IMPLEMENTATION OF A ROBUST SOLVER FOR PREDICTING HIGHLY LOCALIZED DEFORMATIONS IN MICROELECTRONICS

A Thesis
Presented to
The Academic Faculty

by

Jean-Baptiste P.M. Bouquet

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2011

IMPLEMENTATION OF A ROBUST SOLVER FOR PREDICTING HIGHLY LOCALIZED DEFORMATIONS IN MICROELECTRONICS

Approved by:

Professor Mohammed Cherkaoui,
Committee Chair
School of Mechanical Engineering
Georgia Institute of Technology

Professor Tudor Balan
Department of Mechanical Engineering
Art et Metiers ParisTech, Metz, France

Dr. Olaf van der Sluis
Department of Mechanical Engineering
Eindhoven University of Technology, Eindhoven, The Netherlands

Dr. Laurent Capolungo
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: May 2011

This thesis is dedicated to my parents who taught me the importance of working hard and the value of learning.

Their continued support and faith in me helped me in completing this study.

This thesis is also dedicated to my brother who has been a great motivator and inspirator.

ACKNOWLEDGEMENTS

I would like to express my greatest gratitude to many people for their help and support in completing this project. First of all, I want to thank my supervisor in Philips, Dr. Olaf van Der Sluis for lending his guidance throughout my thesis. He inspired me to step into finite element analysis and provided useful critical comments on my work, which help me conducting my research and organizing my ideas. I really appreciated his patience and understanding which have made this thesis possible. I also thank every people in Philips who helped me. I particularly devote my thank to Ruud Voncken for his expertise and counsel. I acknowledge all the time and effort he has put in to help me on the implementation and his explanations were really useful for me. I also thank Roy Engelen for his help with computer and software issues.

During this work I faced several difficulties but I got the help of people, specialists in this domain. I want to thank Cees Gelten, from MSc Software, for taking time from his really busy schedule to providing me with his wide experience of Marc/Mentat. Without him I may have never found the solution to some problems. I also want to express my gratitude to Joris Remmers and Clemens Verhoosel, both Assistant Professor at the department of Mechanical Engineering in Eindhoven University of Technology, for the time and the knowledge they imparted to me in order to explain me the subtleties and difficulties of the energy release method, throughout a generous assistance and especially kindly sharing.

I thank my two academic supervisors : Mohammed Cherkaoui, my Georgia Tech supervisor, who provided me the opportunity to work on an exciting and challenging

project I truly enjoyed, and Tudor Balan, my academic supervisor at Ecole Nationale Supérieure d'Arts et Métiers, who brought me support and enthusiasm in this project. I want to particularly acknowledge the help of Josyane Roschitz, Academic Office Director at Georgia Tech Lorraine, and Glenda Johnson, Academic Advisor at Georgia Tech, who have been so helpful and so attentive to all my questions on paperwork.

I really want to thank my friends Lucas Lallemant and Kotaro Fukasaku who shared with me this experience throughout France, Netherlands and the United States. They provided me support, enthusiasm and comradeship which made my life abroad even more colorful and joyful. At last, I want to thank my parents and my brother for continually encouraging me throughout the years. Their advices and support gave me an endless motivation to accomplish something important in my education and for studying abroad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Context	Variable	Associated physical magnitude	Unit
Fracture Mechanics	U_T	Total system energy	J
	U_E	Elastic strain energy	J
	U_{CR}	Energy needed for the crack propagation	J
	W	Work due to the applied external force	J
	G_c	Fracture toughness	$N.m^{-1}$
	K	Stress intensity factor (SIF)	$MPa.\sqrt{m}$
	Q	Separation history variable	m
Cohesive zone	$\boldsymbol{\delta}$	Separation vector	m
	$\boldsymbol{\tau}$	Traction vector	$N.m^{-2}$
	$\boldsymbol{\tau_{max}}$	Maximum traction	$N.m^{-2}$
	λ	Effective separation	m
	β	Ratio between the maximum shear traction and the maximum normal traction	
	δ_c	Critical opening	m
	G_c	Cohesive energy	$N.m^{-1}$
	$\boldsymbol{K_v}$	Virgin stiffness	$N.m^{-3}$
	D	Damage variable	
	Q	Separation history variable	m
Arc-Length method	$\boldsymbol{\epsilon}$	Cauchy Strain	
	$\boldsymbol{\sigma}$	Cauchy Stress	Pa
	$\mathbf{B}$	Strain-displacement matrix	
	E	Young's modulus	MPa
	ν	Poisson's ratio	
	G	Energy release rate	$J.m^{-2}$
	α	Load factor	
	$\mathbf{u}$	Displacement	m
	$\boldsymbol{K_t}$	Tangent stiffness matrix	$N.m^{-1}$
	$\boldsymbol{f}_{int}$	Internal force	N
	$\bar{\boldsymbol{f}}$	External total force	N
	P	Power of the external forces	W
	$\dot{\bar{\mathrm{V}}}$	Rate of the elastic potential energy	W
	$\mathbf{r}$	Residual vector	N

Context	Variable	Associated physical magnitude	Unit
Arc-Length method	$\hat{\mathbf{u}}$	Displacement due to the residual	m
	$\bar{\mathbf{u}}$	Displacement due to the external force	m
	q	Constraint equation	
	$\Delta\tau$	Arc-length	
	γ	Displacement factor	
	$\boldsymbol{C}$	Constraint matrix	
	κ	Control function of the local arc-length method	
Mathematical notation	Dx	Incremental value of the variable x	
	dx	Iterative value of the variable x	
	$\mathrm{x}_n^{(i)}$	ith iteration of the nth increment of the variable x	
	$\dot{\mathrm{x}}$	Derivative of the variable x with respect to time	

SUMMARY

With the improvement of the computational performances, simple nonlinear finite element analysis can be performed in a relatively short time. However, considering strongly non-linear models, the achievement of converged results may require many hours. The actual challenge resides in obtaining a reliable result in a reasonable time of calculation. The present work considers the simulation of crack propagations in multilayer electronic systems, such as System in Package (SIP) devices. It puts forward the implementation of a nonlinear finite element solver, developed for the MSc Software Marc/Mentat package software. This method is based on the energy released during the propagation of the crack, what offers the advantage of being directly linked to the failure process. In order to clearly understand the issues of this problem, this report makes a brief description of the fracture mechanics and reviews existing nonlinear finite element solvers, based on the arc-length method. After explaining the principle of the energy release solver and the different issues due to its implementation, its efficiency is compared to already implemented arc-length solvers.

CHAPTER I

INTRODUCTION

Nowadays, two main goals of micro-electronic companies can be clearly identified as the miniaturization of the electronic devices and the increasing number of functionalities. The combining of these two aims results in the emergence of systems in package, also called SIP, which consist in any combination of integrated circuits of different functionalities, incorporated into a single advanced package or module. In order to achieve such a challenge, SIP microsystems contain stacked, multiple, thin layers, manufactured using different materials, as illustrated in figure 1. Various circumstances, such as low velocity impacts, mechanical loading or temperature fluctuations lead to high stresses at the interface between two layers. Due to that phenomenon, these products are prone to interface failures, mainly in the form of delamination, which is a precursor to a critical failure mechanism. This irreversible process leads to reduction of stiffness and strength of laminates. It especially happens when considering composite materials, such as patterns of metal interconnects and mismatched matrix materials which present an insufficient adhesion at the interface.

The important scale difference of the individual materials and components in those devices makes them multiscale in nature, and the miniaturization makes structural dimensions pass from the macroscale to the nanoscale. That is a why this thesis is an entire part of the European NanoInterface Project [19], which is supported by the European Framework Program 7 and coordinated by Philips Applied Technologies, located in Eindhoven, Netherlands. The purpose of this project is to develop multiscale design tools on the three levels: atomistic/nanoscale, mesoscale,

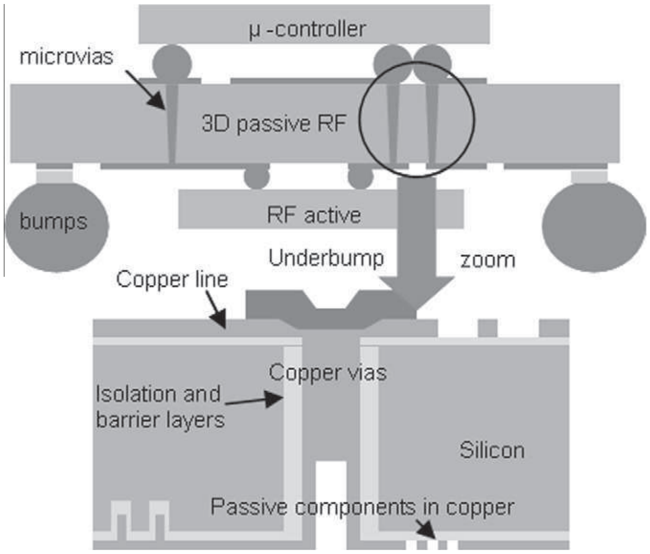


Figure 1: Example of SIP [4]

and continuum/macroscale. So in this approach, models at atomic level are linked to macroscopic models, represented by finite element analysis, as explained in [18].

Over the years, many finite element software packages have been developed, which provides a way to predict the failure behavior of materials, in order to optimize the structure of the devices. Considering structural analysis, they are mainly based on energy principles such as the virtual work principle or the minimum potential energy principle. In order to obtain ease and less computational cost, finite element methods use linear assumptions, expressed principally in terms of smallness of certain quantities in the formulation. These simplifications provide more tractable analysis of structures and in many problems, linearity assumptions lead to substantial idealization of the behavior of the system. In our considered case, energy-based linear elastic fracture mechanics has been massively used for delamination modeling of composites [10]. However, most physical processes contain complex interactions that are inherently nonlinear to a certain extent.

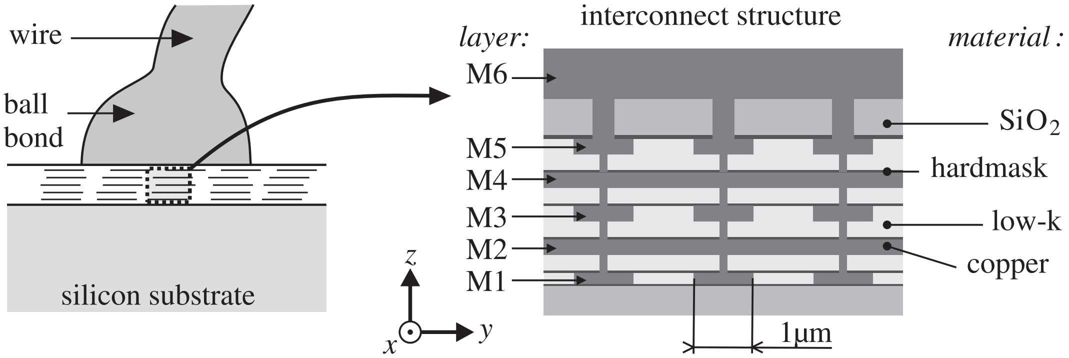


Figure 2: Interconnect structure of SIP [3]

In the case of delamination of composite devices found in SIP, the presence of high local stresses near the crack front causes inherently nonlinear constitutive properties, which often cannot be solved by a linear finite element method. The nonlinear software mainly used in this thesis is MARC, developed by MSc Software Corporation. It is the first commercial nonlinear finite element analysis program, and was introduced in 1972. Over the years different nonlinear solvers have been developed and implemented, such as Newton-Raphson procedure, automatic load step enlargement tool, line search algorithm or arc length algorithm. However, it is important to notice that even these solvers can easily break down when dealing with highly localized deformations. The main focus of this research is to improve the ability of finite element solvers to carry out such simulations, by developing a robust solver that is able to follow snap-back/snap-through behavior very efficiently, and so trace the equilibrium path properly. These snap-back/snap-through points, also known as limit points, are typically encountered when dealing with numerical predictions of highly localized deformations. This new solver [14] is a general arc-length method with a constraint equation based on the maximum energy dissipation of the system. Its direct link to the failure process should enable a stable convergence behavior.

CHAPTER II

FRACTURE MECHANICS

Fracture mechanics is an analysis dealing with the study of crack propagations in materials. This process starts with the nucleation of a micro-separation, which frequently occurs in multi-layered structures, in such a way that small cracks appear at inhomogeneities, where the stresses are particularly high. This phenomenon is an irreversible process, which reduces the strength and the stiffness of the material. If the applied load is increased, then the crack runs quickly into the material until it reaches a free edge or can no longer grow. This research considers a particular failure phenomenon which refers to interfacial cracking, also called delamination. Fracture mechanics provides two different criteria in order to determinate the crack propagation: the Griffith theory and the Irwin theory.

2.1 Griffith Approach

In 1920, Alan Arnold Griffith formulated the energetic approach about fracture propagation in [1]. For linear elastic fracture mechanics, this Griffith's energy balance is based on the 1^{st} law of thermodynamics, which expresses that energy can be transformed but can neither be created nor destroyed. By considering a brittle body containing an internal crack of length 2a subjected to an external applied load **F** at the boundary, the crack propagation is stable if the strain energy released during the propagation is absorbed by the increase in surface energy. So, the fracture mechanics can be considered as an equilibrium problem. The expressions of the thermodynamic equilibrium and the total system energy give the following system of equations :

$$\begin{cases} \dfrac{\partial U_T}{\partial a} = 0 \\ U_T = U_E + U_{CR} - W \end{cases} \tag{1}$$

where U_T is the total system energy, U_E is the elastic strain energy, U_{CR} is the energy needed for the crack propagation and W is the work due to the external applied load $\mathbf{F}$. By substitution, the Griffith's energy balance can be expressed as:

$$\underbrace{\frac{\partial W}{\partial a} - \frac{\partial U_E}{\partial a}}_{\text{Energy release rate: } G} = \underbrace{\frac{\partial U_{CR}}{\partial a}}_{\text{Fracture toughness : } G_c} \tag{2}$$

So, the crack grows if the energy release rate G is larger than the fracture toughness G_c required to propagate the crack:

$$G \geq G_c \tag{3}$$

2.2 Irwin theory

In 1957, Irwin formulated a new method in [9] to calculate the stress field in the area of the crack tip. This theory is based on the stress intensity factors (SIF) K, which are used to predict the stress state. The magnitude of K is a function of the loading, the geometry of the sample, the size and the location of the crack. The Irwin criterion compares the stress intensity factor K to a critical value K_c, called the tenacity or fracture toughness, which is a property of the material and defined for each fracture mode. The criterion states that:

$$\begin{cases} \text{if } K < K_c, \text{ there is no propagation of the crack.} \\ \text{if } K \geq K_c, \text{ the crack grows.} \end{cases} \tag{4}$$

This method takes into account the fracture modes. Indeed, by considering only plane crack propagations in their own plan, most general propagations can be reduced to the superposition of the three fracture modes defined below. They constitute a base for describing any fractures.

