_i}{\partial x_1} \; ds \right), \tag{7}$$

where $dx_2 = n_1 ds$. By applying the divergence theorem, (7) can be converted to an area integral

$$J^* = \int_{A^*} \left[\frac{\partial W}{\partial x_1} - \frac{\partial}{\partial x_j} (\sigma_{ij} \frac{\partial u_i}{\partial x_1}) \right] \, dA, \tag{8}$$

where A^* is the area enclosed by Γ^*, see Fig.(1). By (4.2) we have

$$\frac{\partial W}{\partial x_1} = \frac{\partial W}{\partial \varepsilon_{ij}} \frac{\partial \varepsilon_{ij}}{\partial x_1} = \sigma_{ij} \frac{\partial \varepsilon_{ij}}{\partial x_1}. \tag{9}$$

Note that (9) applies only when W has the properties of an elastic potential. Applying the strain- displacement relationship to (9) gives

$$\frac{\partial W}{\partial x_1} = \frac{1}{2} \sigma_{ij} \left[\frac{\partial}{\partial x_1} \left(\frac{\partial u_i}{\partial x_j} \right) + \frac{\partial}{\partial x_1} \left(\frac{\partial u_j}{\partial x_i} \right) \right] = \sigma_{ij} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_1} \right), \tag{10}$$

since $\sigma_{ij} = \sigma_{ji}$. By equilibrium $\frac{\partial \sigma_{ij}}{\partial x_j} = 0$ we have

$$\sigma_{ij} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_1} \right) = \frac{\partial}{\partial x_j} \left(\sigma_{ij} \frac{\partial u_i}{\partial x_1} \right), \tag{11}$$

which is identical to the the second term in (8). Thus the integrand in (8) vanishes and $J = 0$ for any closed contour.

Consider now two arbitrary contours, Γ_1 and Γ_2 around a crack-tip, illustrated in Fig.(2). If Γ_1 and Γ_2 are connected by segments along the crack face (Γ_3 and Γ_4) a closed contour is formed. The total J is the sum of the contributions and sums to zero (just proven in the above section)

$$J = J_1 + J_2 + J_3 + J_4 = 0. \tag{12}$$

On the crack face (Γ_3 and Γ_4), $T_i = dx_2 = 0$. Thus $J_3 = J_4 = 0$ and we obtain

$$J_1 = -J_2. \tag{13}$$

Therefore any arbitrary, counter-clockwise (normal vector pointing out from the crack-tip) will yield the same value of J; J is *path-independent*. See Fig.(2).

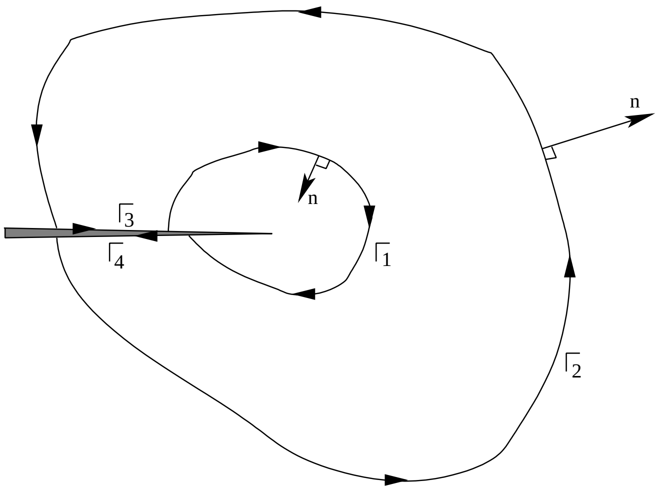

Figure 2: Closed contour around the crack-tip. The clock-wise integral over Γ_1 makes the normal vector $\boldsymbol{n}$ point in the opposite direction as the counter-clockwise integral over Γ_1.

The J-contour Integral As a Nonlinear Elastic Energy Release Rate

Consider a 2D cracked body bounded by Γ and let A denote the area of the body Fig.(3). The global coordinate axis (X_1, X_2) are random and the local coordinate axis (x_1, x_2) are attached to the crack tip. The vector $\boldsymbol{r}$ locates the midpoint of the crack. The local coordinates can be expressed as

$$x_1 = X_1 - a - r_1, \tag{14}$$

$$x_2 = X_1 - r_2. \tag{15}$$

Under quasi-static (no kinetic energy density) conditions and in the absence of body forces, the potential energy is given by

$$\Pi = \int_A W \, dA - \int_\Gamma T_i u_i \, ds. \tag{16}$$

Note: The contour integral over Γ is zero for prescribed displacements and fractions due to the fact that they are independent of a.