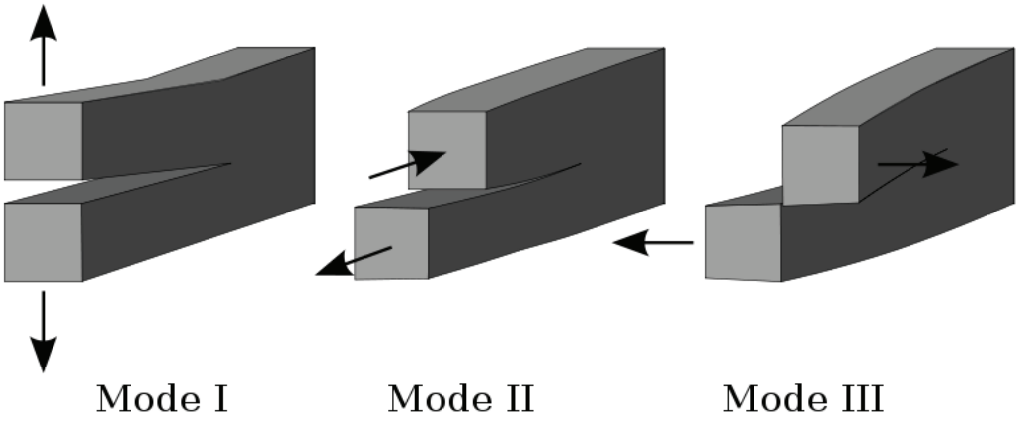


Figure 3: The three fracture modes [23]

- Mode I corresponds to the opening mode. It is characterized by a stress perpendicular to the crack faces.
- Mode II corresponds to the shearing mode. It is characterized by a shear stress perpendicular to the crack front.
- Mode III corresponds to the tearing mode. It is characterized by a shear stress parallel to the crack front.

The energy release rate G can be easily related to the stress intensity factor, by using the Young's modulus and the Poisson's ratio as shown below, for a mode I crack:

$$\begin{cases} G = \dfrac{K^2}{E} \text{ for plane stress problem.} \\ \\ G = \dfrac{K^2}{E}(1-\nu^2) \text{ for plane strain problem.} \end{cases} \tag{5}$$

Within finite elements analysis, several methods have been developed to predict the crack driving force. The most frequent used ones, which are explained below, are the J-integral method, the Virtual Crack Closure Technique, also called VCCT method and the cohesive zones model.

2.3 J-integral method

In 1968, Rice developed in [12] the J-integral method which expresses the energy release rate G in a material, in terms of field variables. The main idea is that the J-integral is a path-independent contour integral around the crack, as long as it stays within the part domain. It vanishes if it encloses no singularity. The J-integral is defined as:

$$\int_C \left(W_n - \boldsymbol{\sigma}_{ij} n_j \frac{\partial \boldsymbol{u}_i}{\partial x} \right) \mathrm{d}C \tag{6}$$

where W is strain energy density, n is the unit outward vector normal to an arbitrary contour C around the tip of the crack, $\boldsymbol{\sigma}_{ij}$ are stresses, $\mathbf{u}_i$ is the displacement. The J-integral is represented in figure 4.

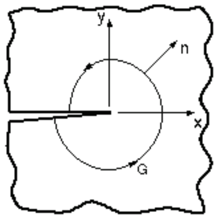


Figure 4: J-integral method

2.4 VCCT method

In 1977, Rybicki and Kanninen proposed in [7] the Virtual Crack Closure Technique, also called VCCT. This method is based on the assumption that the energy released during the crack extension is equal to the energy needed to close the crack and so it can determine the energy release rate associated with the crack. Figure 5 shows the evolution of the crack within the finite element model. The solver computes locally the total energy release rate by using the nodal forces at the crack front and the nodal displacements behind the crack front. This method is the most commonly employed in fracture analysis, but it requires a remeshing if the calculation of the crack path is needed, so it is not used in this thesis. All the calculations are explained in the NASA report [20].

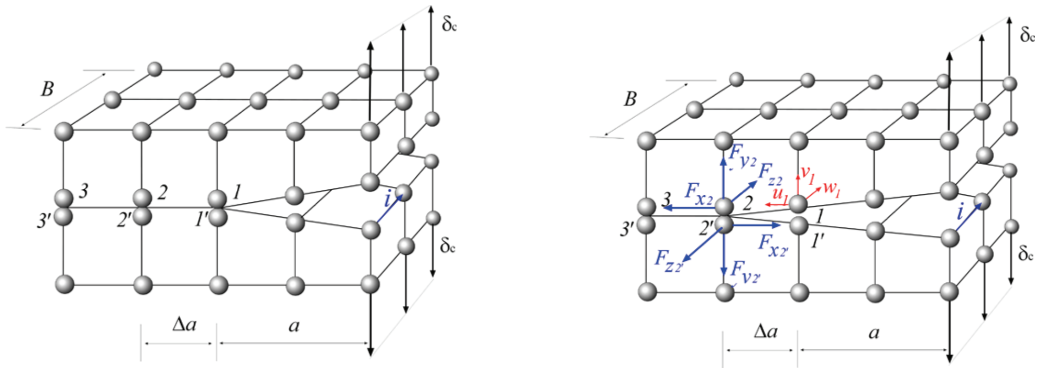


Figure 5: Opening of the crack for the VCCT method. Degrees of freedom at the crack front are released and the corresponding forces and displacements are computed.

The both methods presented above have some limitations. First a remeshing is needed each time the crack grows what generates extra time of calculation, second they are able to simulate the propagation of an existing crack, but they cannot predict its nucleation. The present research adopts another solution which does not encounter the previous problems. This solution is the cohesive zone model developed by Xu and Needleman [3, 21] and it is thoroughly developed in the section below.

2.5 Cohesive zone elements

In order to study the nucleation and the growth of interface delamination and brittle cracking, the sample is modeled with cohesive zone elements using nonlinear relations between only two parameters : the separation vector λ [m] of the two material faces at the interface and the traction vector τ $[N.m^{-2}]$, as pointed out on the figure 7. Accordingly, τ and λ have the same direction. These cohesive zone elements do not correspond to any physical material, but describe the cohesive forces occurring

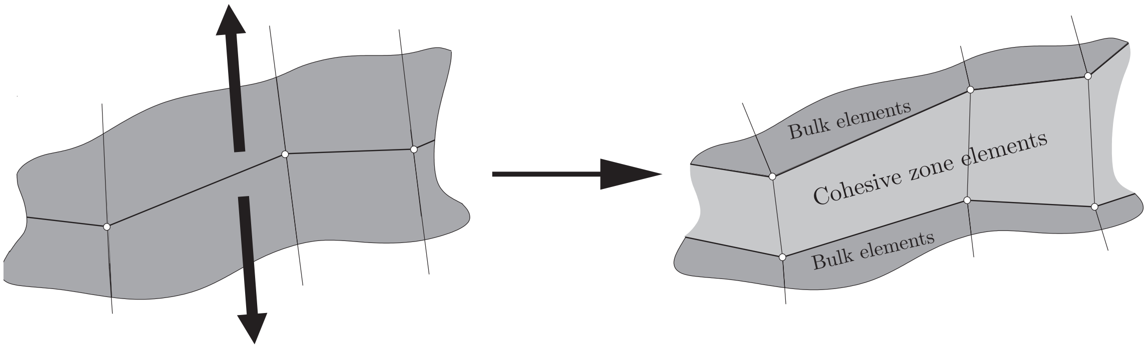


Figure 6: Opening of the cohesive zone [21]

when material elements are separated out. They especially depict the degradation of the adhesion between the materials along the interface. Therefore, cohesive zone elements are inserted at the interface, in between bulk elements, as shown on the figure 6. When damages occur, the cohesive zone elements open in order to represent the crack initiation or the crack propagation. During the opening, energy is dissipated until complete loss of traction occurs. The active zone wherein the energy is dissipated is called the process zone. However, according to the location of the cohesive zone elements, the propagation of the crack is predetermined, because it necessarily occurs along the cohesive zone. Nevertheless, the crack can propagate along any path where the cohesive zone elements are placed.

Many traction-separation laws, which describe the failure behavior, have been

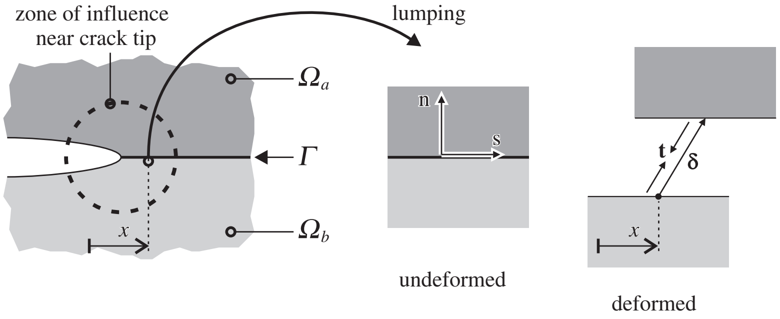


Figure 7: Parameters of the cohesive zone [3]

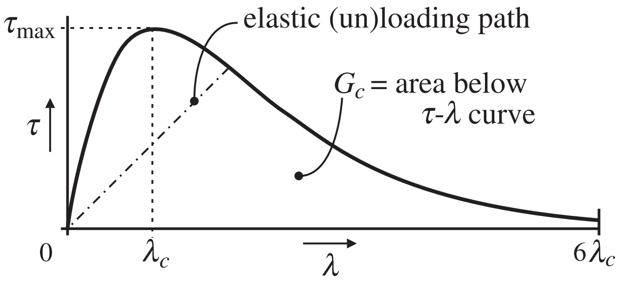


Figure 8: Smith-Ferrante traction-separation law [3]

proposed in the literature [17]. In most two-dimensional engineering applications, fracture energy is dissipated in both normal opening (mode I) and shearing (mode II) during a mixed-mode delamination process, so tractions have a normal and a shear component. Thus, the traction-separation laws depict the relations between both components (indicated by the subscripts **n** and **s**) of the separation vector and the traction vector, respectively defined by $\boldsymbol{\delta} = \delta_n \mathbf{n} + \delta_s \mathbf{s}$ and $\boldsymbol{\tau} = \tau_n \mathbf{n} + \tau_s \mathbf{s}$. The effective separation, defined in the equation (7), simplifies the formulation of a mixed-mode cohesive law.

$$\lambda = \sqrt{\langle \delta_n \rangle^2 + \beta^2 \delta_s^2} \tag{7}$$

$\beta = \tau_{s,max}/\tau_{n,max}$ is the ratio between the maximum shear traction and the maximum normal traction ; and the McCauley brackets means $\langle \delta_n \rangle = \frac{1}{2}(\delta_n + |\delta_n|)$, it implies that no negative separation component contributes to the effective separation. The model is also described with contact algorithms which prevent negative values of τ and λ. It means that penetration of the bulk materials is not possible.

The equation (8) presents the Smith-Ferrante exponential type of traction-separation law. This model, represented on the figure 8, gives a smooth traction-separation curve which is more stable than discontinuous laws, such as the bilinear one.

$$\tau = \tau_{max} \frac{\lambda}{\lambda_c} \exp\left(1 - \frac{\lambda}{\lambda_c}\right) \tag{8}$$

As clearly explained by Bas van Hal in [3], three constitutive parameters are commonly used, in exponential traction-separation laws, to describe the traction-separation law: the critical opening λ_c [m], the maximum traction τ_{max} $[N.m^{-2}]$ and the cohesive energy G_c $[N.m^{-1}]$. These parameters are linked together by the relation (9), so G_c is equal to the area below the curve τ-λ.

$$G_c = \int_0^\infty \tau(\lambda)\, d\lambda = \exp{(1)}\tau_{max}\lambda_c \tag{9}$$

The original law (8) can be rewritten in a damage mechanics formulation as in the equation (10).

$$\tau = K_v(1 - D)\lambda \tag{10}$$

K_v $[N.m^{-3}]$ is the virgin stiffness of the cohesive zone model, defined by the relation (11). It corresponds to the initial undamaged cohesive zone and it has a significant influence on the total elastic deformation.

$$K_v = \frac{G_c}{\lambda_c^2} = \frac{\tau_{max}\exp{(1)}}{\lambda_c} \tag{11}$$

In equation (10), D is the damage variable given by the relation (12). It starts from 0 for the undamaged case to end up with 1 after the crack propagation.

$$D = D(Q) = 1 - \exp\left(\frac{-Q}{\lambda_c}\right) \tag{12}$$

Q [m] is the separation history variable, which satisfies the following Kuhn-Tucker conditions:

$$(Q - \lambda) \geq 0, \qquad \dot{Q} \geq 0, \qquad \dot{Q}(Q - \lambda) = 0 \tag{13}$$

These relations assure that the unloading does not follow the same path than the loading but it follows the secant stiffness, as shown on the figure 8. That insures the irreversible behavior of the specimen.

However, it is important to notice that, like all elements, the cohesive zone elements are size-dependent. When the mesh of the model is too coarse, some numerical instabilities, such as oscillations in the equilibrium path, may happen. It can even lead to a divergent result. A refinement of the mesh is the perfect solution to eliminate this problem, however most of the times it requires long computational time. The arc-length control solver based on the released energy, presented in this report, will end up with a better solution without refining the mesh. Indeed, this method does not reduce the number of oscillations in the equilibrium path, but it deals with them in order to prevent any numerical issue. At the end, the computational time is greatly reduced.

CHAPTER III

ARC-LENGTH METHOD

3.1 Introduction

The basic idea of finite element analysis is to follow the equilibrium path of a considered structure. Finite element methods are generally based on constant step time solvers which are, depending of the used boundary conditions, either force (also called load) controlled solvers or displacement controlled solvers. Those constant step solvers can be coupled with a Newton-Raphson method in order to improve the computation of the simulation.

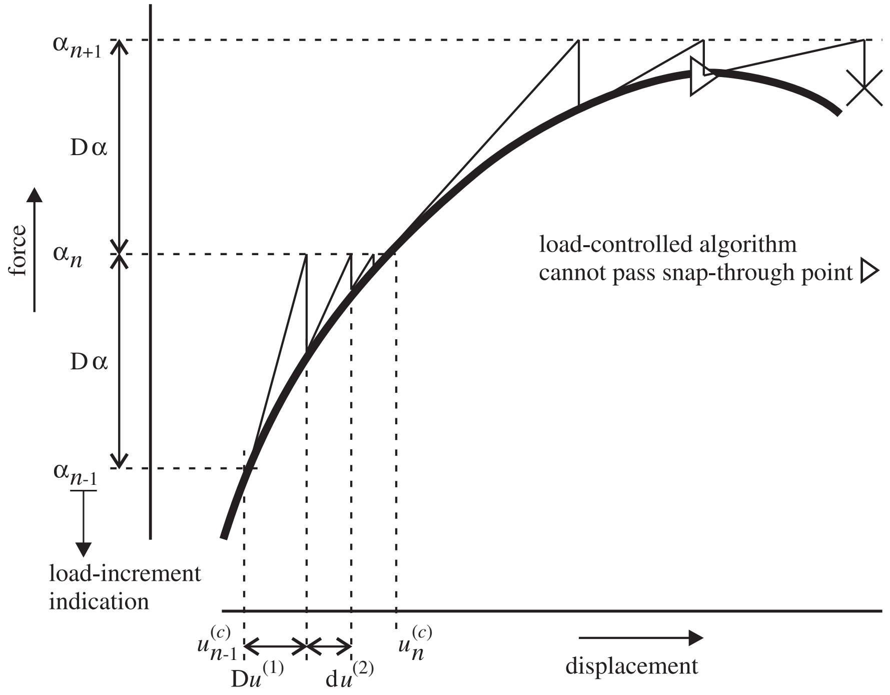


Figure 9: Snap-through presents in a force controlled solver [3]

However, these solvers often break down when trying to achieve most of the nonlinear finite element analysis with highly localized deformations. Indeed, as clearly shown on the figure 9, the force-controlled solver is not able to pass through snap-throughs and the displacement-controlled one is not able to handle snap-backs. Moreover, as explained later on in chapter 4, nonlinear analysis often provides equilibrium paths containing spurious oscillations. The easiest solution to deal with these limit points is to refine the mesh, but it implies an increase in computational cost. The second option consists in using other solvers, which are able to pass through limit points. These are the considered arc-length methods, which can calculate negative load factors.

3.2 Basics of the method

The equilibrium equation of a non-linear system, as explained in [8], can be written as:

$$\mathbf{R}_n^{(i)} = \mathbf{f_{int}}(\mathbf{u}_n^{(i)}) - \alpha_n^{(i)}\bar{\mathbf{f}} \tag{14}$$

where $\mathbf{f_{int}}$ is the internal force vector, which is a nonlinear function of the solution vector $\mathbf{u}$; $\bar{\mathbf{f}}$ is the external unit applied load vector and α is the scalar load factor, the dot product of these two terms gives the external force vector $\mathbf{f_{ext}}$. Finally, $\mathbf{R}$ is the *out-of-balance force vector*, also called *the residual vector.* This equation is called the residual force equation. Considering the non-linearity of this equation, the basic idea is to follow the equilibrium path as the control and state parameters vary by small steps.

The incrementation-iteration procedure, as shown on the figure 9, takes into account these both two parameters. In each increment, when considering force-controlled boundary conditions, the external load is increased in a stepwise incremental manner and so the corresponding equilibrium solution $\mathbf{u}$ is calculated iteratively, by differentiating the equation (14) for each iteration:

$$\mathbf{r}_n^{(i-1)} = \mathbf{K_t}(\mathbf{u}_n^{(i-1)})d\mathbf{u}_n^{(i)} - \bar{\mathbf{f}}d\alpha_n^{(i)} \tag{15}$$

$\mathbf{K_t}$ is the tangent stiffness matrix. This equation gives the following update iterative solution $d\mathbf{u}_n^{(i)}$:

$$\begin{aligned} d\mathbf{u}_n^{(i)} &= \mathbf{K_t}(\mathbf{u}_n^{(i-1)})^{-1}\mathbf{r}_n^{(i-1)} + \mathbf{K_t}(\mathbf{u}_n^{(i-1)})^{-1}\bar{\mathbf{f}}d\alpha_n^{(i)} \\ &= d\hat{\mathbf{u}}_n^{(i)} + d\bar{\mathbf{u}}_n^{(i)}d\alpha_n^{(i)} \end{aligned} \tag{16}$$

Different writings can be used to express these iterations, the displacement is updated in the following way:

$$\begin{cases} \mathbf{u}_{n+1}^{(c)} = \mathbf{u}_n^{(c)} + D\mathbf{u}_{n+1}^{(j)} \\ D\mathbf{u}_{n+1}^{(j)} = \displaystyle\sum_{p=0}^{j} d\mathbf{u}_{n+1}^{(p)} = d\mathbf{u}_{n+1}^{(0)} + \sum_{p=1}^{j} d\mathbf{u}_{n+1}^{(p)} \\ D\mathbf{u}_{n+1}^{(m)} = D\mathbf{u}_{n+1}^{(m-1)} + d\mathbf{u}_{n+1}^{(m)} \end{cases} \tag{17}$$

$\mathbf{u}_n^{(c)}$ is the total converged displacement at the start of the $(n+1)^{th}$ increment, $D\mathbf{u}_{n+1}^{(j)}$ is the $(n+1)^{th}$ incremental step, $d\mathbf{u}_{n+1}^{(m)}$ is the m^{th} iterative step of the $(n+1)^{th}$ incremental step. In the summation notation, 'j' represents the number of iterations after the result converges, which means that the equilibrium is achieved. These notations are clearly represented in the figure 10.

The arc-length method is an incremental-iterative procedure which treats the applied load factor α as an additional variable. So, the load factor follows the same scheme and with the same notations, it gives:

$$\begin{cases} \alpha_{n+1}^{(c)} = \alpha_n^{(c)} + D\alpha_{n+1}^{(j)} \\ D\alpha_{n+1}^{(j)} = \displaystyle\sum_{p=0}^{j} d\alpha_{n+1}^{(p)} = d\alpha_{n+1}^{(0)} + \sum_{p=1}^{j} d\alpha_{n+1}^{(p)} \\ D\alpha_{n+1}^{(m)} = D\alpha_{n+1}^{(m-1)} + d\alpha_{n+1}^{(m)} \end{cases} \tag{18}$$

This load factor is governed by an extra constraint equation (19), which refers to the arc-length of the equilibrium path. Thus, the new converged point is not only

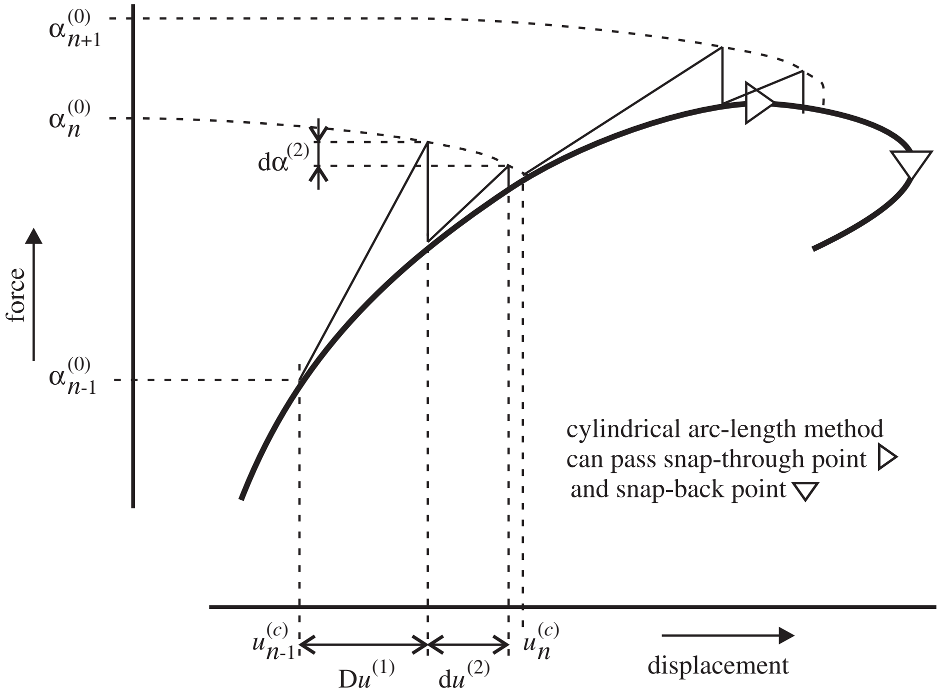


Figure 10: Arc-length method procedure for specific iteration [3]

dependent of a unique parameter, but it is a function of both displacement and load factor. Finally, the constraint equation is an expression of the arc-length and can be expressed as:

$$\boxed{D\mathbf{u}_n^{(i)^T} D\mathbf{u}_n^{(i)} + D\alpha_n^{(i)^2}\Phi^2\overline{\mathbf{f}}^T\overline{\mathbf{f}} - \Delta\tau = 0} \tag{19}$$

Φ is a scalar parameter that governs the relative contribution of the displacement and load increment. It is equal to 0 for the commonly used "cylindrical arc-length method". $\Delta\tau$ represents the arc-length parameter for the current increment. The following linearized version of the constraint equation 19, obtained by logarithmic differentiation and named q, is used by the solver:

$$q_n^{(i-1)} = 2D\mathbf{u}_n^{(i-1)^T} d\mathbf{u}_n^{(i)} + 2D\alpha_n^{(i-1)} d\alpha_n^{(i)}\Phi^2\overline{\mathbf{f}}^T\overline{\mathbf{f}} \tag{20}$$

Finally, by considering the (Z+1) unknowns, the Z equations obtained with the Z degrees of freedom and the constraint equation, the following matrix system is formed from (16) and (20):

$$\begin{bmatrix} d\mathbf{u}_n^{(i)} \\ d\alpha_n^{(i)} \end{bmatrix} = - \begin{bmatrix} \mathbf{K_t}(\mathbf{u}_n^{(i-1)}) & -\bar{\mathbf{f}} \\ 2D\mathbf{u}_n^{(i-1)^T} & 2D\alpha_n^{(i-1)}\Phi^2\bar{\mathbf{f}}^T\bar{\mathbf{f}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_n^{(i-1)} \\ q_n^{(i-1)} \end{bmatrix} \tag{21}$$

This method is called the global arc-length method due to the fact that it involves the control of all the degrees of freedom.

3.3 Local arc-length method

Sometimes, to obtain a convergent solution - especially in the case of localized deformations - the solver may require the control of a limited number of degrees of freedom. However, this method requires an extra step in order to select the appropriate degrees of freedom. Two different kinds of local arc-length method have to be distinguished as explained in [15]:

- local subplane method: the number of degrees of freedom in the constraint equation is confined, but the dominant degrees of freedom have to be selected beforehand. The subplane is commonly associated with significant incremental deformations present in the model.

- weighted subplane method: it uses a control function defined as a weighted linear sum of all the degrees of freedom or of functions of the degrees of freedom such as: strains or internal variables. The function is redefined at the beginning of each increment.

The method used is this paper is a weighted subplane method, introduced by Bas van Hal et al. [2]. Here, it is assumed that the damage D in the cohesive zone elements controls the load. Indeed the control function κ is a weighted sum of the damage values D in the Gauss points of all cohesive zone elements, as defined below:

$$\kappa = \sum_{gp} W_{gp} D_{gp} \tag{22}$$

The relations between increment and iterative updates are similar to the ones obtained in the global arc-length method:

$$\begin{cases} D\kappa_n^{(i+1)} = D\kappa_n^{(i)} + d\kappa_n^{(i+1)} \\ d\kappa_n^{(i)} = d\hat{\kappa}_n^{(i)} + d\bar{\kappa}_n^{(i)} d\alpha_n^{(i)} \end{cases} \tag{23}$$

This system of equations can be considered equivalent to the system (17), but using the control function κ instead of the load factor. Then the solution procedure is the same as previously explained, with the computation of a matrix system.

3.4 Energy release arc-length method

3.4.1 Theory

This method is a global arc-length method but the constraint equation is based on the energy release rate during failure. It was introduced by Gutiérrez [14], in such a way that the amount of dissipated energy in the deformation process is always a maximum for a given displacement and load increment. The advantage of this method is due to the released energy which is a global quantity and can thus even be used when the failure process is unknown. Moreover, such a constraint is linked up to the failure process, so a stable convergence behavior is obtained even for advanced stages of the equilibrium path. However, when the damage is not evolving, the rate of dissipation is equal to zero and so this method cannot be considered, so the solver has to switch to another method.

The energy release rate G during failure can be written as:

$$G = P - \dot{V} \tag{24}$$

where $\dot{V}$ is the rate of the elastic potential energy and P is the exerted power. This power, due to the external applied load is expressed as:

$$P = \alpha \bar{\mathbf{f}}^T \dot{\mathbf{u}} \tag{25}$$

The expression of the elastic potential energy depends on the constitutive behavior of the material. In our considered case of a geometrically linear model in presence of damage, the stored elastic potential energy can be stated as:

$$V = \frac{1}{2}\int \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega = \frac{1}{2}\mathbf{u}^T \int \mathbf{B}^T \boldsymbol{\sigma} d\Omega = \frac{1}{2}\mathbf{u}^T \mathbf{f_{int}} = \frac{1}{2}\alpha \mathbf{u}^T \bar{\mathbf{f}} \tag{26}$$

By differentiating this equation with respect to time, the rate of the elastic potential energy is easily calculated:

$$\dot{V} = \frac{1}{2}\alpha \dot{\mathbf{u}}^T \bar{\mathbf{f}} + \frac{1}{2}\dot{\alpha}\mathbf{u}^T \bar{\mathbf{f}} \tag{27}$$

By substituting equations (25) and (27) into (24), the amount of energy dissipated from the system is:

$$G = \frac{1}{2}(\alpha \dot{\mathbf{u}}^T - \dot{\alpha}\mathbf{u}^T)\bar{\mathbf{f}} \tag{28}$$

Finally, by applying an Euler-forward integration scheme in the particular form of the parametrization and by considering the arc-length path parameter $\Delta\tau$, which corresponds to the amount of energy dissipated in one increment, the new energy release constraint equation is given by:

$$\boxed{\frac{1}{2}(\alpha_n^{(0)} D\mathbf{u}_n^{(i)^T} - D\alpha_n^{(i)}\mathbf{u}_n^{(0)^T})\bar{\mathbf{f}} - \Delta\tau = 0} \tag{29}$$

A geometrical interpretation of the method is presented in figure (11) , where the shaded area corresponds to the released energy during a dissipative increment.