$$\frac{\partial \Pi}{\partial a} = \int_A \frac{\partial W}{\partial a} \, dA - \int_\Gamma T_i \frac{\partial u_i}{\partial a} \, ds. \tag{17}$$

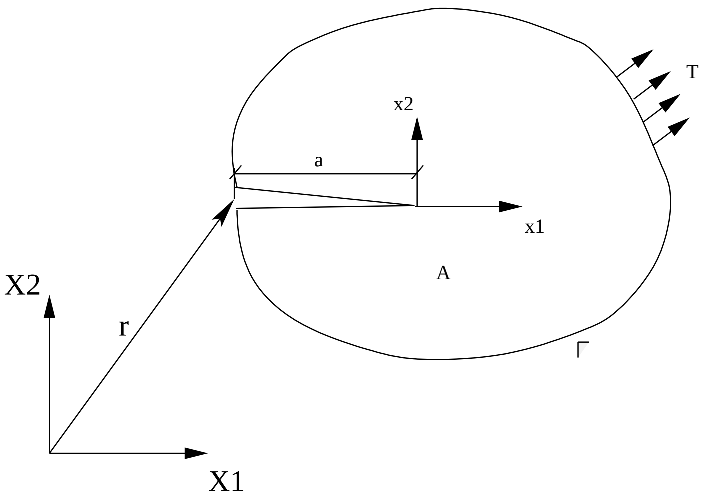

Figure 3: The J-contour integral as an energy release rate. a is the crack length. x_1, x_2 are the local coordinate axis located at the crack tip. X_1, X_2 are the global coordinate axis. The vector $\boldsymbol{r}$ is locating the crack. The vector $\boldsymbol{T}$ is traction on the boundary Γ.

When the crack grows, the coordinate axis moves and the derivative w.r.t. crack length is written as

$$\frac{d}{da} = \frac{\partial}{\partial a} + \frac{\partial x_1}{\partial a}\frac{\partial}{\partial x_1} = \frac{\partial}{\partial a} - \frac{\partial}{\partial x_1}, \tag{18}$$

since $\frac{\partial x_1}{\partial a} = -1$, see (14). Applying this to (17) yields

$$\frac{\partial \Pi}{\partial a} = \int_A \left(\frac{\partial W}{\partial a} - \frac{\partial W}{\partial x_1} \right) \, dA - \int_\Gamma T_i \left(\frac{\partial u_i}{\partial a} - \frac{\partial u_i}{\partial x_1} \right) \, ds. \tag{19}$$

By making the same assumptions as in (9) and (10) we obtain

$$\frac{\partial W}{\partial a} = \frac{\partial W}{\partial \varepsilon_{ij}} \frac{\partial \varepsilon_{ij}}{\partial a} = \sigma_{ij} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial a} \right). \tag{20}$$

The principle of virtual work gives

$$\int_A \sigma_{ij} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial a} \right) \, dA = \int_\Gamma T_i \frac{\partial u_i}{\partial a} \, ds. \tag{21}$$

Using (20) and (21) in (19) we found that the leading term in both parenthesis cancels out and we are left with

$$\frac{\partial \Pi}{\partial a} = \int_A -\frac{\partial W}{\partial x_1}\, dA + \int_\Gamma T_i \frac{\partial u_i}{\partial x_1}\, ds. \tag{22}$$

Applying the divergence theorem and multiplying with -1 leads to

$$-\frac{\partial \Pi}{\partial a} = \int_\Gamma \left(W\, n_1 - T_i \frac{\partial u_i}{\partial x_1} \right)\, ds = \int_\Gamma \left(W\, dx_2 - T_i \frac{\partial u_i}{\partial x_1} \right). \tag{23}$$

The last term is the J-contour integral and therefore, the J contour integral is equal to the *energy release rate* for a linear or nonlinear elastic material under quasi static conditions.