The augmented system of equations can be solved with the following matrix expression, in an iterative manner by using a Newton-Raphson approach:

$$\begin{bmatrix} d\mathbf{u}_n^{(i)} \\ d\alpha_n^{(i)} \end{bmatrix} = - \begin{bmatrix} \mathbf{K_t}(\mathbf{u}_n^{(i-1)}) & -\bar{\mathbf{f}} \\ \frac{1}{2}\alpha_n^{(0)}\bar{\mathbf{f}}^T & -\frac{1}{2}\mathbf{u}_n^{(0)^T}\bar{\mathbf{f}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_n^{(i-1)} \\ q_n^{(i-1)} \end{bmatrix} \tag{30}$$

The loss of the band structure of the matrix can be compensated with the use of the Sherman-Morrison formula as explained in [11] and in appendix D, in order to

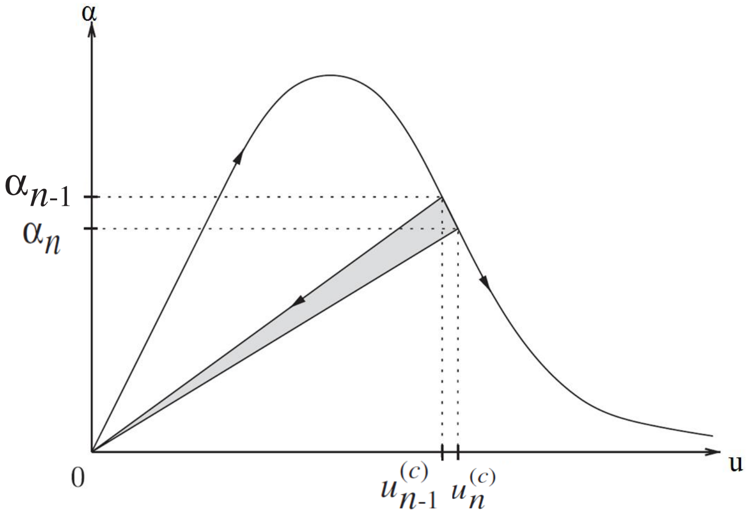


Figure 11: Geometrical interpretation of the energy release method

obtain the solution system:

$$\begin{bmatrix} d\mathbf{u}_n^{(i)} \\ d\alpha_n^{(i)} \end{bmatrix} = \begin{bmatrix} d\hat{\mathbf{u}}_n^{(i)} \\ q_n^{(i-1)} \end{bmatrix} - \frac{1}{\mathbf{g}^T d\bar{\mathbf{u}}_n^{(i)} + w} \begin{bmatrix} d\bar{\mathbf{u}}_n^{(i)}(\mathbf{g}^T d\hat{\mathbf{u}}_n^{(i)} + q_n^{(i-1)}) \\ \mathbf{g}^T d\hat{\mathbf{u}}_n^{(i)} + (1 + \mathbf{g}^T d\bar{\mathbf{u}}_n^{(i)} + w) q_n^{(i-1)} \end{bmatrix} \tag{31}$$

with the following variables:

$$\begin{cases} \mathbf{g} = \dfrac{1}{2}\alpha^c_{(n-1)}\bar{\mathbf{f}} \\ w = -\dfrac{1}{2}\mathbf{u}^c_{(n-1)}{}^T\bar{\mathbf{f}} \end{cases} \tag{32}$$

The variables $\mathbf{u}^c_{(n-1)}$ and $\alpha^c_{(n-1)}$ are respectively equal to the variables $\mathbf{u}^0_{(n)}$ and $\alpha^0_{(n)}$.

3.4.2 Implementation

The main relations which have to be implemented, are introduced in the above presentation of the method. The flowchart of the appendix B gives the global solution procedure, in a typical finite element code. The implementation part of this project considers more particularly the 'matrix solution' procedure. The flowchart of this procedure is given in figure 12.

The system (31) corresponds to the matrix representation of a set of two equations, which can be simplified as shown on the system below. Finally, the easiest way to calculate the new iterative step is given below:

$$\boxed{\begin{cases} d\alpha_n^{(i)} = -\dfrac{\mathbf{g}^T d\hat{\mathbf{u}}_n^{(i)} + q_n^{(i-1)}}{\mathbf{g}^T d\bar{\mathbf{u}}_n^{(i)} + w} \\ d\mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} + d\alpha_n^{(i)} d\bar{\mathbf{u}}_n^{(i)} \end{cases}} \tag{33}$$

The second equation of the system is dependent on the first one. Performing (33) requires the value of both incremental displacement due to the residual $d\hat{\mathbf{u}}_n^{(i)}$ and incremental displacement due to the external forces $d\bar{\mathbf{u}}_n^{(i)}$. These values are already calculated in an earlier step, so the value of the incremental load factor $d\alpha_n^{(i)}$ is easily performed. Once this value is obtained, it is substituted in the calculation of the incremental displacement $d\mathbf{u}_n^{(i)}$. It can be noticed that the second equation of the above system (33) is the same as the one used previously in (16), for the 'default' arc-length method, which is already available in Marc. This is an important point because this part of the Marc solver can be conserved and only the one with the calculation of the iterative load factor has to be rewritten.

3.4.3 Estimation of the arc-length

The step-size estimation is a key parameter in the success of any non-linear method. So, in order to calculate the complete equilibrium path in as few steps as possible, the arc-length parameter $\Delta\tau$ needs to be readjusted at the beginning of each increment. The aim of this adjustment is to obtain a number of iterations equal to the optimal value k_{opt} given by Riks [6], which is usually considered to be 5 in these kind of computations. Several adjustment formula have been proposed in the literature. The one provided here is the following :

$$\Delta\tau_{n+1} = \Delta\tau_n \left(\frac{1}{2}\right)^p \tag{34}$$

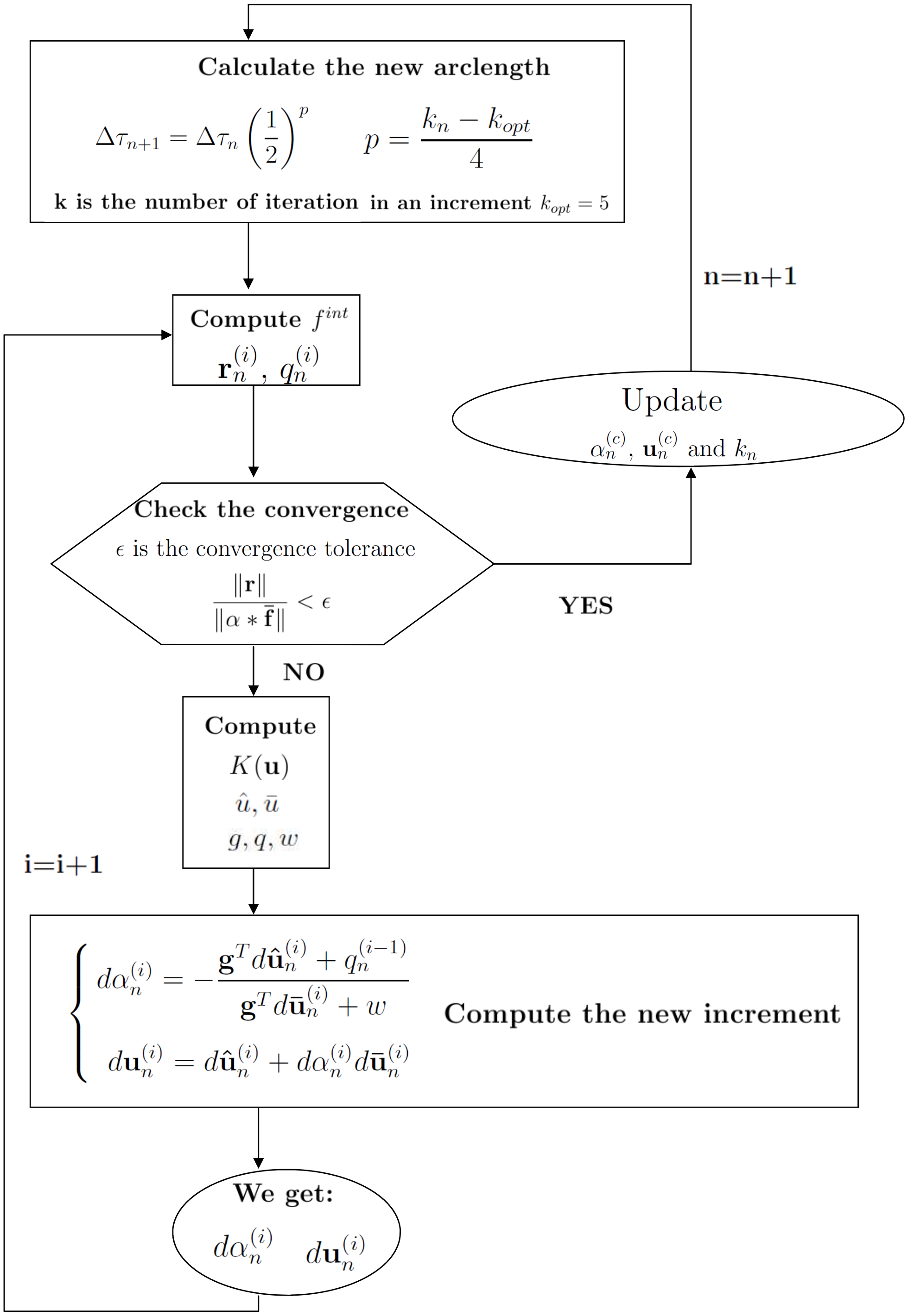

Calculate the new arclength
$\Delta\tau_{n+1} = \Delta\tau_n \left(\frac{1}{2}\right)^p$
$p = \frac{k_n - k_{opt}}{4}$
k is the number of iteration in an increment $k_{opt} = 5$
n=n+1
Compute f^{int}
$\mathbf{r}_n^{(i)}$, $q_n^{(i)}$
Update
$\alpha_n^{(c)}$, $\mathbf{u}_n^{(c)}$ and k_n
Check the convergence
ϵ is the convergence tolerance
$\frac{\|\mathbf{r}\|}{\|\alpha * \bar{\mathbf{f}}\|} < \epsilon$
YES
NO
Compute
$K(\mathbf{u})$
$\hat{u}, \bar{u}$
g, q, w
i=i+1
$d\alpha_n^{(i)} = -\frac{\mathbf{g}^T d\hat{\mathbf{u}}_n^{(i)} + q_n^{(i-1)}}{\mathbf{g}^T d\bar{\mathbf{u}}_n^{(i)} + w}$
$d\mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} + d\alpha_n^{(i)} d\bar{\mathbf{u}}_n^{(i)}$
Compute the new increment
We get:
$d\alpha_n^{(i)}$ $d\mathbf{u}_n^{(i)}$

Figure 12: Flowchart of the energy release procedure.

with:

$$p = \frac{k_n - k_{opt}}{4} \tag{35}$$

The relation (34) clearly shows that the new arc-length parameter is based on its previous value and on the number of iterations required in the previous step, represented by k_n.

3.4.4 Switch

This energy release solver can only be used when a certain quantity of energy is dissipated in a single load step. However, in most of the cases, a non-dissipative part exists in the equilibrium path. For instance, in both used benchmarks, the initial deformation of the beam is only elastic, which means that no energy is released, the fracture arrives later in the simulation. In order to pass through these non-dissipative parts, other constraint equations can be used. Since this study is especially about the highly localized deformations, which generally result in snap-backs, Crisfield's method appears to be a logical alternative. When the fracture starts and enough energy is dissipated, the solver switches from Crisfield's method to the energy release one. As shown in [5], several switching criteria can be selected, the one used in this paper takes into account the amount of energy dissipated in Crisfield's method. When it exceeds a predefined value, chosen by the user, the solver switches to the energy release method. A reverse switch is also configured when the ratio $\frac{\Delta\tau}{\|\Delta\alpha\|}$ is smaller than the predefined value.

3.4.5 Differences between displacement and force prescribed boundary conditions

In order to depict the effect of the external environment on the model, the user has to specify boundary conditions in order to correctly constrain the model. Different kinds of boundary conditions exist to represent most of the physical phenomenon, such as mechanical, thermal and electromagnetic. In our considered case, mechanical

Table 1: Stored variables in Marc, used by the energy release solver

Variable	**Physical meaning**	**Marc variable**
$d\alpha_n^{(i)}$	new iterative load factor	xlam
$d\hat{\mathbf{u}}_n^{(i)}$	iterative displacement due to the external force	tx3_d
$d\bar{\mathbf{u}}_n^{(i)}$	iterative displacement due to the residual	dsx_d
$\alpha_{n-1}^{(c)}$	load factor at the end of the previous increment	autacc
$\mathbf{u}_{n-1}^{(c)}$	displacement at the end of the previous increment	dsxts_d
$\bar{\mathbf{f}}$	external unit force	pload_d

boundary conditions will be prescribed, however several kinds of boundary conditions can be selected, such as displacement, acceleration point load, edge load and face load. In the two benchmarks presented in the next chapter, the boundary conditions are prescribed directly on the nodes by using displacement boundary conditions and/or force boundary conditions.

3.4.5.1 Force prescribed boundary conditions

When the model is prescribed with force boundary conditions, the solver automatically scales the total external force $\mathbf{f_{ext}}$ in such a way that:

$$\mathbf{f_{ext}} = \alpha\bar{\mathbf{f}} \tag{36}$$

This expression of the external force is exactly the same as the one used by the solver, as explained in (14). So, the application of these boundary conditions into the solver is straightforward. If we now consider the new update values, the system to solve is the system (33) with the variables of (32).

The difficulty of implementing the new energy release arc-length method is principally based on the access to the needed data. For proper implementation, the variables required from the software Marc are given in the Table 1.