The Generalized Energy Release Rate

The generalized energy release rate is derived from Newton's second law and we refer to [1] for the derivation. The generalized energy release rate involves a vanishingly small contour Γ, and is defined as

$$J = \lim_{\Gamma \to 0} \int_\Gamma \left[(W + T)\, \delta_{1i} - \sigma_{ij} \frac{\partial u_j}{\partial x_1} \right] n_i\, d\Gamma. \tag{24}$$

W is the strain energy density and T is the kinetic energy density defined as

$$W = \int_0^{\epsilon_{ij}} \sigma_{ij}\, d\epsilon_{ij}, \tag{25}$$

$$T = \frac{1}{2} \rho \frac{\partial u_i}{\partial t} \frac{\partial u_i}{\partial t}. \tag{26}$$

Equation (24) is valid for time- and history-dependent material behavior. When applying (24) to time-dependent material W can be expressed as

$$W = \int_{t_0}^{t} \sigma_{ij} \dot{\epsilon}_{ij}\, dt, \tag{27}$$

where $\dot{\epsilon}_{ij}$ is the strain rate. The form of (24) involves of a vanishingly small contour Γ. Numerically this is not a problem since the integration can be carried out at some distance $r > 0$ from the crack tip. I.e $h \to 0$ for the continuous problem. In some textbooks [1] this proposed difficulty has led to a closed contour, constructed as illustrated in Fig.(4). The outer contour

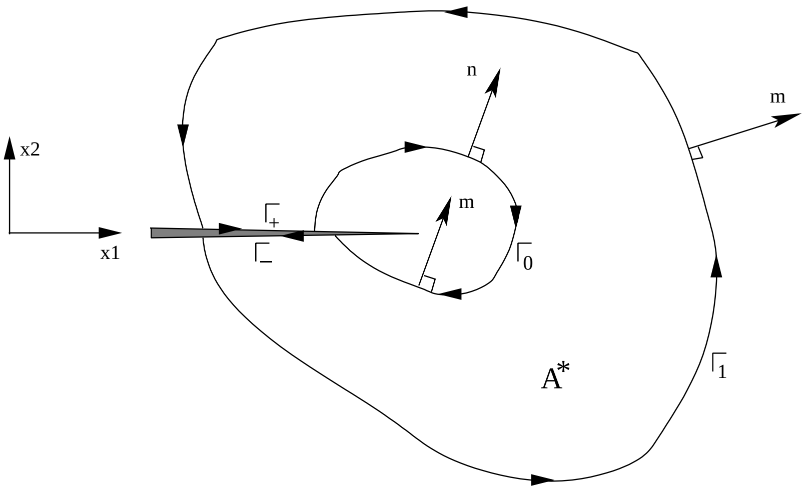

Figure 4: A closed contour formed by Γ_0 and Γ_1 when connected by Γ_+ and Γ_-. $\boldsymbol{m}$ is the counter-clockwise outward normal on Γ^*.

Γ_1 is finite while Γ_0 is vanishingly small. J could be evaluated on both Γ_0 and Γ_1 but only Γ_0 would give the correct value of J in the general case. For quasi static conditions where $T = 0$, (24) can be written

$$J = \int_{\Gamma_*} \left[\sigma_{ij}\frac{\partial u_j}{\partial x_1} - W\delta_{1i}\right] q\, m_i\, d\Gamma - \int_{\Gamma_+ + \Gamma_-} \sigma_{2j}\frac{\partial u_j}{\partial x_1}\, q\, d\Gamma, \tag{28}$$

where

$$\Gamma_* = \Gamma_1 + \Gamma_+ + \Gamma_- - \Gamma_0. \tag{29}$$

Γ_+ and Γ_- are upper and lower crack faces. q is an arbitrary but smooth weight function that is equal to unity on Γ_0 and zero on Γ_1. $\boldsymbol{m}$ is the counter-clockwise outward normal on Γ^*. Note that $\boldsymbol{m} = -\boldsymbol{n}$ on Γ_0. $m_1 = 0$ and $m_2 = \pm 1$ on Γ_+ and Γ_-.

Assuming traction-free crack faces (the second integral in (28) vanishes) and applying the divergence theorem to (28) results in

$$J = \int_{A^*} \frac{\partial}{\partial x_i}\left(\left[\sigma_{ij}\frac{\partial u_j}{\partial x_1} - W\delta_{1i}\right] q\right) dA \tag{30}$$

$$= \int_{A^*} \left(\left[\sigma_{ij}\frac{\partial u_j}{\partial x_1} - W\delta_{1i}\right] \frac{\partial q}{\partial x_i}\right) dA + \tag{31}$$

$$+\underbrace{\int_{A^*}\left[\frac{\partial}{\partial x_i}\left(\sigma_{ij}\frac{\partial u_j}{\partial x_1}\right)-\frac{\partial W}{\partial x_1}\right]q\,dA}_{=0\ \ Eq.(8)}, \tag{32}$$

i.e.

$$J \;=\; \int_{A^*}\left(\left[\sigma_{ij}\frac{\partial u_j}{\partial x_1}-W\delta_{1i}\right]\frac{\partial q}{\partial x_i}\right)\,dA. \tag{33}$$

This area approach has been implemented in this thesis.