The calculation of the constraint equation based on the energy release rate $q_n^{(i-1)}$ requires to take into account the predictor step differently than the other iteration

Table 2: Calculated variables by the energy release solver

Variable	**Physical meaning**	**Marc variable**
$\alpha_n^{(0)}$	load factor at the end of the previous increment	autacc
$D\mathbf{u}_n^{(i)}$	actual incremental displacement	dynd_d
	predictor of the incremental displacement	dsx2_d
$D\alpha_n^{(i)}$	actual incremental load factor	curper
	predictor of the incremental load factor	curprs
$\mathbf{u}_n^{(0)}$	displacement at the end of the previous increment	dsxts_d
$\Delta\tau$	arc-length parameter	arclen

steps, according to the logarithmic differentiation of the equation (29):

$$q_n^{(i-1)} = \frac{1}{2}(\alpha_n^{(0)} d\mathbf{u}_n^{(i)^T} - d\alpha_n^{(i)} \mathbf{u}_n^{(0)^T})\bar{\mathbf{f}} \tag{37}$$

The variables used in the calculation of the constraint equation are given in the Table 2.

3.4.5.2 Displacement prescribed boundary conditions

When considering displacement prescribed boundary conditions, the solver calculates the nodal displacement of the model. However, some displacements are already prescribed, the displacement vector has to be decomposed according to :

$$\mathbf{u} = \mathbf{C}\mathbf{u_f} + \mathbf{u_p} \tag{38}$$

where $\mathbf{u_f}$ and $\mathbf{u_p}$ are respectively the free and the prescribed displacement and $\mathbf{C}$ is the constraint matrix whose size depends on the number of prescribed degrees of freedom. In most of the software package, the constraint matrix is directly calculated. This is what happens in MSc Software Marc, so we do not really need to take into consideration the constraint matrix.

As for prescribed displacement boundary conditions, the solver automatically scales the prescribed boundary conditions, so in case of displacement prescribed boundary conditions, we end up with the following equation about the prescribed displacement, which is written in terms of a unit prescribed displacement $\mathbf{\bar{u}_p}$ times a displacement factor γ:

$$\mathbf{u_p} = \gamma \mathbf{\bar{u}_p} \tag{39}$$

In order to calculate the different needed expressions given in the table 2, the external force is required. But, no external force $\mathbf{f_{ext}}$ is applied, so it is expressed as the product of the stiffness matrix and the displacement, and we come up with:

$$\mathbf{Ku} = \mathbf{f_{ext}} \tag{40}$$

Then, from here we can calculate the energy release rate, such that:

$$q_n^{(i-1)} = \frac{1}{2}\mathbf{K}(\mathbf{u}_n^{(i-1)})\mathbf{\bar{u}}(\alpha_n^{(0)} d\mathbf{u}_n^{(i)} - d\alpha_n^{(i)} \mathbf{u_n^{(0)}}) \tag{41}$$

Finally, by considering system (33), the solution is easily achieved, however two more variables need to be adapted according to:

$$\begin{cases} \mathbf{g} = \dfrac{1}{2}\alpha_n^{(0)}\mathbf{K}(\mathbf{u}_n^{(i-1)})\mathbf{\bar{u}} \\ w = \dfrac{1}{2}\mathbf{u_n^{(0)}}^T\mathbf{K}(\mathbf{u}_n^{(i-1)})\mathbf{\bar{u}} \end{cases} \tag{42}$$

This method has been implemented into Mentat, however some troubles are encountered to obtain the current stiffness matrix. A common block has been created for this purpose which basically works, but the values are not updated. Right now, the help of MSc Software is required. The implementation is, however, straightforward.

MATHEMATIC EXPRESSION OF THE CONSTRAINT EQUATION

Method	Constraint equation	Matrix system	New iterative values
Newton-Raphson	$D\mathbf{u}_n^{(i)} = constant > 0$	$d\mathbf{u}_n^{(i)} = -\mathbf{K_t}(\mathbf{u}_n^{(i-1)})^{-1}\mathbf{r}_n^{(i-1)}$	$\begin{cases} d\alpha_n^{(i)} = 0 \\ d\mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} \end{cases}$
Crisfield AL	$D\mathbf{u}_n^{(i)T} D\mathbf{u}_n^{(i)} - \Delta\tau = 0$	$\begin{bmatrix} d\mathbf{u}_n^{(i)} \\ d\alpha_n^{(i)} \end{bmatrix} = -\begin{bmatrix} \mathbf{K_t}(\mathbf{u}_n^{(i-1)}) & -\bar{\mathbf{f}} \\ 2D\mathbf{u}_n^{(i-1)T} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_n^{(i-1)} \\ q_n^{(i-1)} \end{bmatrix}$	$\begin{cases} a_1 d\alpha_n^{(i)^2} + a_2 d\alpha_n^{(i)^2} + a_3 = 0 \\ a_1, a_2, a_3 = f(d\hat{\mathbf{u}}_n^{(i)}, d\hat{\mathbf{u}}_n^{(i)}, \Delta\tau) \\ \mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} + d\alpha_n^{(i)} d\bar{\mathbf{u}}_n^{(i)} \end{cases}$
Local AL	$D\kappa_n^{(i)T} D\kappa_n^{(i)} - \Delta\tau = 0$ with $\kappa = \sum_{gp} W_{gp} D_{gp}$		$\begin{cases} d\alpha_n^{(i)} = \frac{-D\kappa_n^{(i-1)} - d\hat{\kappa}_n^{(i)} + \Delta\tau}{d\bar{\kappa}_n^{(i)}} \\ d\mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} + d\alpha_n^{(i)} d\bar{\mathbf{u}}_n^{(i)} \end{cases}$
Energy Release AL	$\frac{1}{2}(\alpha_n^{(0)} D\mathbf{u}_n^{(i)^T} -$ $D\alpha_n^{(i)} \mathbf{u}_n^{(0)^T})\bar{\mathbf{f}} - \Delta\tau = 0$	$\begin{bmatrix} d\mathbf{u}_n^{(i)} \\ d\alpha_n^{(i)} \end{bmatrix} = -\begin{bmatrix} \mathbf{K_t}(\mathbf{u}_n^{(i-1)}) & -\bar{\mathbf{f}} \\ \frac{1}{2}\alpha_n^{(0)}\bar{\mathbf{f}}^T & -\frac{1}{2}\mathbf{u}_n^{(0)^T}\bar{\mathbf{f}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_n^{(i-1)} \\ q_n^{(i-1)} \end{bmatrix}$	$\begin{cases} d\alpha_n^{(i)} = \frac{\mathbf{g}^T d\hat{\mathbf{u}}_n^{(i)} - q_n^{(i-1)}}{\mathbf{g}^T d\bar{\mathbf{u}}_n^{(i)} - w} \\ d\mathbf{u}_n^{(i)} = d\hat{\mathbf{u}}_n^{(i)} + d\alpha_n^{(i)} d\bar{\mathbf{u}}_n^{(i)} \end{cases}$

GEOMETRIC REPRESENTATION OF THE CONSTRAINT EQUATION

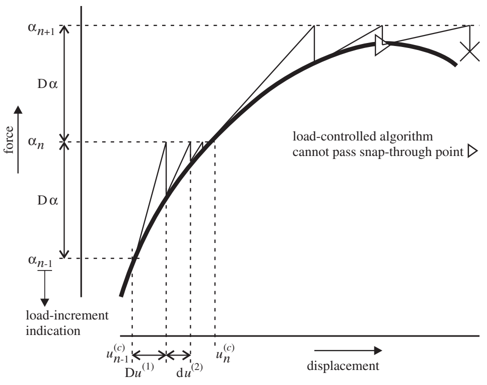
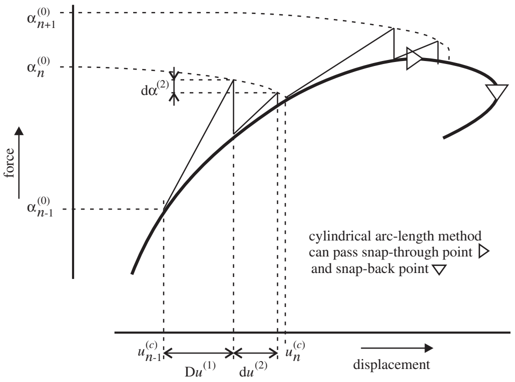
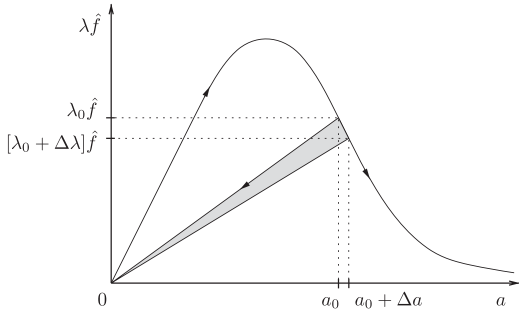
	Newton-Raphson	Crisfield arc-length
Graphic representation	 α_{n+1} $D\alpha$ force α_n $D\alpha$ α_{n-1} load-increment indication load-controlled algorithm cannot pass snap-through point ▷ $u_{n-1}^{(c)}$ $Du^{(1)}$ $du^{(2)}$ $u_n^{(c)}$ displacement	 $\alpha_{n+1}^{(0)}$ $\alpha_n^{(0)}$ $d\alpha^{(2)}$ force $\alpha_{n-1}^{(0)}$ cylindrical arc-length method can pass snap-through point ▷ and snap-back point ▽ $u_{n-1}^{(c)}$ $Du^{(1)}$ $du^{(2)}$ $u_n^{(c)}$ displacement
Encountered problems	• Not able to pass through limit points. • Not able to treat brittle failure.	• Difficulties in choosing the good $d\alpha_n^{(i)}$. • Unloading happens under strong localizations.
	Local arc-length	**Energy Release arc-length**
Graphic representation	The graphic representation is the same as for the Crisfield arc-length method. The method is similar, only the weights put on the degrees of freedom is different.	 $\lambda\hat{f}$ $\lambda_0\hat{f}$ $[\lambda_0 + \Delta\lambda]\hat{f}$ 0 a_0 $a_0 + \Delta a$ a
Encountered problems	• No problem : accurate results. • The time of calculation is quite long.	• No problem : accurate results. • The time of calculation is shorter.

Figure 13: Comparison of the different arc-length methods

CHAPTER IV

BENCHMARKS

4.1 Double Cantilever Beam benchmark (DCB)

4.1.1 Description of the benchmark

The double cantilever beam (DCB) test is an often used two-dimensional benchmark test for cohesive zone elements. This model, presented in the figure 14, consists of two strips of length L = 100 mm, width W = 30 mm and height H = 3 mm, glued together with cohesive zone elements which are placed at the interface. An initial crack of length a_0 is inserted at one end of the interface. During the test, analyzed under plane strain conditions, the two parts of the beam are separated by pulling apart in mode I. Indeed, the two end points of the beam are given a displacement in opposite y- directions up to u = 15 mm, while the other end of the sample is fixed in x- and y- direction. In this benchmark, in order to change the brittleness of the cohesive zone, the critical opening is decreased while keeping the cohesive energy G_c constant. The brittleness of the interface is a relative notion, which can be defined as the ratio between the size of the mesh and the size of the critical opening. When the critical opening decreases, the interface becomes more brittle, which is in accordance with the reality, however this also implies some calculation difficulties ; that is why this test considers different critical openings. The beam is meshed with 8 x 25 elements along the initial crack and 8 x 150 elements along the rest of the beam. These elements are plane strain linear quadrilateral ones. Figure 14 presents the geometry and the mesh of the benchmark, the parameters are given in the Table 3. The analytical result is introduced in appendix C.

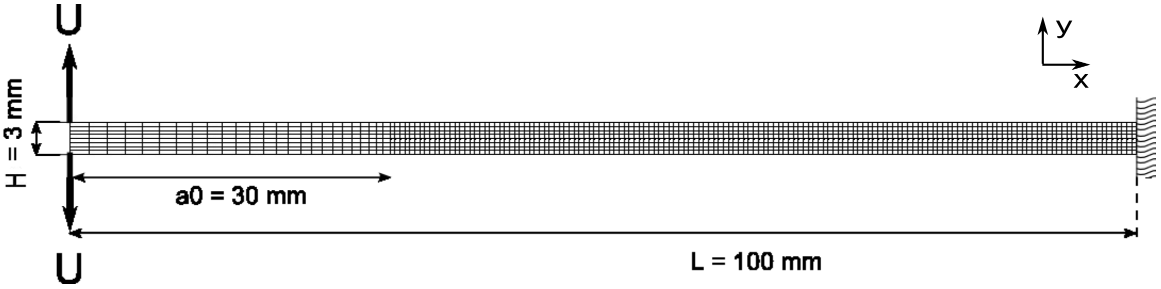


Figure 14: Mesh and boundary conditions applied on the DCB benchmark

4.1.2 Results

For this benchmark, relevant solvers are able to achieve calculation which gives results close to the analytical solution, presented in appendix C. In order to observe such results, three different pre-implemented solvers are tested, with a convergence tolerance of 10^{-3} and an initial fraction of 0.01.

4.1.2.1 The Newton-Raphson method (see figure 15)

This procedure is a constant time step method, which is configured for 1000 increments in this work. The test considers displacement controlled boundary conditions, so this procedure is equivalent to a displacement controlled method. However, a standard displacement controlled procedure with an iterative Newton-Raphson solution fails to converge in the case of snap-back. These ones are in fact artifacts of discretization due to the failure of a cohesive zone element and lead to numerical issues. So,

Table 3: DCB benchmark parameters

Young's modulus E [GPa]	130
Poisson's ratio ν	0.3
Cohesive energy G_c [N/mm]	0.36
Shear/Normal ratio β	1
Critical opening 1 (δ_{c1})	0.1
Critical opening 2 (δ_{c2})	0.01
Critical opening 3 (δ_{c3})	0.001

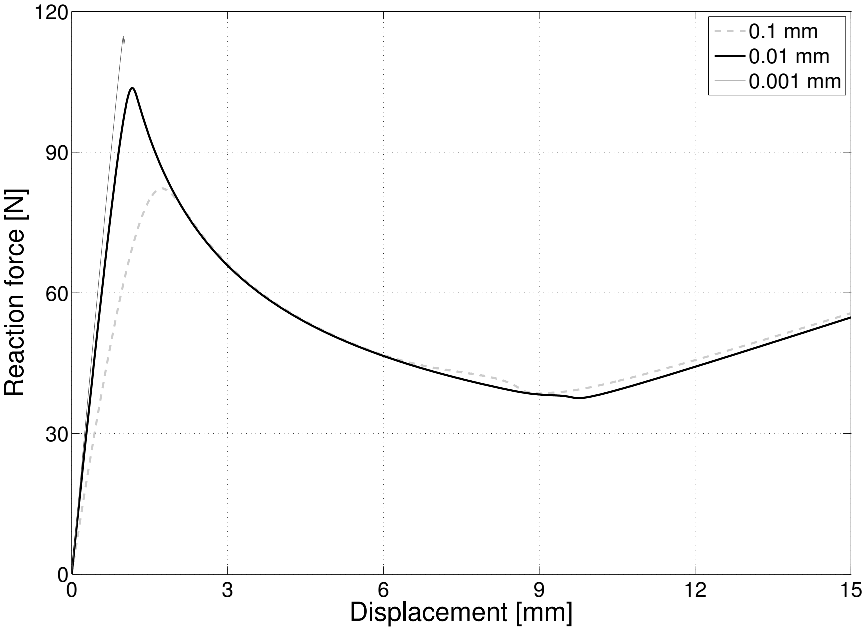


Figure 15: Force-displacement curves for the DCB benchmark - 3 different critical opening values - Newton-Raphson method

achieving the calculation for δ_c=0.001mm seems to be impossible and the calculation even fails in a early stage for the brittle material. This method is not adapted to brittle interfaces.

4.1.2.2 The global arc-length method (see figure 16)

It is a frequently used method in non-linear mechanical engineering. In this procedure, global means that all degrees of freedom have the same weight in the load control. The result for the more brittle material points out that this procedure does not always converge, especially in the case of sharp artificial limit points, which are the result of a too coarse mesh. As a result, for the more brittle material, this method wants to follow the unloading path and needs also many iterations. Although the convergence behavior is better than with the Newton-Raphson method, the global method is also not capable to handle strong localized deformations.

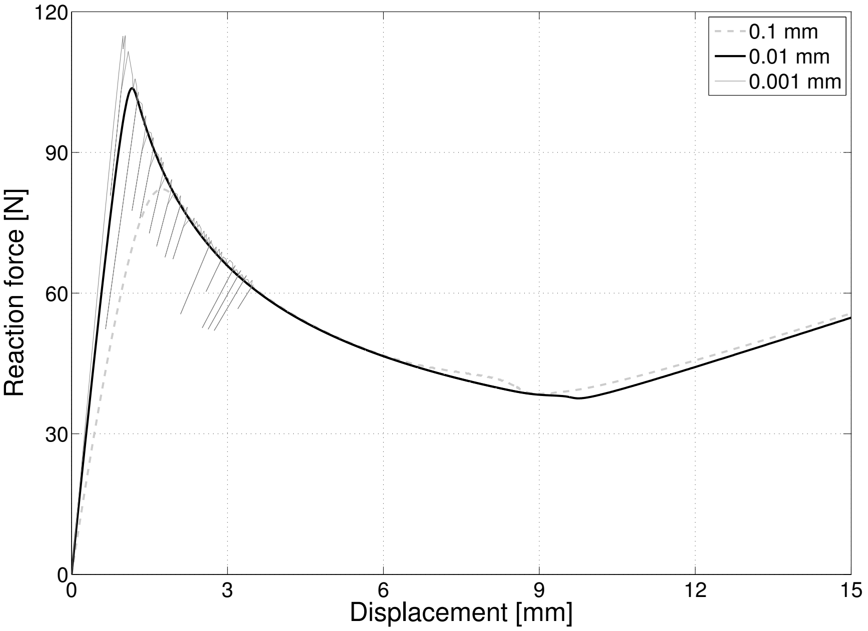


Figure 16: Force-displacement curves for the DCB benchmark - 3 different brittlenesses - Global arc-length method

4.1.2.3 The local arc-length method (see figure 17)

This solver is a weighted subplane control [15], which uses a function defined as a weighted sum of the degrees of freedom. This method is more robust than the previous one, because only the relevant degrees of freedom are taken into account. So, the convergence of the result is improved and the solver can easily perform the more brittle case, while the more ductile interface tests are solved faster, as shown in Table 4. However, for the delamination regime, moderate oscillations with the critical opening $\delta_C = 0.001$ mm appear. These fluctuations are due to a too coarse mesh, which leads to numerical issues. Each oscillation is the result of the failure of a cohesive zone element. These are artifacts of the discretization, but this solver is able to follow sharp snap-backs, contrary to the two previous ones. A refinement of the mesh is a solution in order to make the fluctuation disappear, as shown on the figure 18. However, the time of calculation increases in a drastically way. It can be

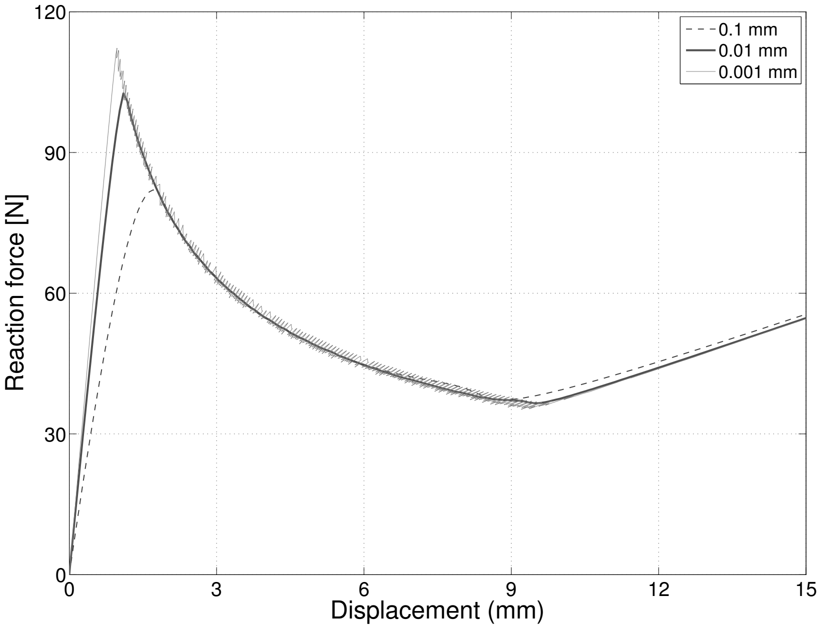


Figure 17: Force-displacement curves for the DCB benchmark - 3 different brittlenesses - Local arc-length method

concluded that the local arc-length solved is well-suited to capture strongly localized deformations.

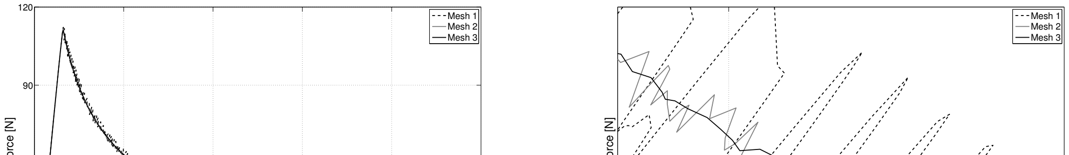


Figure 18: Force-displacement curves for the DCB benchmark - 3 different meshes - Local Arc-Length method

Table 4: Performances of the different solvers for the DCB benchmark

Solver	δ_c	Increments	Time
Newton-Raphson	0.1	1000	453
Newton-Raphson	0.01	1000	1285
Newton-Raphson	0.001	-	-
Global arc-length	0.1	662	256
Global arc-length	0.01	1443	2544
Global arc-length	0.001	446	10713
Local arc-length	0.1	203	66
Local arc-length	0.01	205	152
Local arc-length	0.001	541	576

4.1.2.4 Energy release method

The only difference between the benchmarks below and the one from the previous part 4.1 is the boundary conditions. Indeed, due to the current implementation, a force prescribed boundary condition replaces the displacement one. In order to obtain an evaluation of the performance of the solver and to appraise the results achieved by the energy based equations, the DBC benchmark has first been tested on another software package, called Dawn and co-developed by J.J.C. Remmers from the TU/e (Eindhoven University of Technology). The obtained results were convincing, so the decision to adapt this solver on the MSc Software package has been done.

In figure 19, the curves point out that the quality of the resolution is exactly the same: same global equilibrium path, some oscillations on the delamination part. However, the time of computation is smaller for the energy release method. Since the previous part 4.1 shows that the local arc-length method is the more effective, all energy release method results are compared with the local arc-length method ones.

The number of iterations and the times of computation are given in the table 5. The number of increments is around 2 times lower when considering the energy release arc-length method. This is due to the fact that each incremental step is optimized, with a well evaluated predictor. The difference in CPU time is even more impressive.

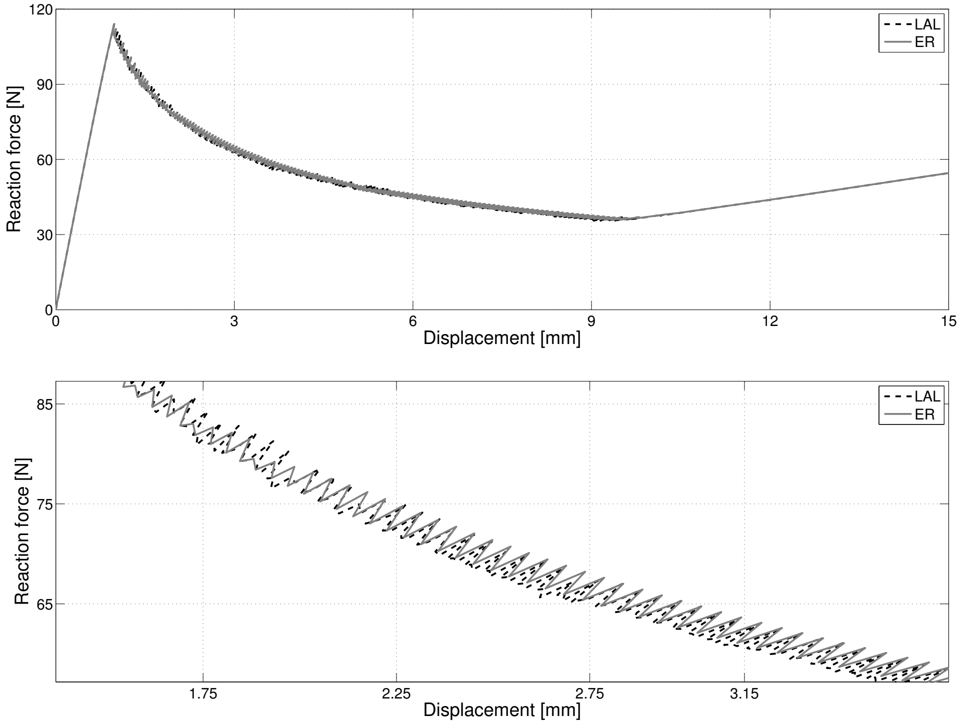


Figure 19: Force-displacement curves for the DCB benchmark - Comparison ER method vs LAL method - Mesh 1 - Critical opening of 0.001 mm

The explication is related to the number of iterative steps. This one is close to the optimized value (5 iterations) in the case of the energy release method, whereas it oscillates with the local arc-length method. At the end, the total number of iterations is heavily reduced, which means that the total number of calculation is reduced too, so is the time of calculation .

4.1.3 Finer meshes

The result obtained for the more brittle material, $\delta_c = 0.001$ mm, can be smoothed with a finer mesh. The figure 18 points out that fluctuations are really artifacts of calculation. The mesh 2 (mesh 3) corresponds to a mesh 2 times (respectively, 4 times) finer than the mesh 1 which is the one used in the previous section. The improvement is obvious, but the time of calculation increases significantly, with more than 3 hours of calculation for mesh 3.

4.2 Perforated Double Cantilever Beam benchmark (PDCB)

4.2.1 Description of the benchmark

The perforated double cantilever beam (PDCB) test is an interesting benchmark test for cohesive zone elements, in presence of real snap-backs. It is inspired by the paper from Verhoosel et al [5]. This specimen consists of a beam of length L = 2.0625 mm, width W = 1 mm and height H = 1 mm, regularly perforated by holes with a diameter of 0.2 mm and spaced from 0.375 mm. This test, also analyzed under plane strain conditions, consists in pulling apart the two parts of the beam. Indeed, the two end points of the beam are prescribed a displacement in opposite y- direction up to u = 0.06 mm, while the other end of the sample is fixed in x- and y- direction. The cohesive zone elements are placed in the middle of the beam, on the line where the centers of holes are located. In this benchmark, the parameters of the cohesive zone are kept constant, while the Young's modulus of the linear elastic material of the beam is changed, in order to modify the severity of the snap-backs in the equilibrium path. The beam is meshed with 218 x 64 elements which are plane strain linear quadrilateral elements. The critical opening of the interface imposes such a fine mesh. The benchmark and the mesh are shown in figure 20 and the parameters are

Table 5: ER method vs LAL method. Performances for the DCB benchmark

Solver	Mesh	δ_c	Increments	Time
Energy Release AL	1	0.01	102	62
Local AL	1	0.01	205	152
Energy Release AL	2	0.01	106	242
Local AL	2	0.01	324	1013
Energy Release AL	1	0.001	379	275
Local AL	1	0.001	541	576
Energy Release AL	2	0.001	392	1090
Local AL	2	0.001	437	2265

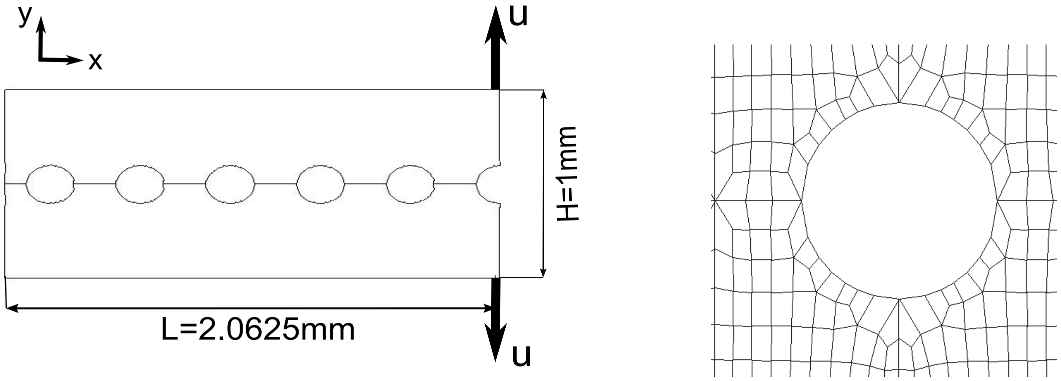


Figure 20: Mesh and boundary conditions applied on the PDCB benchmark

given in Table 6.

4.2.2 Results

This benchmark is used in order to analyze how the solver can pass through the snap-back points, so the Newton-Raphson method is not applicable for this case. In order to perform the calculation, both global and local arc-length methods will be used:

4.2.2.1 The global arc-length method

Figures 21 are illustrative of the limitation of the global arc-length method. Indeed, this solver is not able to pass through severe snap-backs, even with this relatively fine mesh. The main problem of the solver is that it may not distinguish the elastic unloading path from the dissipative one and so the equilibrium path follows only

Table 6: PDCB benchmark parameters

Young's modulus E_1 [MPa]	500
Young's modulus E_2 [MPa]	100
Poisson's ratio ν	0.3
Cohesive energy G_c [N/mm]	2.5×10^{-3}
Maximum traction t_{max} $[N/mm^2]$	1
Shear/Normal ratio β	1

Table 7: Performances of the different solvers for the PDCB benchmark

Solver	E [MPa]	Increments	Time
Global AL	500	873	2527
Local AL	500	433	879
Energy Release AL	500	493	720
Global AL	100	-	-
Local AL	100	508	6019
Energy Release AL	100	589	1064

the elastic solution, what is not physical. This phenomenon is more pronounced in the case of the brittle material (Figure 21 - right), where many unloading paths are present. This problem is due to the high localization of the deformations, which involves that some local degrees of freedom dominate the overall mechanical behavior. However, the global arc-length method uses all the degrees of freedom what generates an unstable numerical result.

4.2.2.2 The local arc-length method

This solver also takes into account all the degrees of freedom, however they are weighted in order to give more importance to the relevant ones, in such a way that the damage in the cohesive zone elements is used as weighting factor. The enhancement of the method is obvious, this solver can follow perfectly the equilibrium path, even for the brittle material. However, the table 7 points out that the time of calculation significantly increases in that case. The quality of the mesh can not be made coarser, otherwise the solver is not able to achieve correctly the calculation.

4.2.2.3 Energy release method

The results of the energy release method appears one more time as a good improvement of local arc-length method. Due to the force prescribed boundary conditions,

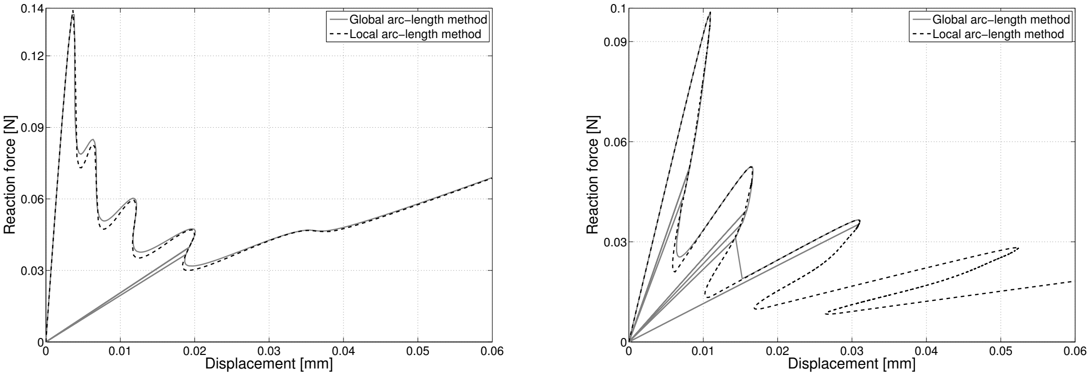


Figure 21: Force-displacement curves for the PDCB benchmark - E=500 MPa (left) - E=100 MPa (right)

one more snap-back is present as showed on the figure 22. Moreover, this snap-back is the more severe encountered during the simulation, which is stopped by selecting a maximum displacement value.

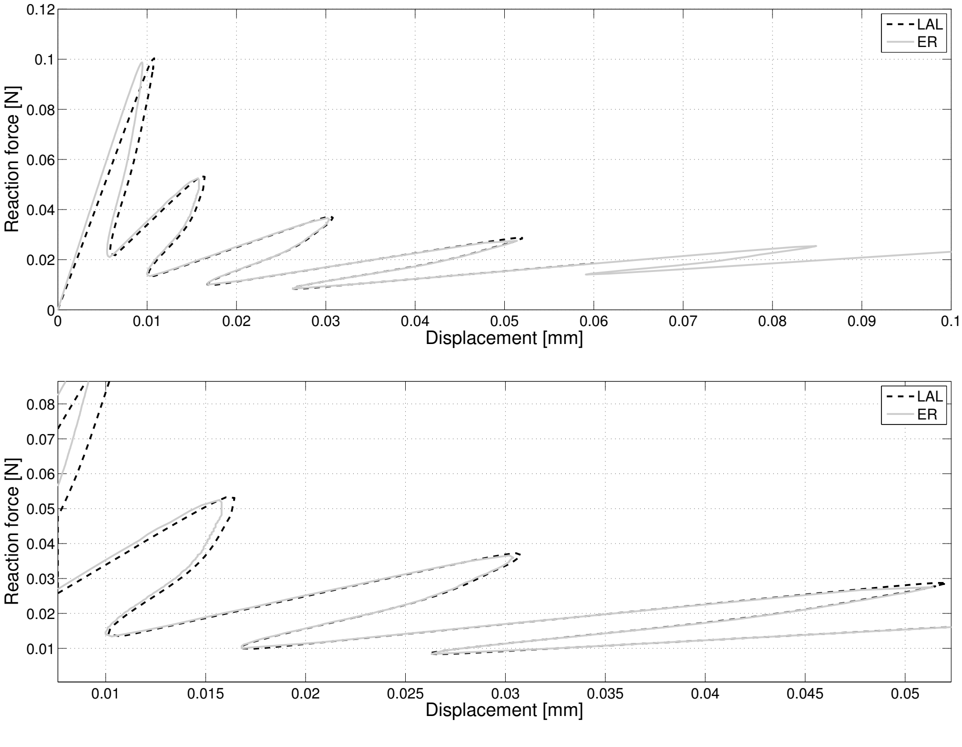


Figure 22: Force-displacement curves for the PDCB benchmark - Comparison ER method vs LAL method - E = 100 MPa

As pointed out in the chart 7, the time of calculation are still better.

CHAPTER V

CONCLUSION AND FUTURE WORK

The need of improvement of the ability of finite element solvers to predict highly localized deformations has motivated the implementation of this energy release solver, which is able to deal with the consider problem. Although the previously implemented LAL solver gives acceptable results as well, it appeared that this solver is more efficient. Indeed, the improvement brought by the new energy release constraint equation is obvious in terms of robustness and computational time. This constraint equation derived from the first law of thermodynamics is so an efficient way to study the crack propagation using cohesive zone elements. The programmed algorithm, for the software package MSc Marc/Mentat, is able to switch from the standard global arc-length method to the energy released one in order to achieve both the non-dissipative part and the dissipative one of the equilibrium path. The great improvement comes from the fact that the released energy is always an increasing variable during a failure process. Thus, this solver is particularly able to deal with snap-back and snap-through behaviors. The results provided in the two numerical examples presented in this study by the double cantilever beam and the perforated cantilever beam, demonstrated the efficiency of the method, which gives better results than the pre-implemented solvers of the software package. The efficiency is clearly demonstrated in the case of a boundary conditions expressed in terms of external forces.

The future steps of the work can be clearly defined and concerned mostly the implementation of the solver using boundary conditions expressed in terms of displacement. The first step is to obtain a correct expression of the stiffness matrix, because the matrix is currently not updated at every step. From that, using the presented equations and variables in the part 3.4.5, the achievement of the method is straightforward. Once this implementation finished, the next step is to combine this solver with the pre-implemented cohesive zone element provided by MSC.Marc/Mentat instead of using user-defined cohesive zone elements. Since the displacement control will be applied this step may be relatively easy. Finally, the displacement solver has to be extended to 3D.

APPENDIX A

TRACTION-SEPARATION LAWS

A.1 Xu & Needleman Traction Separation Law [24]

Xu and Needleman traction separation law is a model to study the void nucleation at the interface of particle and matrix material, especially in brittle material and to deal with the crack propagation along the interface of bimaterials.

This traction separation law is based on the potential function Φ which allows normal and tangential decohesion:

$$\Phi\left(\mathbf{\Delta}\right) = \Phi_n + \Phi_n exp\left(-\frac{\Delta_n}{\delta_n}\right)\left\{\left[1 - r + \frac{\Delta_n}{\delta_n}\right]\frac{1-q}{r-1} - \left[q + \left(\frac{r-q}{r-1}\right)\frac{\Delta_n}{\delta_n}\right] exp\left(-\frac{\Delta_t^2}{\delta_t^2}\right)\right\} \tag{43}$$

with:

$$\begin{cases} q = \dfrac{\Phi_t}{\Phi_n} \\ r = \dfrac{\Delta_n^*}{\delta_n} \end{cases} \qquad \begin{cases} \Phi_n = exp\left(1\right)\tau_{max}\delta_n \\ \Phi_t = \sqrt{\dfrac{exp\left(1\right)}{2}}\sigma_{max}\delta_t \end{cases}$$

where Φ_n and Φ_t are respectively normal and tangential cohesive energy (Φ_n=G_c) and Δ_n^* is the value of Δ_n after the complete shear separation. τ_{max} and σ_{max} are normal and tangential cohesive strength and δ_n and δ_t are the critical openings.

The cohesive surface traction is given by :

$$\mathbf{T} = \frac{\partial\Phi}{\partial\mathbf{\Delta}} \tag{44}$$

So, by considering the equation (43), the traction separation law is given by:

$$\begin{cases} T_n = -\dfrac{\Phi_n}{\delta_n}\left\{exp\left(-\dfrac{\Delta_n^2}{\delta_n^2}\right) + \dfrac{1-q}{r-1}\left[1 - exp\left(-\dfrac{\Delta_n^2}{\delta_n^2}\right)\right]\left[r - \dfrac{\Delta_n}{\delta_n}\right]\right\} \\ T_t = -\dfrac{\Phi_n}{\delta_n}\left(2\dfrac{\Delta_n}{\delta_t}\right)\dfrac{\Delta_t}{\delta_t}\left\{q + \dfrac{r-q}{r-1}\dfrac{\Delta_n}{\delta_n}\right\} exp\left(-\dfrac{\Delta_n}{\delta_n}\right) exp\left(-\dfrac{\Delta_t^2}{\delta_t^2}\right) \end{cases} \tag{45}$$

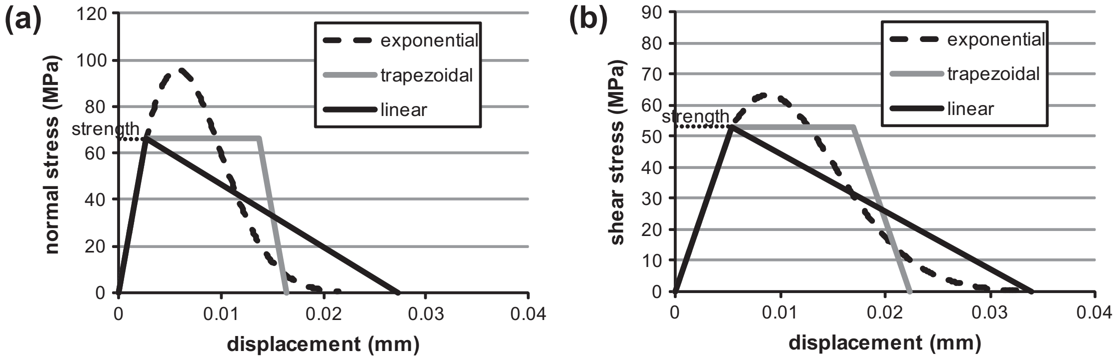


Figure 23: Examples of different traction separation laws considering normal traction (a) and shear traction (b) [13]

A.2 Other traction separations laws

Many other traction separation laws have been implemented : 'Barenblatt' (1959), 'Tvergaard & Hutchinson' (1992), 'Geubelle & Bayler' (1997), 'Ortiz & Pandolfi' (1999),... Regarding the considered problem, it may be useful to consider a specific law. Many of them are presented and compared in different papers such as in [17] and in [16], and are able to treat different problems : particle-matric decohesion, solute segregation, decohesion of interface under hydrostatic tension, impact... Most of them are based on exponential model. However, as shown in Figure 23, there are also some non-smooth traction separation laws such as the trapezoidal one which is adapted to deal with the crack propagation in elasto-plastic material, or the bi-linear traction separation law which can treat the delamination engendered by low-velocity impact.

APPENDIX B

FLOW DIAGRAM OF THE SOLUTION PROCEDURE

Figure 24 shows the flow sequence of MSC.Marc for solving a nonlinear system

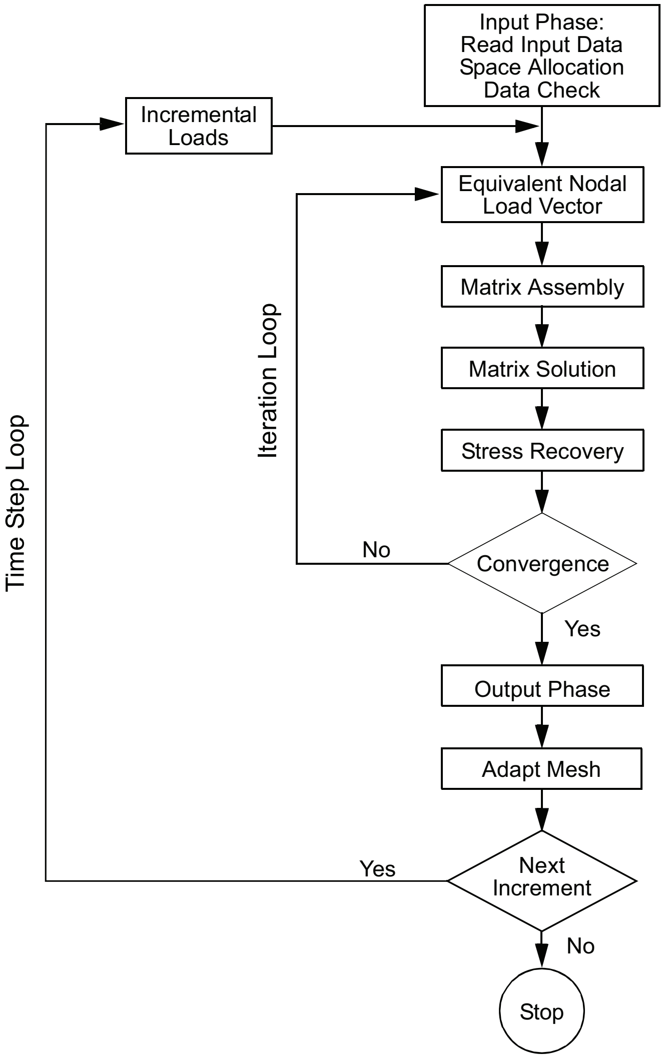


Figure 24: Marc Flow Diagram [22]

APPENDIX C

DCB BENCHMARK - ANALYTICAL SOLUTION

As explain in [21], the analytical solution can be found out. Figure 25 shows the global response in terms of the reaction force versus the prescribed displacement. The used parameters are the one expressed in the Table 3. This curve clearly points out three different regimes:

- The initial bending : This regime is linear elastic, and is governed by the stiffness of the two strips. The two parts of the beam bent over the initial crack. The slope of the curve depends on the virgin stiffn The governing equation is :

$$F = \frac{3EI}{a_0^3} u \tag{46}$$

- The delamination : This regime corresponds to the debonding of the interface. The cohesive zone elements are opening and so the stiffness decreases. As shown on the governing equation (47), it is a hyperbolic response. This regime links the cohesive energy released to the forces required for the delamination, which decreases when the cohesive zone elements are opening. A high adhesive

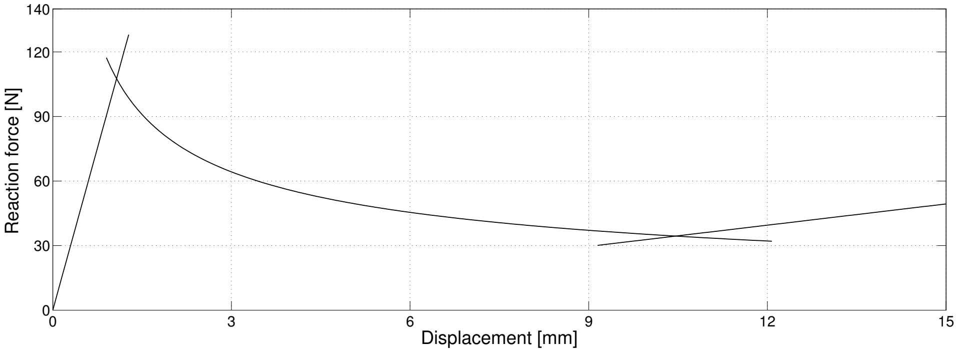


Figure 25: Analytical force-displacement curve for the DCB benchmark

strength of the interface, which means a high value of G_c, results in higher values of the reaction force F.

$$F = \sqrt{\frac{(WG_cEI)^{3/2}}{3EIu}} \tag{47}$$

- The final bending : The interface crack has almost reached the other end of the beam and another linear regime is observed. This relation is the same than the initial bending, however this time the measure of the opening is not a_0, but the total length of the beam L.

$$F = \frac{3EI}{L^3}u \tag{48}$$

The parameters used in the governing equations are the following :

- F : the reaction force of one strip of the beam
- E : the Young's modulus of the beam
- I : the moment of inertia
- u : the prescribed displacement
- a_0 : the length of the initial crack
- W : the thickness of the beam
- G_c : the fracture toughness

APPENDIX D

SHERMAN-MORRISON FORMULA

In order to have an easier reading of the demonstration,

the indices have been neglected.

Let consider the augmented system of equilibrium equation which can be written as :

$$Ax = y \tag{49}$$

$$\text{with :} \qquad A = \begin{bmatrix} \mathbf{K} & -\bar{\mathbf{f}} \\ \mathbf{g}^T & w \end{bmatrix} \qquad \begin{cases} \mathbf{g} = \dfrac{1}{2}\alpha\bar{\mathbf{f}} \\ w = -\dfrac{1}{2}\mathbf{u}^T\bar{\mathbf{f}} \end{cases} \qquad \begin{bmatrix} d\mathbf{u} \\ d\alpha \end{bmatrix}$$

All these notations refer to the ones used in the equations of the part (3.4.1). In order to solve the system (49), the inverse of Jacobian **A** is calculated by using the Sherman-Morrison formula for a non-singular matrix, as shown just below:

$$\left(\mathbf{C} + \mathbf{u}\mathbf{v}^T\right)^{-1} = \mathbf{C}^{-1} - \frac{\mathbf{C}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{C}^{-1}}{1 - \mathbf{v}^T\mathbf{C}^{-1}\mathbf{u}} \tag{50}$$

.

This matrix **A** can be re-written as a sum of 4 matrices $\mathbf{A}_i$, with $i = 1..4$:

$$\begin{bmatrix} \mathbf{K} & -\bar{\mathbf{f}} \\ \mathbf{g}^T & w \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\bar{\mathbf{f}} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{g}^T & -1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & w \end{bmatrix} \tag{51}$$

In order to be able to use this expression in the Sherman Morrison formulation, we have to decompose the third last matrices as a product of vectors. So we come up with:

$$
\begin{aligned}
A_2 &= \begin{bmatrix} \mathbf{0} & -\bar{\mathbf{f}} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} -\bar{\mathbf{f}} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix} \\
A_3 &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{g}^T & -1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{g}^T & -1 \end{bmatrix} \\
A_4 &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & w \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ w \end{bmatrix} \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix}
\end{aligned}
\tag{52}
$$

We can now apply the Sherman Morrisson expression to the two first matrices, such that:

$$
\begin{aligned}
A_{12}^{-1} &= (A_1 + A_2)^{-1} \\
&= \begin{bmatrix} \mathbf{K}_{-1} & \mathbf{K}_{-1}\bar{\mathbf{f}} \\ \mathbf{0} & 1 \end{bmatrix}
\end{aligned}
\tag{53}
$$

Let now add the matrix A_3, such that:

$$
\begin{aligned}
A_{123}^{-1} &= (A_1 + A_2 + A_3)^{-1} = (A_{12} + A_3)^{-1} \\
&= \begin{bmatrix} \mathbf{K}^{-1} & 0 \\ \mathbf{0} & 1 \end{bmatrix} - \frac{1}{\mathbf{g}^T\mathbf{K}\bar{\mathbf{f}}} \begin{bmatrix} \mathbf{K}^{-1}\bar{\mathbf{f}}\mathbf{g}^T\mathbf{K}^{-1} & \mathbf{K}^{-1}\bar{\mathbf{f}} \\ \mathbf{g}^T\mathbf{K}^{-1} & 1 + \mathbf{g}^T\mathbf{K}^{-1}\bar{\mathbf{f}} \end{bmatrix}
\end{aligned}
\tag{54}
$$

The fourth matrix can now be add to obtain to complete inverse of A:

$$
\begin{aligned}
A^{-1} &= (A_1 + A_2 + A_3 + A_4)^{-1} = (A_{123} + A_4)^{-1} \\
&= \begin{bmatrix} \mathbf{K}^{-1} & 0 \\ \mathbf{0} & 1 \end{bmatrix} - \frac{1}{\mathbf{g}^T\mathbf{K}\bar{\mathbf{f}} - w} \begin{bmatrix} \mathbf{K}^{-1}\bar{\mathbf{f}}\mathbf{g}^T\mathbf{K}^{-1} & \mathbf{K}^{-1}\bar{\mathbf{f}} \\ \mathbf{g}T\mathbf{K}^{-1} & 1 + \mathbf{g}^T\mathbf{K}^{-1}\bar{\mathbf{f}} - w \end{bmatrix}
\end{aligned}
\tag{55}
$$

The multiplication of this system with the vector y of the matrix system (49) gives the following result :

$$
\begin{bmatrix} d\mathbf{u}_i^{(n)} \\ d\alpha_i^{(n)} \end{bmatrix} = \begin{bmatrix} d\hat{\mathbf{u}} \\ q \end{bmatrix} - \frac{1}{\mathbf{g}^T d\bar{\mathbf{u}} + w} \begin{bmatrix} d\bar{\mathbf{u}}(\mathbf{g}^T d\hat{\mathbf{u}} + q) \\ \mathbf{g}^T d\hat{\mathbf{u}} + (1 + \mathbf{g}^T d\bar{\mathbf{u}} + w)q \end{bmatrix}
\tag{56}
$$

with the following variables:

$$
\begin{cases} d\hat{\mathbf{u}} = \mathbf{K_t}^{-1}\mathbf{r} \\ d\bar{\mathbf{u}} = \mathbf{K_t}^{-1}\bar{\mathbf{f}} \end{cases}
\tag{57}
$$

APPENDIX E

NANOINTERFACE PROJECT

NanoInterface [19] is the acronym for ”Knowledge-based multi-scale modelling of metal-oxidepolymer interface behaviour for micro- and nanoelectronics”. This research project was initiated by Philips Applied Technologies, started on the 1st September 2008 and will be active until the 31st August 2011.

The aim of the project is to enhance the performance of the microelectronics components while improving the miniaturization. It is very useful for microelectronics system, especially the so-called Systems-In-Package, in order to get ’zero-defect’ products. This initiative also includes the development of a multi-scale approach, from the nanoscale to the macroscale.

This research will result in the development of a user-friendly software which will incorporate chemical, physical and mechanical information for the different level models.

Many different partners take an active part into this project, such as:

- Industrial Partners:
  - Philips Applied Technologies
  - NXP Semiconductors
- Research Partners:
  - Fraunhofer IZM
  - AMIC
- Education Partners:
  - Georgia Institute of Technology Lorraine
  - Delft University of Technology

REFERENCES

[1] A.A. Griffith, “The phenomena of rupture and flow in solids,” *Philosophical Transactions of the Royal Society of London,*, vol. 221, p. 163198, 1921.

[2] B.A.E. van Hal, R.H.J. Peerlings, and M.G.D. Geers, “A local arc-length control method for cohesive zone modelling,” *Submitted to Computer Methods in Applied Mechanics and Engineering*, December 2006.

[3] B.A.E van Hal, R.H.J. Peerlings, M.G.D Geers, and O. van der Sluis, “Cohesive zone modeling for structural integrity analysis of IC interconnects,” *Microelectronics Reliability*, vol. 47, pp. 1251–1261, 2007.

[4] C. Fang, A. Le Corre, and D. Yon, “Copper electroplating into deep microvias for the sip application,” *Microelectronic Engineering*, vol. 88, pp. 749–753, May 2011.

[5] C.V. Verhoosel, J.J.C. Remmers, and M.A. Gutiérrez, “A dissipation-based arc-length method for robust simulation of brittle and ductile failure,” *International Journal for Numerical Methods in Engineering*, vol. 77, no. 9, pp. 1290–1321, 2008.

[6] E. Riks, “An incremental approach to the solution of snapping and buckling problems,” *International Journal of Solids and Structures*, vol. 15, pp. 529–551, 1979.

[7] E.F. Rybicki and M.F. Kanninen, “A finite element calculation of stress intensity factors by a modified crack closure integral,” *Engineering Fracture Mechanics*, vol. 9, no. 4, pp. 931–938, 1977.

[8] Felipp, C. A., “Chapter 3 : Residual force equations.” Nonlinear Finite Element Methods (ASEN 6107) - Fall 2010 - Department of Aerospace Engineering Sciences University of Colorado at Boulder, 2010.

[9] G. Irwin, “Analysis of stresses and strains near the end of a crack traversing a plate,” *Journal of Applied Mechanics*, vol. 24, pp. 361–364, 1957.

[10] H. Sun, S. Rajendran and D. Q. Song, *Finite Element Analysis on Delamination Fracture Toughness of Composite Specimens.* Materials Technology Application Centre Singapore Productivity and Standards Board 1, Science Park Drive Singapore 118221.

[11] J. Remmers, *Discontinuities in materials and structures*, ch. Appendix C.1, pp. 213–215. 2006.

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

[13] M. Ridha, V.B.C. Tan and T.E. Tay, “Traction separation laws for progressive failure of bonded scarf repair of composite panel,” *Composite Structures*, vol. 93, p. 12391245, 2011.

[14] M.A. Gutiérrez, “Energy release control for numerical simulations of failure in quasi-brittle solids,” *Communications in Numerical Methods in Engineering*, vol. 20, pp. 19–29, 2003.

[15] M.G.D. Geers, “Enhanced solution control for physically and geometrically non-linear problems. part 1 - the subplane control approach,” *International Journal for Numerical Methods in Engineering*, vol. 46, pp. 177–204, 1999.

[16] N. Chandra and C. Shet, “Analysis of energy balance when using cohesive zone models to simulate fracture processes,” *Journal of Engineering Materials and Technology*, vol. 124, pp. 440–451, October 2002.

[17] N. Chandra, H. Li, C. Shet, and H. Ghonem, “Some issues in the application of cohesive zone models for metalceramic interfaces,” *International Journal of Solids and Structures*, vol. 39, no. 10, pp. 2827–2855, 2002.

[18] Olaf van der Sluis, Sander P.M. Noijen, Peter H.M. Timmermans1, and Jean-Baptiste Bouquet, “The influence of microscopic roughness on macroscopic adhesion properties of polymer-metal interfaces,”

[19] Philips Applied Technologies, “Nanointerface project.” urlhttp://www.nanointerface.eu/, October 2010.

[20] R. Krueger, “The virtual crack closure technique: History,approach and applications,” Tech. Rep. 2002-10, NASA, 2002.

[21] Rene Kregting, Coaches: R.H.J. Peerlings, O. van der Sluis, M.G.D. Geers, “Cohesive zone models towards a robust implementation of irreversible behaviour,” internal report, Eindhoven University of Technology, February 2005.

[22] Software, M., *MSC.Marc Volume A: Theory and User Information*, 2004.

[23] Wikipedia, “Fracture.” urlhttp://en.wikipedia.org/wiki/Fracture, October 2010.

[24] X.-P. XU and A. NEEDLEMAN, “Numerical simulations of fast crack growth in brittle solids,” *Journal of the Mechanics and Physics of Solids*, vol. 42, pp. 1397–1434, September 1994.